Exploiting edge features for scene text understanding systems and scene text searching applications

Dinh Nguyen Van

THÈSE DE DOCTORAT DE SORBONNE UNIVERSITÉ -
l'UNIVERSITÉ PIERRE ET MARIE CURIE

Spécialité

Informatique

École doctorale Informatique, Télécommunications et Électronique (Paris)

Présentée par

Dinh NGUYEN VAN

Pour obtenir le grade de

DOCTEUR de SORBONNE UNIVERSITÉ - l'UNIVERSITÉ PIERRE ET MARIE CURIE

Sujet de la thèse :

Exploitation de la détection de contours pour la compréhension de texte dans une scène visuelle

Directeur de thèse : Mounir MOKHTARI
Co-directeur de thèse : Shijian LU

Soutenue le 02/ May/ 2018

devant le jury composé de :

M. François BRÉMOND	Rapporteur
M. Frédéric LERASLE	Rapporteur
M. Nicolas LOMÉNIE	Examinateur
Ms. Marie BABEL	Examinateur
M. Mounir MOKHTARI	Directeur de thèse

DOCTORAL THESIS OF SORBONNE UNIVERSITÉ - UNIVERSITÉ PIERRE AND MARIE CURIE

specialty

Informatique

École doctorale Informatique, Télécommunications et Électronique (Paris)

Presented by

Dinh NGUYEN VAN

To obtain the degree of

DOCTOR of SORBONNE UNIVERSITÉ - l'UNIVERSITÉ PIERRE ET MARIE CURIE

Thesis title :

Exploiting edge features for scene text understanding systems and scene text searching applications

Director of thesis : Mounir MOKHTARI
Thesis co-advisor : Shijian LU

Defended on 02/ May/ 2018

Jury committee composed of :

M. François BRÉMOND	Reviewer
M. Frédéric LERASLE	Reviewer
M. Nicolas LOMÉNIE	Examiner
Ms. Marie BABEL	Examiner
M. Mounir MOKHTARI	Director of thesis

Acknowledgements

At first, I would like to thank my parents who raised me up. Without their great efforts and sacrifices, I could not have had the opportunity to obtain this high education program and make my dream come true. Without their encouragements, I could not concentrate on my research work with full energy and complete it properly.

I am gratefully indebted to Professor Mounir MOKHTARI, the Director of the France-Singapore joint laboratory, Image and Pervasive Access Lab (IPAL), in Singapore as well as the Director of my thesis. He gave me a great chance to start my doctoral research and has supported me during my study. I am thankful for his patience, guidance and ideas which encouraged me a lot to transfer my research work to an application which can participate in improving human life quality.

I own the deepest gratitude to my co-supervisor, Professor Shijian LU, Assistant Professor in School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore. I am thankful for sharing his immense knowledges in Computer Vision, especially his research work on scene text detection and recognition. He also encouraged me to improve my experience and skill in writing and organizing scientific papers and reports.

My sincere thanks also goes to Professor Nizar OUARTI, Associate Professor in Sorbonne University - University Pierre and Marie CURIE (UPMC), Paris 6, France, and French delegate of Centre National de la Recherché Scientific (CNRS) in IPAL, Singapore. My success in this research includes his participation by a lot of useful discussions. Some of my difficulties have been solved under his support.

I thank the Agency for Science, Technology and Research - Institute for Infocomm Research (ASTAR - I2R) for supporting this research work. They have provided comfortable facilities, including an office room, computers, printers, scanners and a lot of assistance from their Information Technology help desk.

I would like to acknowledge all members of the IPAL laboratory in Singapore, who provided a friendly research environment : Hoa T., Ibrahim S., Martin K., Davi, Matthew, Thomas B., Joaquim B.M., Anna, Antoine M., Angela Sáenz, Christophe J., Fabien Clavier, Anssuya. I have special thanks to Hoa T., Ibrahim S., Martin K., Davi, Matthew, Thomas B., Antoine M., Angela S. for sharing their experiments on which I can solve my difficulties during my studies. I also want to say thank you to Martin K. for his help in setting the server connection for my developed online scene text searching system and to Matthew for his advice on web based application development.

Finally, please allow me to express my genuine thanks to my thesis reviewers, Professor François BRÉMOND, and Professor Frédéric LERASLE, and the whole jury members. Their judgements and comments were very helpful in edifying this thesis.

"Education is the key to unlock the golden door of freedom"

George Washington Carver

For

Dinh NGUYEN VAN

Resume

L'intérêt porté à la détection de contours pour la compréhension de texte dans une scène visuelle a été croissant au cours des dernières années comme en témoignent un grand nombre d'applications telles que les systèmes de reconnaissance de plaque d'immatriculation de voiture, les systèmes de navigation, les voitures autonomes basées sur la reconnaissance des panneaux de signalisation, etc. Dans cette recherche, nous abordons les défis de la conception de systèmes de lecture de texte de scène automatique robustes et fiables. Deux étapes majeures du système, à savoir, la localisation de texte dans une scène et sa reconnaissance, ont été étudiées et de nouveaux algorithmes ont été développés pour y remédier.

Nos travaux sont basés sur l'observation qu'indiquer des régions de texte de scène primaire qui ont forte probabilité d'être des textes est un aspect important dans la localisation et la reconnaissance de cette information. Ce facteur peut influencer à la fois la précision et l'efficacité des systèmes de détection et de reconnaissance. Inspirées par les succès des recherche de proposition d'objets dans la détection et la reconnaissance objet général, deux techniques de proposition de texte de scène ont été proposées, à savoir l'approche Text-Edge-Box (TEB) et l'approche Max-Pooling Text Proposal (MPT). Dans le TEB, les fonctionnalités bottom-up proposées, qui sont extraites des cartes binaires de contours de Canny, sont utilisées pour regrouper les contours connectés et leur attribuer un score distinct. Dans la technique MPT, une nouvelle solution de groupement est proposée, qui est inspiré de l'approche Max-Pooling. À la différence des techniques de regroupement existantes, cette solution ne repose sur aucune règle heuristique spécifique liée au texte ni sur aucun seuil pour fournir des décisions de regroupement. Basé sur ces résultats, nous avons conçu un système pour comprendre le texte dans une scène visuelle en intégrant des modèles a l'état de l'art en reconnaissance de texte, où une suppression des faux positifs et une reconnaissance de mot peut être traitée simultanément. De plus, nous avons développé un système assisté de recherche de texte dans une scène en construisant une interface web en complément du système de compréhension de texte. Le système peut être consulté via le lien : *dinh.ubismart.org:27790*.

Des expériences sur diverses bases de données publiques montrent que les techniques proposées surpassent les méthodes les plus modernes de reconnaissance de textes sous différents cadres d'évaluation. Le système complet propose surpasse également d'autres systèmes complets de reconnaissance de texte et a été soumis à une compétition de lecture automatique dans laquelle il a montré sa performance et a atteint la cinquième position dans le classement (Dec-2017) : *http ://rrc.cvc.uab.es/ ?ch =2&com=evaluation&task=4*.

Abstract

Scene texts have been attracting increasing interest in recent years as witnessed by a large number of applications such as car licence plate recognition systems, navigation systems, self-driving cars based on traffic sign, and so on. In this research, we tackle challenges of designing robust and reliable automatic scene text reading systems. Two major steps of the system as a scene text localization and a scene text recognition have been studied and novel algorithms have been developed to address them.

Our works are based on the observation that providing primary scene text regions which have high probability of being texts is very important for localizing and recognizing texts in scenes. This factor can influence both accuracy and efficiency of detection and recognition systems. Inspired by successes of object proposal researches in general object detection and recognition, two state-of-the-art scene text proposal techniques have been proposed, namely Text-Edge-Box (TEB) and Max-Pooling Text Proposal (MPT). In the TEB, proposed bottom-up features, which are extracted from binary Canny edge maps, are used to group edge connected components into proposals and score them. In the MPT technique, a novel grouping solution is proposed as inspired by the max-pooling idea. Different from existing grouping techniques, it does not rely on any text specific heuristic rules and thresholds for providing grouping decisions. Based on our proposed scene text proposal techniques, we designed an end-to-end scene text reading system by integrating proposals with state-of-the-art scene text recognition models, where a false positive proposals suppression and a word recognition can be processed concurrently. Furthermore, we developed an assisted scene text searching system by building a web-page user interface on top of the proposed end-to-end system. The system can be accessed by any smart device at the link : *dinh.ubismart.org:27790.*

Experiments on various public scene text datasets show that the proposed scene text proposal techniques outperform other state-of-the-art scene text proposals under different evaluation frameworks. The designed end-to-end systems also outperforms other scene-text-proposal based end-to-end systems and are competitive to other systems as presented in the robust reading competition community. It achieves the fifth position in the champion list (Dec-2017). *http ://rrc.cvc.uab.es/ ?ch=2&com =evaluation&task=4.*

Keywords : Scene text proposal, scene text detection, end-to-end scene text reading, word spotting system, scene text searching application, max pooling scene text proposal, text edge boxes, YoLo based scene text proposal

Publications

Conference papers :

1. Text-Edge-Box : An Object Proposal Approach for Scene Texts Localization, **Dinh NV**, Shijian L, Nizar O, Mounir M, *IEEE Winter Conference on Applications and Computer Vision*, IEEE-WACV-2017, USA, Mar-2017,
DOI : 10.1109/WACV.2017.149

2. Max-Pooling based Scene Text Proposal for Scene Text Detection, **Dinh, NV**, Shijian L, Bai X, Nizar O, Mounir M, *IEEE International Conference on Document Analysis and Recognitions*, IEEE-ICDAR-2017, Kyoto, Japan, Nov-2017,
DOI : 10.1109/ICDAR.2017.213

Journal papers

3. Max-pooling based Scene Text Proposal for Scene Text Reading System, **Dinh, NV**, Shijian L, Tian SX, Nizar O, Mounir M, *Journal of Pattern Recognition, Elsevier*, IF-4.9 (**Under revision**)

Contents

List of figures

List of tables

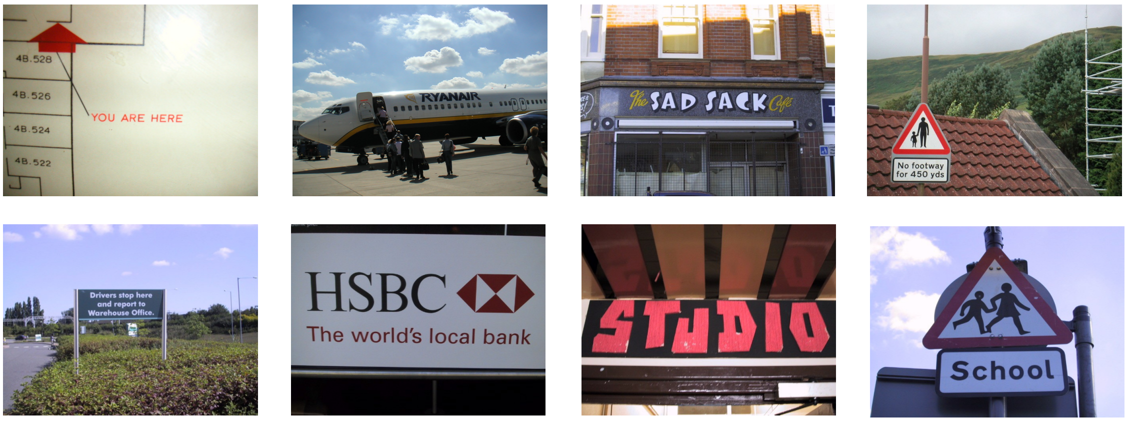
Chapter 1

Introduction

Contents

1.1 Scene texts and applications

Text is an effective tool in transferring information among people and between human and machines. Nowadays, it is worldly used for marking and describing many aspects in environments such as objects, events, shop panels, traffic signs, house plate numbers and so on, which are shown in Figure 1.1. Text information can be used for capturing rich semantic contexts for situational awareness applications such as a scene understanding, an action recognition, a navigation, and further



(a) Semantic information (b) Branch name (c) Shop panel (d) Traffic sign

FIGURE 1.1 – Some examples of texts in scenes. They can contain (a) semantic information, (b) branch names, (c) shop names, (d) traffic information.

for analysis purposes. Reading scene texts therefore becomes an increasing interest research topic. A lot of datasets and competitions in the scene text reading research have been announced [1, 2, 3, 4]. Many productive applications have adopted scene text reading systems in practice. Google Translation [5] and Microsoft Translator [6] are two very popular applications which translate detected scene texts from one language to another language. Orcam [7] is a recent application which is developed to assist visual impairment people in reading magazines, mails, and different texts in environment, such as texts in notification boards, street names. This product is still in a development phase and a lot of other visual based functions are targeted to be integrated, for example a general object recognition, a face identification. Scene text reading is also considered to adopt for a navigation application [8], a street house number reader [9], a traffic sign recognition [10], and a license plate recognition [11]. The Figure 1.2 shows some commercialized products which have been developed base on scene text reading systems.

Even though there are many scene text reading based applications released, these applications are only reliable in fine arrangement of camera holders and standard text fonts. For example, the licence plate reader based car tracking system performs well only with standard fonts assigned to licence plate numbers. The OrCam is developed to mainly support visual impairment patients in reading printed texts such

(a) Google Trans (b) Microsoft Translator (c) Licence Plate Reader

(d) Visual Aid Camera

FIGURE 1.2 – Examples of applications have been developed base on scene text reading systems including scene text translators (a,b), a licence plate reader based car tracking system (c), a visual aid camera (d). The applications a and b execute on smart devices as smart phones and tablets, while the applications c and d require specific hardware platforms.

as texts in magazines, menus. The Google Translate and the Microsoft Translator applications fail to read unpopular text fonts and oriented texts, as evaluated in Chapter 5.2. Reading scene texts therefore is still a high challenging topic and attracts a lot of researchers to participate in developing more reliable and precise systems.

1.2 Challenges

Different from scanned document texts which are in fine alignment, texts in scenes are more diverse in appearance due to variants of fonts, sizes, perspectives, colours, orientations and so on. In addition, their appearance is also distorted under various environmental impacts, such as light, haze, motion, illumination.

There is no standard text appearance in scene images. It could be printed texts as texts in panels, information boards, posters (Figure 1.3(a, b, c)) which could be detected by traditional OCR techniques [12]. On the other hand, text forms could be modified due to decoration purposes, for example texts in restaurant and shop panels

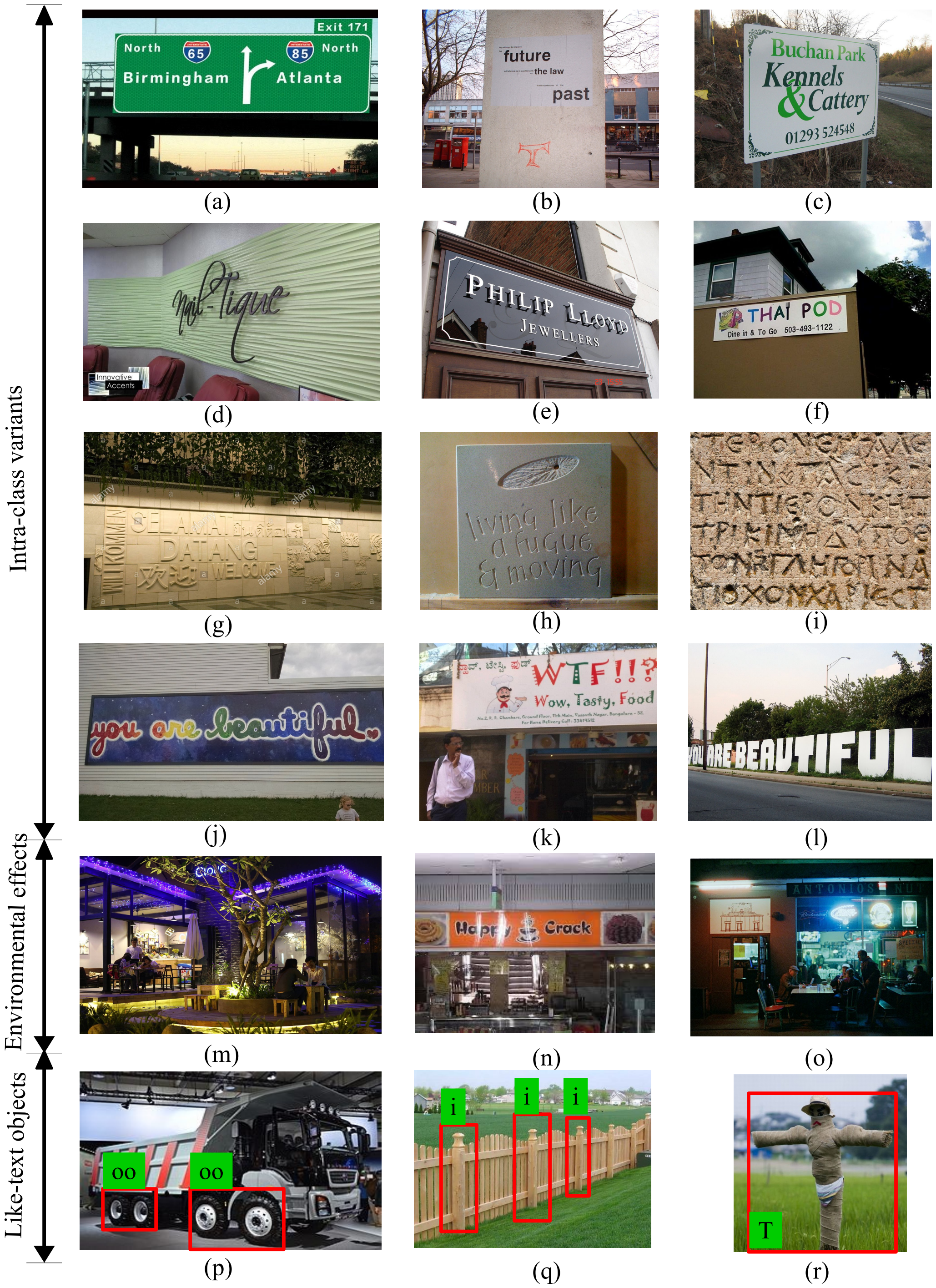


FIGURE 1.3 – Detecting and recognizing scene texts challenges including intra-class variation due to text fonts, text size, illumination, color, occlusion, and so on (the first five rows) and extra-class variance due to a lot of like-text objects in scenes (the last row)

(Figure 1.3(d, e, f)). Text appearance is also degraded because of material surfaces and written methods, as shown in Figure 1.3(g, h, i) where texts were sculptured and carved on rocks and stones. It can also be in different sizes (Figure 1.3 k), different perspectives (Figure 1.3 l) and colors (Figure 1.3 j). Even though texts have been printed in standard fonts, styles, colours, their appearance could be diverse under environment effects or uneven capturing image conditions. For examples, it could be occluded by other objects (Figure 1.3 m), blurred by underexposed condition (Figure 1.3 o) and illuminated by bright light (Figure 1.3 n).

Not only intra-class variants, reading scene text systems also have to deal with extra-class variants where scene images contain a lot of like-text objects. Systems have to clarify them to avoid false positive detections. For example, vehicle wheels, balloons, rings could be misclassified as character 'O' or number zero (Figure 1.3 p). Poles are similar to character 'l', character 'i' or character 't' (Figure 1.3 q-r).

In normal usages, standard fonts printed in posters, dashboards, menus, panels are mostly considered. Recent developed applications such as Google translator, Microsoft translator and so on can solve this challenge properly even though texts are in difference perspective. In fact, reading texts algorithms which developed for scanned documents [12] also can handle. However, it requires users' experiences in handling their cameras for capturing quantitative pictures. In order to be independent from this requirement, researchers in robust reading community are moving forward to improve their systems so that they can perform properly on more challenging cases.

1.3 Scene text processing systems

In past two decades, many scene text reading/spotting systems have been proposed, observed by a major number of review reports [13, 14, 15, 16]. Traditionally, the system consists of two main tasks including a scene text detection and a scene text recognition. The scene text detection task aims to localize texts in scene images. The scene text recognition task targets to recognize actual texts in located regions. Detected regions were usually seen as parts of texts, characters, words, and text lines. In recent years, detected regions are more considered as word regions for boosting up scene text reading/spotting systems' performance because the word-level region is a standard input of recent developed recognition models. Each task has to tackle different difficulties and raises independent competitions as observed from recent competitions [17, 18]. Scene text proposal, which is inspired by success of object proposal in detecting non-labelled objects, has been proposed to substitute for scene text detection. In fact, it provides a larger chance of localizing different texts variants in scene images. A new scheme of scene text reading/spotting systems therefore is proposed including a scene text proposal and a scene text recognition, and it has been adopted in recent state-of-the-art scene text reading/spotting systems [19, 20, 21, 22].

In this section, we are going to overview a major number of recent techniques and systems, which have been developed to address different reading-scene-text challenges, including scene text proposal techniques, scene text detection techniques, scene text recognition techniques and automatic scene text reading/spotting systems.

Scene text proposal techniques

The scene text proposal aims to generate region proposals with high recall. Despite the cost of a large number of false positive detections, it has a contribution toward reducing dramatically search spaces compared to sliding window. It's often evaluated according to the recall rate as well as the number of needed proposals - typically the smaller the better at a similar recall level [23]. False positive scene text proposals are eliminated by either a text/nontext classifier [24, 21] or a scene text recognition model [19, 25] in end-to-end scene text reading systems.

Different scene text proposal approaches have been explored. One widely adopted approach combines generic object proposal techniques with text-specific features for a scene text proposal generation. For example, EdgeBoxes [26] is combined with Aggregate Channel Features (ACF) and an AdaBoost classifier to search for text regions in [20]. In [19], Selective Search [27] framework is adapted for scene text proposal, where Maximal Stable External Regions (MSER) [28] is used for generating atomic regions which are later grouped together to form dendrogram based on a vast number of region features : a mean gray value of a region, a mean gray value of intermediate outer boundary, a region major axis, a mean stroke width, a mean of gradient magnitude at region borders and their coordinators. A text-specific symmetry feature is explored by using symmetry filters in [24] to search for text line proposals directly, where false positive text lines are removed by training a CNN classifier. Deep features have also been used for scene text proposal due to its superior performance and efficiency on general object detection and object proposal in recent years. For example, inception layers are built on top of the last convolution layer of the VGG16 for generating text proposal candidates in [21]. The Region Proposal Network (RPN) in Faster R-CNN [29] structure is adopted for scene text proposal generation in [22, 30]. In [31], a class activation map (CAM-conv) has been proposed and built on top of the convolutional state-5 of the VGG16 network, which is generated by using global average pooling and spatial pyramid pooling.

Scene text detection techniques

A large number of scene text detection techniques have been reported in the literature. Sliding window has been widely adopted [32, 33, 34, 35] in which scene texts are simply localized by scanning over image space by searching windows with different aspect ratios and sizes. These windows are classified into text or non-text by classifiers such as a real AdaBoost, fast cascade booting algorithms, the K-mean cluttering. In these approaches, selected regions are usually considered as characters which are later grouped together into words and text lines. However, sliding window is a low efficient technique because it uses a huge amount of windows in different sizes and aspect ratios to address large varieties of text appearances in scene images.

Region based techniques have been proposed to overcome an inefficiency constraint of the sliding window approach. The Maximal Stable External Regions (MSER) has been widely used for scene text detection due to its sensitivity to homogeneous color/gray regions which are usually considered as scene text foregrounds. Segmented regions are grouped together into word candidates, and false positive ones are eliminated by text/non-text classifiers [36, 37]. On the other hand, segmented regions can be classified into text or nontext class before being grouped into words and text lines [38, 39]. In some researches, MSER is substituted by the External Regions (ER) that is more robust to blur, illumination [40, 41]. Various hand-craft text-specific features have also been extensively investigated. Stroke Width Transform (SWT) [42] is proposed based on the hypothesis that characters in the same word have the same stroke width. The SWT output is a stroke image having the same size with an input image and contains stroke width associated with pixels. Pixels having the same stroke are grouped into characters and then text lines. Text specific heuristic rules are employed for filtering out false positive candidates. Stroke Feature Transform (SFT) [43] is a variant of the SWT. It integrates SWT with two additional cues as a color uniformity and a local relationship of edge pixels during a stroke width search process. In [44], three edge based text specific features are designed to score edge connected components. Non-text connected components are coarsely eliminated by a threshold of 0.5 and remaining connected components are grouped into word candidates. False positive words are filtered out by a trained support vector regression. Stroke End Keypoints (SEK) and Stroke Bend Keypoints (SBK) are proposed in [45]. They are designed to search for terminals or corners of objects which are high-frequency features of text. Gray values of pixels at these key points are used to calculate local MSER thresholds which are used to segment images into MSER regions. Convolution Neural Networks (CNNs) are also trained for atomic regions initialization. In [46], a CNN network is deployed to generate three different maps including a text region map, a character map and a linking orientation map, on which scene text bounding boxes are produced. On the other hand, a pre-trained network is only used to provide scene text region maps on which atomic text regions are extracted. MSER and heuristic rules are adopted for grouping and splitting atomic regions into words [47, 48].

Different from CNN models trained for atomic regions generation, several other CNN models have been developed for detecting text bounding boxes directly, including horizontal and quadrilateral shapes. For example, the DeepText [21] adopts the VGG16 convolutional layers [49] for deep features extraction and inception layers for horizontal bounding boxes predictions including boxes' top-left corners coordinates (x,y) and their heights (h) and widths (w). Predicted boxes are then fed into last convolution layers of the network for eliminating false position ones. The TextBox [50] adapts the Single Shot Multiboxes Detector (SSD) [51] architecture. Text bounding boxes are predicted from outputs of different intermediate convolution layers to address the multi-scale texts problem. In TextBox, they modified anchor boxes aspect ratios for predicting better bounding boxes which overlap ground truth boxes with higher IoU due to original anchor initialization only for general objects. Quadrilateral anchor boxes have also been proposed for detecting tighter scene text boxes [52]. In addition, a direct regression solution has also been proposed [53] to remove the hand-crafted anchor boxes, which are usually initialized with certain number of boxes and aspect ratios in traditional CNN based regression models [21, 50, 52]. In the proposed idea, bounding box coordinates are regressed from a single point such as a center of a studying region. Different other CNN based detection approaches have also been explored. For example, some works adopt a bottom-up approach that first detect characters and then group them to words or text lines [54, 22, 55]. Some systems instead define a text boundary class for pixel-level scene text detection [56, 57]. In addition, weakly supervised and semi-supervised learning approach [58] have also been studied to address the image annotation constraint [59].

Scene text recognition techniques

Scene text recognition assumes that text regions have been detected correctly, and it aims to read actual words in located regions. Traditional recognition models usually localize and recognize individual characters consequentially, and then sort them in proper orders to provide final recognition results. In recent approaches, convolution neural network (CNN) are employed. Almost CNN models consider word images as characters sequences [60, 61, 62, 63, 64, 65]. On the other hand, Jaderberg's model [66] converts a recognition task into a classification task and considers each word as a class.

In [67], strokelets feature for character identification is proposed and a combination between Random Forest model and feature vectors based on Bag of Strokelets as well as Bag of HoG is implemented for character recognition. Recognized characters are later linked together from left to right into words. This traditional method is improved by integrating with top-down and button-up cues and the Conditional Random Field model [61, 68]. The button-up cues are texture features and the top-down cues are usually initialized lexicons. In [61], potential character regions are pointed by image binarization and false positive regions are removed by heuristic rules. In the [68], a combination of sliding windows and HoG feature vectors is implemented for a potential character regions generation and Support Vector Machine (SVM) model is employed for a recognition. In [69], Conditional Random Forest model is used as a language model for eliminating strange detected characters by minimizing its cost function, where potential characters are detected base on their proposed part-based tree-structure.

In [60, 61, 62, 63, 64, 65], convolution neural networks are trained to treat words like characters sequences. They convert input images into sequences of feature vectors which are then decoded to transcripts. In [60], they implemented sliding window integrating with the HoG features for generating features sequences and Long Short-Term Memory (LSTM) model for transforming features sequences into labels sequences which are later converted to transcripts. In order to remove hand-craft features as the HoG which fails to deal with extremely blur or fragmented texts, deep feature extraction models have been implemented. Feature maps are produced by applying deep features extraction models either on sliding windows shifting along image width [70] or on whole image space [61]. In [62], an end-to-end trainable network has also been proposed by linking a convolution neural network, a LSTM, and a transcription layer into one network. Therefore, its feature map generation and transcript decoder can be trained concurrently. Weird text forms are considered in the [63, 64]. In [63], a Spatial Transformer Network (STN) is employed to convert non-horizontal texts into horizontal ones before feeding them into a Sequence Recognition Network (SRN), which consists of two combinations : one is a combination between the CNN with the LSTM for sequence encoder and one is a combination between the GRU and the attention structure for a sequence recognition. In contrast, a model in [64] does not employ STN. Instead, a recurrent network is applied from left to right on sequences of attended regions generated by CNN networks for a word recognition. In [65], CNN network has been suggested to use for a sequence decoder instead of RNN. The target is to perform parallel operations and to be easier in training.

Automatic scene text reading/spotting systems

Automated scene text reading system aims to detect and recognize texts in scene images. Traditional framework consists of a scene text detection and a scene text recognition [71, 72, 73, 74, 75]. In recent year, scene text proposal is proposed to substitute a scene text detection step, largely due to the fact that it is able to locate more text regions as compared with the text detection [19, 20]. Convolution neural network based techniques are also proposed to replace each single steps in the tradition framework by CNN models [34, 76, 66, 50] or develop single models handling both scene text detection and scene text recognition [77].

A popular system is a Google-Translation [71] which performs end-to-end scene text reading by fusing a large number of techniques including three scene text detection methods, three scene text segmentation and grouping methods, deep neural network on Histogram of Oriented Gradient (HOG) features for character classification, and language models for post-processing. In [72], sliding window is combined with HOG feature extraction and Random Ferns Classifier to compute text saliency maps where words are extracted using External Regions (ER) and further re-scored using Support Vector Machine (SVM). In [73], Adaboost classifier and SVM are applied on text regions that are extracted using ER to localize scene texts which are further recognized under an Optical Character Recognition (OCR) framework. Similar approach was also adopted in [74], where Maximal Stable External Regions (MSER) instead of ER was implemented for scene text region localization. In [75], Stroke Width Transform [42] is adopted for scene text region detection and Random Forest is used for character recognition and words are further recognized by a component linking, a word partition and a dictionary based correction. In [19, 20], potential text regions are first localized using EdgeBox (EB) [26] or adapted simple selective search for scene text [19] and scene texts are further recognized using Jaderberg's scene text recognition model [78].

Quite a number of CNN based end-to-end scene text reading systems have been reported in recent years. In [34, 76], a CNN based character recognition model was developed where word information is extracted from text saliency map using sliding windows. The same framework has been implemented in [66], where a more robust end-to-end scene text reading system is developed by training a model handling three functions including a text and non-text classification, a case-insensitive characters recognition and a case-sensitive characters recognition. In [50], an advanced end-to-end scene text reading system is designed where the Single Shot Multi-boxes Detector (SSD) was employed for scene text detection and a scene text recognition model proposed in [62] was adopted for recognition. End-to-end trainable scene text reading system has also been proposed where it can provide texts location and text transcription concurrently by a single model [77].

1.4 Objectives

This thesis aims to develop state-of-the-art automatic scene text reading systems which can be integrated into relevant applications supporting visual impairment people, especially elderly in their outdoor activities in the future phase of our project. For developing, we mainly focus on challenges of scene text localizations when there is not any prior knowledge about texts known in advance, for example text font, text size, text color. The first goal is to provide high-quality proposal regions. The better performance of this step can support the entire system in providing more precise text localization and recognition. Based on developed scene text proposal techniques, our second goal is to develop a state-of-the art automatic scene text reading system and deploy a relevant application.

In scene text proposal techniques, the number of proposals is an essential parameter that can affect automatic reading systems' efficiency. A huge number of regions will burden further system steps because they have to iterate a lot of times along proposals. In our approach, we are intent on limiting the number of proposals provided by our proposed techniques while still gaining state-of-the-art recall rate by ranking our proposals and collecting top n proposals in the ranked list. In the first approach, we focussed on heuristic rules which gain enormous attention of researchers in developing scene text detection techniques. We did study on these rules and proposed two effective heuristic features for evaluating edge pixels, grouping edge connected components into proposals and scoring them. In the second approach, we are intent on being independent from text specific heuristic rules by proposing a max-pooling based grouping strategy. The proposed technique groups edge connected components into proposals naturally based on their internal distances. Effectiveness of the proposed features are indicated in comparisons with other state-of-the-art techniques.

In order to develop automatic scene text reading systems from scene text proposal, scene text classification and scene text recognition models are usually adopted to remove false positive proposals and recognize their actual texts. There is an observation that we can lean on recognition models for handling a classification task. For example we can use a threshold of recognition scores for eliminating false positive proposals. This idea is also implicitly embedded in deep-learning based object detection systems [79, 80], where proposals generated by a region proposal network (RPN) are scored by recognition layers and low recognition confident score ones are eliminated. In this thesis, we did evaluations on state-of-the-art scene text recognition model developed by Jardeberg [78] and integrated it on top of our proposed scene text proposals as a classification and recognition model for designing our automated reading system.

In scanned document analysis, there are numerous applications developed to support people in reading such as : a searching-word, a translating-word, an automatic pronouncing-word. Developing an automatic scene text reading system will bring those effective tools into our daily life. Orcam [7] is an application that supports visual impairment patients in reading articles by interacting with users fingers. Google translation [5] is proposed to detect words in scene images and translate them into expected languages. Inspired from the searching-word tool, we prefer to build the first application supporting people in searching their keywords in scene images. It could help people by saving their time in searching specific items in menus, aware environment around them based on expected words and so on.

1.5 Contributions

In this thesis, three major contributions are produced : scene text proposal techniques, an automatic scene text reading system and a scene text searching application.

Scene text proposal techniques

We have proposed two state-of-the-art scene text proposal techniques that provide high-quality text proposals. They are developed from binary edge maps, gradient maps and oriented gradient maps generated by the Canny edge detector. In the scene text proposal generation step, two edge based features are determined to clarify text edges from other edges in binary edge maps. Two grouping algorithms are proposed to merge effectively connected components into proposals. The first algorithm is developed from text specific heuristic relationship rules between connected components. The second algorithm provides a natural dendrogram grouping based on internal distances between connected components, which is inspired by a process of a max-pooling layer. The proposal scoring/ranking problem is also addressed. Two proposed scoring functions are developed from edge based features. It supports the proposed techniques in providing state-of-the-art performances while a number of proposals is limited.

An automatic scene text reading system

Our automatic scene text reading system has been developed by combining our scene text proposal techniques and existing scene text recognition models. False positive proposals are eliminated based on a threshold of recognition confidence score, non maximal suppression and provided lexicon lists. Evaluated by the evaluation framework published in the ICDAR competition, the proposed system outperforms other scene text proposal based systems and be competitive to other state-of-the-art scene text reading systems, including systems are developed based on deep learning frameworks, observed by its performances in the ICDAR competition website (http://rrc.cvc.uab.es/?ch=2&com=evaluation&task=4) under the name MPT_sys.

A scene text searching application

In order to evaluate reliability of the proposed automated reading system, we developed a demo of a scene text searching application and it is applied on real images captured by users' cameras. The demo has been developed based on web platform, and it can be accessed by several devices such as smart phones, tablets, laptops. Users can provide searching keywords by using microphones or keyboards. The goal of this demo is to provide a reliable and accurate scene text searching system. It has been compared with the Google Trans in term of finding keywords in scene based on our evaluation scheme. In further phase of this project, we expect to integrate this system into more relevant applications interacting with various services such as navigation, circumstance alert, and so on.

1.6 Evaluation criteria

In this research, we developed two scene text proposal techniques and an automatic scene text reading system. Two evaluation frameworks therefore have been adopted in order to evaluate these techniques and system. For evaluating scene text proposal techniques, we utilize a framework which is widely used in an object proposal evaluation [26, 19, 23]. For evaluating an automatic scene text reading system, we follow a framework provided by the robust reading competition community [17].

1.6.1 Datasets

The ICDAR dataset and the Street View Text dataset : These two datasets are widely used in a community working on developing scene text analysis systems and they are utilized in this research evaluation. The ICDAR is the dataset provided by the International Conference on Document Analysis and Recognition (ICDAR) when they launch their robust reading competitions. This dataset was first announced in 2003 and rearranged in 2011, 2013, and 2015 [81, 2]. Another dataset is the Street View Text dataset (SVT) [3] which contains images from the google

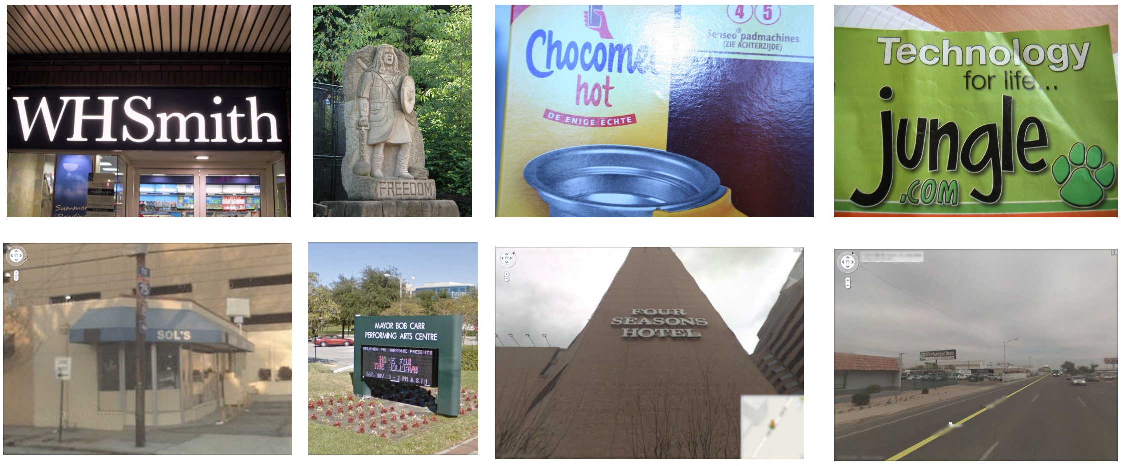


FIGURE 1.4 – Example images in the ICDAR2013 dataset (the first row) and the Street View Text dataset (the second row)

street view database. Images in both datasets are labelled by users with bounding boxes covering text areas and words within these areas. They have been designed for four different challenging competitions : a scene text localization, a scene text recognition, a scene text segmentation and an automated scene text reading system. Images are already separated into a training set and a testing set. The ICDAR2013 dataset contains 229 training images and 233 testing images, and the SVT dataset contains 101 training images and 249 testing images. Images are collected under multi-challenges conditions such as images influenced by blur, uneven light, different perspective or images with multi-variance in color, size, font, and so on, as illustrated in Figure 1.4.

In recent years, there are many other scene text datasets proposed and interested by research community. Each of them has been targeted for different purposes.

COCO Text :[18] The dataset is produced based on the MS COCO dataset, which contains images of complex everyday scenes. The images were not collected with text in mind and thus contain a broad variety of text instances including machine printed texts, hand written texts and others. English is the most common language in this dataset, it also covers Western script such as German, Spanish, French, an so on. Illegible terms of texts are also annotated for advance evaluations. The dataset contains 63686 images and is split into a training set including 43686 images and a validation set including 20000 images. It is designed for developing a scene text location, a scene text recognition and an automatic scene text reading system. Due to huge amount of training images, it is qualified for training convolutional neural network based systems.

ICDAR2015 Incidental Scene Text : This is one of datasets published in the ICDAR2015 competition [1, 17]. Its images were taken in a few months period in Singapore without user's careful prior actions for focussing texts, improving quality of frame. It contains 1670 images and 17548 annotation regions. Images are split into three sets : 1000 images in a training set, 500 images in a testing set and 170 images in an unpublished set. Non-Latin script, illegible words, less-than-three-character words are annotated as *do not care* and will be ignored during evaluation. This dataset is proposed for evaluating a scene text detection, a scene text recognition and an automatic scene text reading system. It provides three contextualizations including strong, weak, generic terms which are depend on length of lexicon as 100 words, all words in the testing set (1071), and 90k words respectively.

Synth Text[82] This dataset is generated by overlaying texts in different scene images (8000 scene images were extracted from Google Image Search and guaranteed that they do not contain texts in their contents). This dataset contains 800000 synthetic scene text images and each image has about ten word instances annotated with word bounding boxes and character bounding boxes. The dataset aims to support training convolutional neural network based systems.

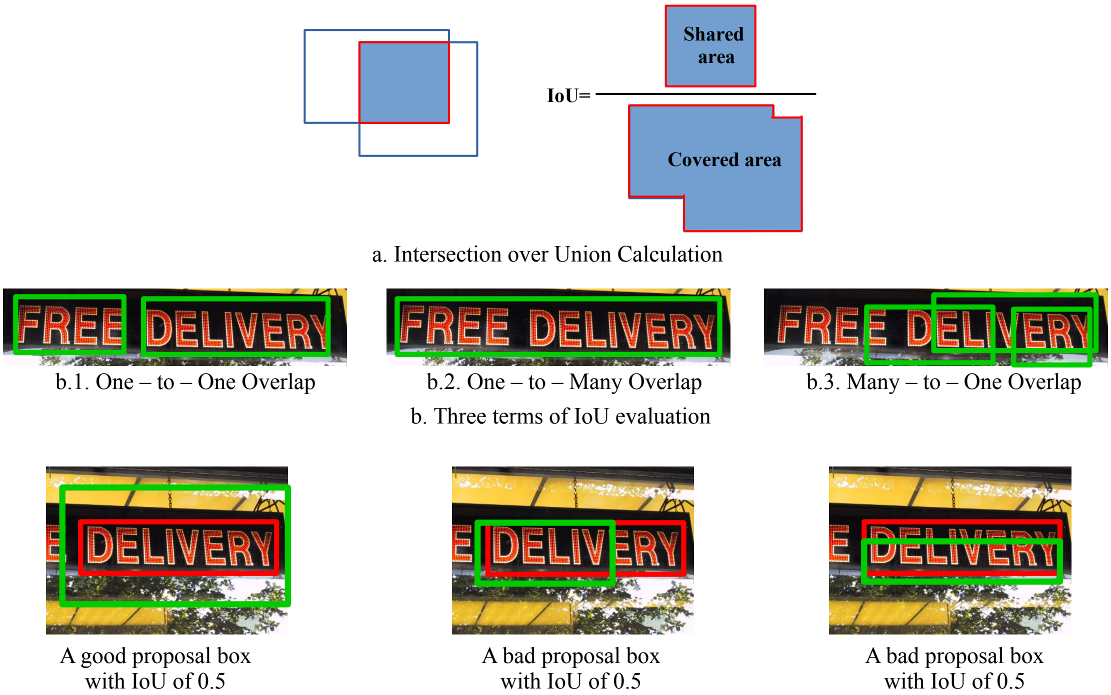
MSRA_TD500 :[83] This dataset is taken from indoor (office and mall) and outdoor (street) scenes using a pocket camera. In door images are mainly signs, doorplates and caution plates while outdoor images are mostly guide boards and billboards in complex background. Images are in high resolution that varies from 1296-by-864 to 1920-to-1280. Not only challenges of text variants in fonts, sizes, colours, orientations and so on but also a multi-languages problem are included. 500 scene text images have been split into a training set containing 300 images and a testing set containing the rest images.

1.6.2 Evaluation of scene text proposal techniques

There are three parameters needed to be measured when evaluating scene text proposal techniques : a recall rate, a number of proposals and processing time. The recall rate presents a technique ability to localize correctly texts in scene images. The better scene text proposal technique is one that can provide the higher recall rate using the smaller number of proposals and the faster process. In order to evaluate our technique as well as other state-of-the-art techniques, we adopted a general evaluation framework which is utilized widely to evaluated object proposal techniques, as are described in [23].

In order to decide whether text objects have been localized successfully, the intersection over union (IoU) has been applied. It measures a fraction between an overlapped area of two boxes and an entire area covered by them, as illustrated in the first row in Figure 1.5. The higher IoU value presents the better overlapping between two boxes, meaning that a proposal box is the more similar to a ground truth box. Note that the IoU in this evaluation considers only a case of one-to-one overlapping instead of cases of one-to-many or many-to-one overlapping, which are mentioned in the evaluation framework for scene text detection techniques [84] (the second row in Figure 1.5). It is because state-of-the-art scene text recognition models mostly focus on a word recognition. Good scene text proposals on many-to-one is useless for scene text recognition models and good scene text proposal techniques on one-to-many need post-processes in order to split text lines into words before feeding into recognition models. IoU of 0.5 is usually considered as threshold of an overlapping condition to decide whether objects are detected in the object detection system. On the other hand, higher IoU is expected for evaluating scene text objects, due to the fact that proposal boxes achieving IoU of 0.5 sometime are not good enough for scene text recognitions, for example, proposal boxes are shown in the third row in Figure 1.5. The first case is a good IoU-of-0.5 proposal and the rest cases in the right are poor IoU-of-0.5 proposals. Therefore, the higher IoU threshold will be utilized to evaluate and compare scene text proposal techniques in this thesis.

Performances of scene text proposal techniques are presented by recall rates under various situations generated by different combinations between a number of proposals and an IoU threshold. When a number of proposal is fixed, the IoU threshold is varied and vice versa. The better technique will provide the higher recall rate at every evaluation scenario, especially when the number of proposal is small and the IoU threshold is high. In addition, this evaluation framework considers an average number of proposals each technique needs to perform their performances. The better technique is one that requests the smaller number. There is a difference between our evaluation and the tradition object proposal evaluation framework that we limit the number of proposals at 2000 instead of 10k proposals as usual, which is mentioned in [79] due to trade off between system efficiency and system accuracy.



a. Intersection over Union Calculation

b.1. One – to – One Overlap

b.2. One – to – Many Overlap

b.3. Many – to – One Overlap

b. Three terms of IoU evaluation

A good proposal box with IoU of 0.5

A bad proposal box with IoU of 0.5

A bad proposal box with IoU of 0.5

c. Proposal boxes (green) with IoU of 0.5 overlaping with groundtruth box (red)

FIGURE 1.5 – Some features of the adopted scene text proposal evaluation framework including : (a) how Intersection over Union (IoU) is calculated, (b) the difference of one-to-one overlap to other types of overlaps and (c) examples of good and poor proposals with IoU of 0.5.

1.6.3 Evaluation of automatic scene text reading systems

The evaluation framework for an automated scene text reading system is adopted from the evaluation framework of the scene text reading competition community [17], which is embodied in [84]. There are three contextualizations (strong, weak and generic evaluation) and two kinds of systems (an end-to-end scene text reading and a scene text spotting) involved. In this evaluation, only words which are alphanumerical words and containing at least three characters are considers, others are ignored. Generally, a word is decided to be detected if there is at least one proposal box overlapping with its' ground truth box over IoU of 0.5 and it is recognized successfully.

Three contextualizations of an automatic scene text reading system are illustrated as a strong contextualization, a weak contextualization and a generic contextualization. They are clarified based on accompanied lexicon lists as a per-image vocabularies of 100 words, whole vocabularies in the testing set and a generic vocabularies of about 90k words, respectively. These lexicons are used to guide proposed automatic scene text reading system to filter out false positive proposals. In each contextualization, the F-measure value is calculated to evaluate system performance. The better system is one that achieves the higher F-measure value. This value is calculated based on the system recall and the system precision following the below equation :

$$F\_measure = 2\frac{Precision \cdot Recall}{Precision + Recall} \tag{1.1}$$

The end-to-end scene text reading system is more general than the scene text spotting system, where it considers alphanumerical words that consist of characters and numbers instead of only-character words on which the scene text spotting system focusses.

1.7 Thesis outline

This thesis goes through our contributions in developing scene text processing systems including scene text proposal techniques, an automatic scene text reading system and a scene text searching application. Chapter 3 focuses on two proposed text specific edge features which are used for clarifying text edge pixels from non-text edge pixels and scoring scene text proposals. These two features are major contributions of the proposed technique named as Text-Edge-Box (**TEB**). Chapter 4 concentrates on grouping-edges solutions which are used to merge single edges into proposals. The unique of this proposed grouping solution is that it does not rely on any text heuristic rules which are widely utilized by traditional scene text detection techniques. The grouping solution is inspired by a max-pooling process in the deep learning framework, and the technique proposed based on this grouping solution is named as max-pooling based scene text proposal (**MPT**). An automatic scene text reading system described in Chapter 5 is constructed by applying scene text recognition models on top of the proposed scene text proposal techniques for both a false positive proposals elimination and a scene text recognition. The proposed system provides state-of-the-art performances on the robust reading competition. Based on success of this system, a scene text searching application has been implemented and its performance is competitive to the commercialized product known as Google Translator in term of searching for specific words in scene images. In this thesis, we also go through state-of-the-art research works in developing scene text proposal techniques and automatic scene text reading systems. They are described and discussed in Chapter 2. Future works are discussed in Chapter 6. It includes an idea to employ the YoLo network for scene text proposal generation and an idea to generate quadrilateral bounding boxes for detected text regions which is a requirement for system outputs in the recent scene text datasets as the CoCo text dataset and the incidental scene text dataset. A relevant application aiming to support visual impairment patients and elderly in their outdoor activities is also described in this Chapter.

Chapter 2

State-of-the-art scene text proposal techniques and scene text reading systems

Contents

In this chapter, we are going to present in detail recent state-of-the-art research works in proposing scene text proposal techniques and automatic scene text reading systems which are also targeted goals of this thesis. Four scene text proposal techniques including the adaptive selective search for scene text proposal [19], the symmetry text line based technique [24], the deep texts [21] and the weakly supervised text attention network [31] as well as three automatic scene text reading systems including the TextBoxes [50], the DeepTextSpotter [77] and the Edgeboxes based scene text reading system [20] are described. Comparisons of our proposed techniques and systems to these research works are discussed in Chapters 3, 4, and 5.1.

2.1 Scene text proposal techniques

The general framework of scene text proposal techniques is inherited from object proposal techniques. It includes two main steps : a proposals generation and a proposals scoring/ranking. In this review section, we will project each state-of-the-art technique into this framework and clarify algorithms which they have used. Note that these scene text proposal techniques are designed to provide word level proposals, which is the standard input of recent scene text recognition models.

2.1.1 Adaptive selective search for scene text proposals

a. Proposal generation

This technique starts with the maximal stable external regions (MSER) algorithm for generating atomic regions and the single linkage clustering (SLC) algorithm for grouping generated atomic regions into clusters, which are later referred to proposals. At the first iteration of their grouping strategy, atomic regions are already considered as clusters. These clusters are then grouped together base on their similarity iteratively. Pairs of the most similar clusters are grouped first and pair of the least similar clusters are grouped at the end. In order to measure similarity among clusters, they defines a distance metric $d(r_a, r_b)$ which is calculated from seven similarity cues extracted from local pixels belonging to clusters and coordinates of cluster centres. A similarity feature $d^{(i)}(r_a, r_b)$ of a pair of two clusters a and b

base on a single similarity cue i^{th} is calculated using Equation 2.1 :

$$d^{(i)}(r_a, r_b) = (f^i(r_a) - f^i(r_b))^2 + \lambda(x_a - x_b)^2 + (y_a - y_b)^2 \tag{2.1}$$

The $f^i(r_a)$ refers to a similarity cue i^{th} of a cluster a. $[x_a, y_a]$ and $[x_b, y_b]$ are centre coordinates of cluster r_a and cluster r_b respectively. The λ is a horizontal priority parameter that controls a priority of horizontal grouping, and it is in a range of (0, 1).

For minimizing processing time, seven chosen similarity cues are all simple and low computational cost features, including two cues in the Lab colour space (a mean colour of pixels in the region (F_{lab}), and a mean colour of pixels in the immediate outer boundary of the region (B_{lab})), a mean intensity value (F), a mean intensity value in the immediate outer boundary of the region (B), a mean value of the distance transformed connected component mask generated by stroke width transform (S), a mean of gradient magnitude on the bounder of the region (G), a major axis of their fitting ellipse (D).

In order to find proper proposals for localizing words, several variants of the proposed technique have been evaluated, using different complementary similarity cues, different colour channels and different spatial pyramid levels. In total, there are 84 technique variants. The optimal one has been found at a combination of $DFBGS$ with two different image scales (1 and 0.5) in three colour channels as RGB.

b. Proposal scoring/ranking

The Real Adaboost classifier has been used to score generated proposals due to low computational cost. Feature vectors of proposals are extracted based on both individual similarity cues $f^i(r_a)$ and proposals' bounding boxes properties .

At each type of similarity cue, a feature is calculated using Equation 2.2.

$$F^i(G) = \frac{\sigma^i}{\mu^i} \tag{2.2}$$

where σ^i and μ^i are respectively the mean and standard deviation of the i^{th} similarity cue in a particular group $G, \{f^i(r) : r \in G\}$. For example, the optimal variant of this technique includes 30 (=5*2*3) similarity cues, so its proposal's feature vector contains 30 dimensions.

Proposal bounding boxes based feature dimensions are extracted base on relationship between the bounding box of the group constituent regions and the bounding box enclose only the regions' centres. A set of five simple features are originated from calculating : (1) the ratio between the areas of both bounding boxes, (2) the ratio between their widths, (3) the ratio between their heights, (4) the ratio between the difference of their left-most x coordinates and the difference of their right-most x coordinates, (5) the ratio between the difference of their top y coordinates and the difference of their bottom y coordinates.

The Real Adaboost classifier has been trained on training images in the ICDAR2013 dataset. In testing phase, the Logistic Correlation is applied on the raw classifier's outputs to obtain a probability value for each proposal. The output of this scoring algorithm is normalized in the range [0 ;1]. Proposals with their confident scores are sorted in the descending order, where the front-most proposals of the list have the highest probability of being texts.

2.1.2 Symmetry text line

This technique is developed base on a hypothesis as text regions usually exhibit high self-similarity to itself and strong contrast to its local background. Symmetry axes are first extracted via the proposed symmetry detector. In order to deal with text size variants, multi-scale input images are implemented. Base on detected text line axes, bounding boxes of text line level proposals are estimated and later split into word level proposals using an internal distance threshold. In this technique, a proposal scoring/ranking step is not implemented. All generated proposals are fed into a word classifier to finalize their scene text detection system.

The proposed symmetry detector consists of a features extraction and a random forest based symmetry axis classifier. In the features extraction, a symmetry template is proposed. It consists of four rectangles with equal size of $s \times 4s$, denoted by RT(a top rectangle), RB(a bottom rectangle), RMT(a top middle rectangle), and RMB(a bottom middle rectangle) as shown in Figure 2.1 The height of rectangle "s"

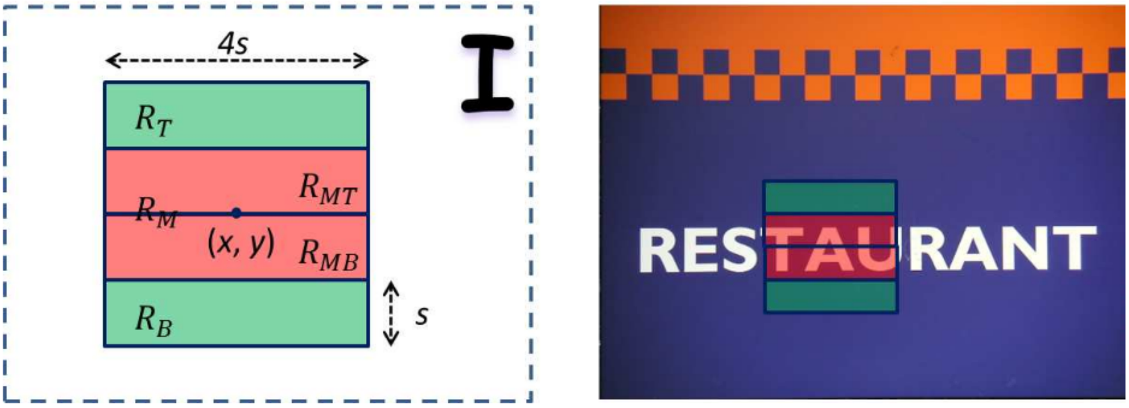


FIGURE 2.1 – The proposed symmetry template consists of four rectangles with equal size of $s \times 4s$, denoted by RT(a top rectangle), RB(a bottom rectangle), RMT(a top middle rectangle) and RMB(a bottom middle rectangle)

is defined as a template scale. Two types of features are employed as a symmetry feature and an appearance feature.

- **Symmetry features** consist of two groups of features as a self-similarity feature and a contrast feature. The self-similarity is developed based on a hypothesis as adjacent characters bear similar color and structure while the contract feature is inspired by a high dissimilarity of text foreground to background. The self-similarity type feature is calculated based on pixels in the two middle rectangles (RMT and RMB). The contrast feature type includes two features as a top-contrast feature calculated from a RT and RMT, and a bottom-contrast feature calculated from a RB and a RMB. Following three equations : 2.3 for the self-similarity feature and two equations 2.4 and 2.5 for two types of contrast features from symmetry templates, respectively.

$$S^c_{x,y} = X^2(h^c_{x,y}(RMT), h^c_{x,y}(RMB)) \tag{2.3}$$

$$Ct^c_{x,y} = X^2(h^c_{x,y}(RT), h^c_{x,y}(RMT)) \tag{2.4}$$

$$Cb^c_{x,y} = X^2(h^c_{x,y}(RB), h^c_{x,y}(RMB)) \tag{2.5}$$

where, X^2 is a distance function, $h^c_{x,y}(RP)$ with $P \in (T, B, MT, MB)$ denotes the histogram of a low feature cue c in the corresponding regions. There are four kinds of low cues adopted as brightness, color, texture, and gradient. The brightness and two color cues are extracted from Lab color space. The texture cue is extracted using implemented textons and the gradient cue is computed from pixels' gradient magnitudes. In total, there are (3*5=) 15 symmetry features extracted from one scale symmetry template at the location (x, y).

- **Appearance feature** is extracted by applying the Local Binary Pattern (LBP) feature [85] on the middle rectangles (RMT+RMB). It provides 59 features

All features are concatenated into a 74-dimension feature vector (=15+59) representing the pixel at location (x, y) in where the corresponding template is centred. Pixels are then classified into two classes as a symmetry axis pixel and a non symmetry axis pixel by using the Random Forest classifier. It forms a symmetry probability map. Due to implementing multi-scale input images, multiple symmetry probability maps are provided.

From a symmetry probability map, symmetry pixels are grouped together to produce symmetry axis fragments if distances between them are less than three pixels. Symmetry axis fragments are then merged together into text line proposals if they are satisfied following two geometric constraints :

- **Angular different constraint :** Two fragments are able to grouped together if their angular difference is less than $\frac{\pi}{16}$, and their angular difference is calculated by Equation 2.6, where, A, B are two fragment, $\phi(A), \phi(B)$ are their direction.

$$\Phi(A, B) = |\phi(A) - \phi(B)|, (\phi(A), \phi(B) \in (-\frac{\pi}{2}, \frac{\pi}{2}) \tag{2.6}$$

- **Distance constraint :** Two fragments are able to grouped together if the minimal distance between them is less than maximal height of the two fragment heights where fragment height is the scale of the corresponding template. The minimal distance between two fragments is computed by Equation 2.7, where p, q are two given points in the two fragments A, B respectively, and an operation $||.||$ is a distance calculation between two points.

$$D(A, B) = min(||p - q||), p \in A, q \in B \tag{2.7}$$

A bounding box of a text line proposal is estimated as follow : The width is determined by the horizontal axis coordinates of the axis pixels belong to the text line proposal and the height is the scale of the corresponding template.

2.1.3 DeepTexts : Region proposal network based scene text proposal

This is a deep learning based scene text detection network resembling to the Fast R-CNN network [80]. A scene text proposal generation and a scene text proposal scoring/ranking are process concurrently. Scene text proposals are first generated by their developed region proposal network named as Inception-RPN. 300-top proposals are then passed to a text detection network for clarifying text proposals from non-text ones. In this review, the Inception-RPN network is concentrated due to relating to a scene text proposal generation task what we are focussing on. The structure of their scene text proposal generation is depicted in Figure 2.2.

The Inception RPN is applied on the top of the convolution feature map (Conv5_3) in the VGG16 model [49]. It is inspired by the Inception block in the GoogleLeNet [86], and it consists of a 3×3 convolution layer, a 5×5 convolution layer and a 3×3 max pooling layer. A 1×1 convolution layer is deployed on top of the 3×3 max pooling layer for dimension reduction. The output of these layers are concatenated into 640-dimension feature vectors which are then fed into two sibling fully connected layers to predict text confidence scores (a classification layer) and proposal regions (a regression layer). Note that there are 24 anchor boxes designed with four scales (32, 48, 64, and 80) and six aspect ratios (as 0.2, 0.5, 0.8, 1, 1.2, and 1.5) at each point in the final feature map. The regression layer is trained to regress these anchor boxes to find the best proposal bounding boxes. Proposal boxes are then pass through non maximal suppression (NMS) to provide a final set of proposals. In scene text proposal evaluation, only 500 proposals are kept from the provided list.

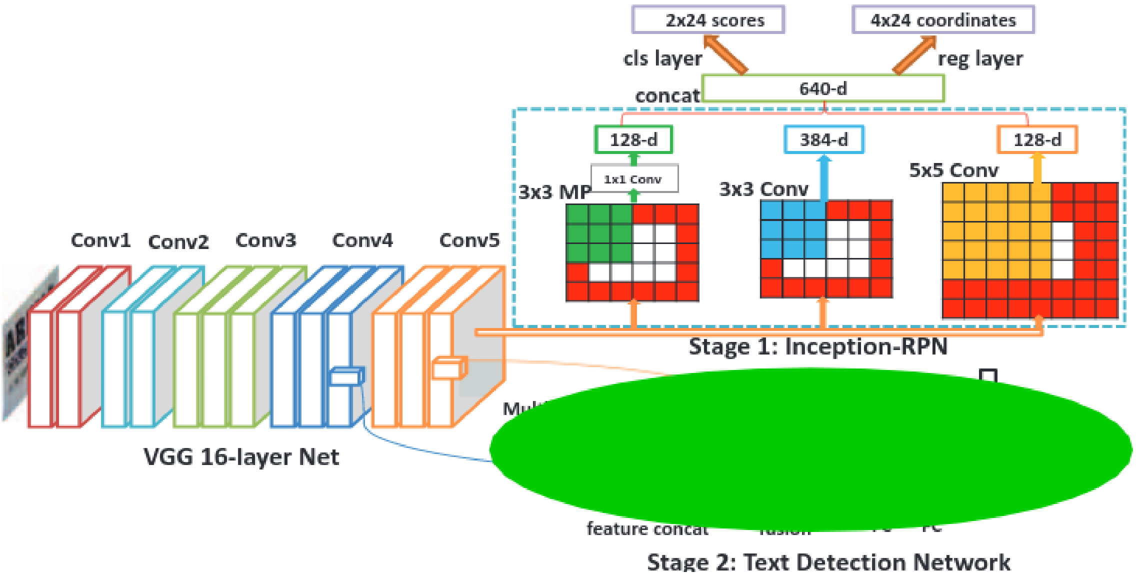


FIGURE 2.2 – Structure of a scene text proposal generation network (Inception-RPN) in the DeepText network which is developed to detect texts in scene images

The same as existing RPN models, the Inception-RPN model is trained using a multi-tasks loss function which is presented in Equation 2.8.

$$L(p, p^*, t, t^*) = L_{cls}(p, p^*) + \lambda L_{reg}(t, t^*) \tag{2.8}$$

where classification loss L_{cls} is a soft-max loss and p and p^* are given as the predicted and true labels, respectively. Regression loss L_{reg} applies the smooth-L1 loss. The $t = \{t_x, t_y, t_w, t_h\}$ and $t^* = \{t_x^*, t_y^*, t_w^*, t_h^*\}$ stand for predicted and ground-truth bounding-box regression, respectively. The t^* is encoded as below :

$$t_x^* = \frac{(G_x - P_x)}{P_w}, t_y^* = \frac{(G_y - P_y)}{P_h}, t_w^* = log(\frac{G_w}{P_w}), t_h^* = log(\frac{G_h}{P_h}) \tag{2.9}$$

The $P = (P_x, P_y, P_w, P_h)$ and $G = (G_x, G_y, G_w, G_h)$ are centre coordinates (x, y), width (w) and height (h) of an anchor P and a ground truth G, respectively. The λ is a loss balance parameter and it is set to 3 in the training process of the Inception-RPN.

2.1.4 Weakly supervised text attention network

This technique adopts the deep learning based framework to produce text confidence score maps in which text pixels obtains higher scores than background pixels. Text confidence score maps are then binarized by using a score threshold. Text regions which have superior confidence scores are segmented coarsely. On each segmented region, the scene text proposal generation framework of the adaptive selective search for scene text proposal technique (TP) [19] is employed to refine segmented regions into scene text proposals. The maximal stable external regions (MSER) is applied to generate MSER regions which are grouped together into scene text proposals by applying the single linkage criterion (SLC) and different complementary distance metrics. Scene text proposals are then scored and ranked by a trained Real Adaboost classifier. In fact, this technique can be considered as an upgraded version of the TP technique since the TP framework is applied on segmented text regions instead of whole image space as the original set-up.

The major contribution of this technique is the text attention network which provides text confidence score maps. It is similar to the discriminate localization network [87] with some of modifications : (1) the global average pooling layer is substituted by the spatial pyramid pooling layer (SPP), (2) a new convolution layer is added on top of the Conv5-3 layer of the VGG16 network and named as class activation map convolution (CAM-conv), (3) outputs of the Conv5-3 layer and the CAM-conv layer are concatenated and passed to the SPP layer for predicting probability of binary classification. Structure of the proposed network is shown in Figure 2.3.

2.1.5 Discussion

Most existing scene text proposal techniques have various limitations. For example, the symmetry text line and the adaptive selective search for scene text proposal are efficient but often generate a large number of false positive proposals. The deep learning based techniques [21, 31] produces a small number of proposals but the recall rate becomes unstable when the Intersection over Union (IoU) threshold increases. In addition, deep learning based scene text proposals require a huge amount of trai-

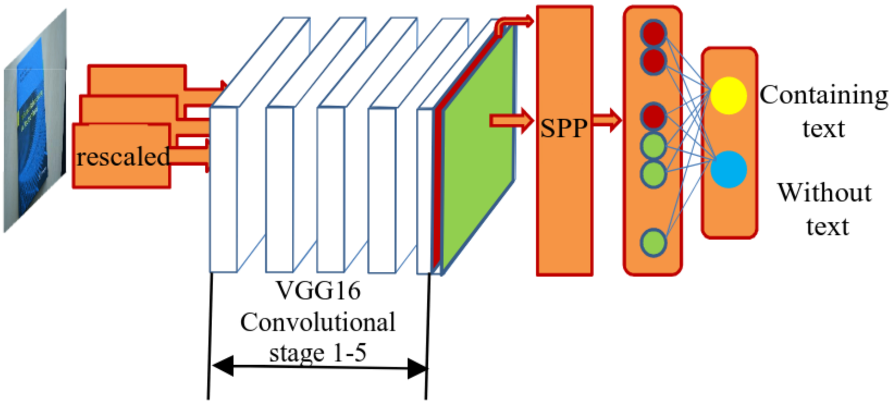


FIGURE 2.3 – Structure of a text attention network which is developed to provide a text confidence score map from an input color image.

ning data to optimize parameters of network architecture. In contrast, the proposed proposal techniques only need training data from ICDAR and SVT dataset, including 330 images. They are able to obtain a high recall rate with a small number of false positive proposals and stable with an IoU threshold increment, as shown in evaluation sections in chapters 3 and 4.

2.2 Automatic scene text reading systems

In this review, structures of state-of-the-art scene text reading systems are projected into a standard scene text reading framework including two major tasks as a scene text detection and a scene text recognition, excluding the DeepTextSpotter system which successfully combines these two tasks into a model. Relevant algorithms employed in each task are then described.

2.2.1 Edgeboxes based scene text reading system

This technique is developed following the standard framework of a scene text reading system, however two tasks are not wholly distinct. Scene text detection performance is still improved base on information gained from a word recognition model, leading to a stronger holistic text spotting system. Entire system is depicted in Figure 2.4

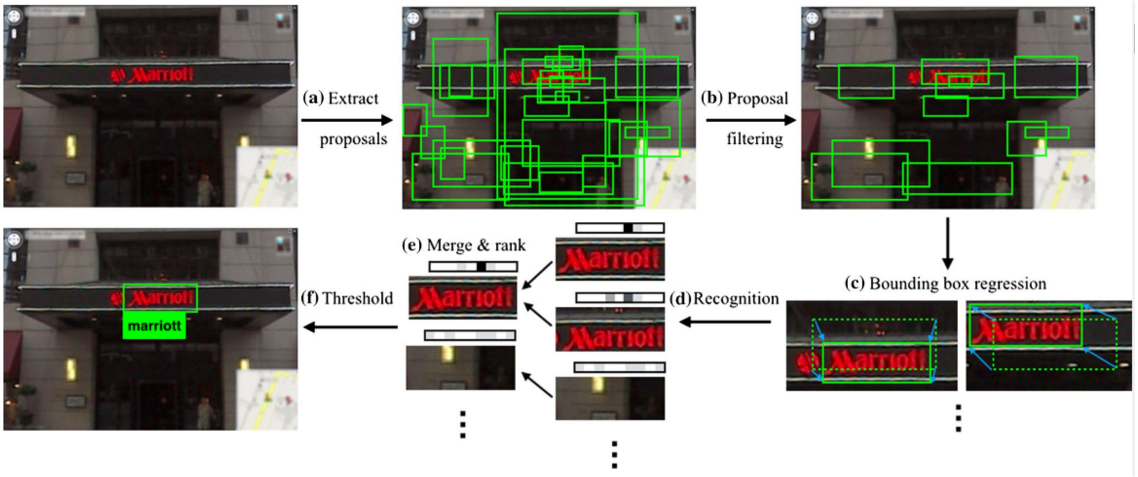


Figure 2.4 – A framework of an automatic scene text reading system developed based on scene text proposals generated by EdgeBox [26] and Aggregate Channel Feature Detector (ACF), and a deep learning based scene text recognition network.

a. Scene text detection

The scene text detection model consists of three steps : a proposal generation, a proposal filter and a proposal bounding boxes regression.

In the proposal generation step, two object proposal techniques are implemented : the EdgeBox technique [26] and the proposed Aggregate Channel Feature Detector (ACF). The ACF is a conventional sliding window detector in which proposals are localized by sliding windows and scored by passing its ACF features to an AdaBoost classifier. ACF features are channel similarity features including a normalized gradient magnitude, a histogram of oriented gradient (in 6 channels) and a raw grayscale input. These features in different input images scales are also taken in an account for scoring proposals. The different scale features approximation solution [88] is adopted to speed up this extraction process. Thresholding on scored ACF based proposals gives a set of word proposal bounding boxes of this detector. Finally, the generated scene text proposals are formed by combining proposals from the EB and the ACF detector.

In the proposal filter step, they are rescored by a random forest classifier applied on their Histogram of oriented gradient features. Proposals having scores falling below a certain threshold are rejected. Bounding boxes of remaining proposals are then refined to overlap better with ground truth boxes in the proposal bounding boxes regression step. A deep learning based regression network is developed consisting of four convolution layers ($\{filter\_size, number\_of\_filter\}$: $\{5, 64\}, \{5, 128\}, \{3, 256\}, \{3, 512\}$) with a stride of 1 and two fully connected layers (4k units and 4 units). An original bounding box is parametrised by its top-left and bottom-right corners coordinates, such as (x_1, y_1, x_2, y_2). The regression model is trained to predict more precise coordinates ((x'_1, y'_1, x'_2, y'_2)) which are tighter to the true text objects and overlap with ground truth boxes in higher IoU values.

The detection set still contains a huge number of false positive proposals and they will be eliminated later based on performance of the scene text recognition model.

b. Scene text recognition

The recognition model is designed base on the hypothesis that a word recognition problem can be solved similarly a multi-class classification problem. Each word in the pre-defined dictionary therefore is considered as a class. The model consists of five convolutional layers : [5, 64], [5, 128], [3, 256], [3, 512], [3, 512] representing as [filter-size, the number of filters] and followed by three fully connected layers : 4000, 4000, 90000 for the number of units. The 90000 units in the last fully connected layer correspond to 90000 pre-defined words in dictionary. Each hidden layers are followed by a rectified linear non-linearity activation (ReLU) and a 2×2 max pooling. Due to integrating with fully connected layers, input image size has to be fixed and it is set at 32×100. The model structure is illustrated in Figure 2.5 It has been trained using Stochastic Gradient Descent (SGD) with Dropout regulation on Synthetic text training data [78].

The final step as a false positive detection elimination is applied using recognition confidence scores (s_b) and predicted words (w_b) of proposal boxes. Non maximum suppression (NMS) is first applied on the same word label proposals to obtain the best proposals at each location. Subsequently, NMS performs to suppress non-maximal detections of different word label proposals with some overlap. Finally, multiple rounds of bounding box regression using the regression model mentioned

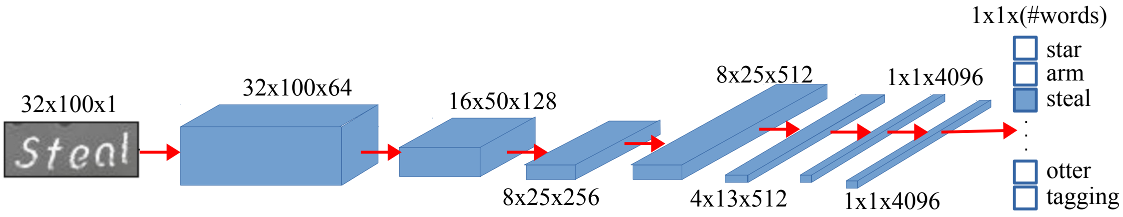


FIGURE 2.5 – A schematic of Jaderberg's recognition model which handles a scene text recognition task as a multi-class classification

in Section 2.2.1 and NMS performs on remaining proposals to improve overlap of detection results.

2.2.2 TextBox

a. Scene text detection

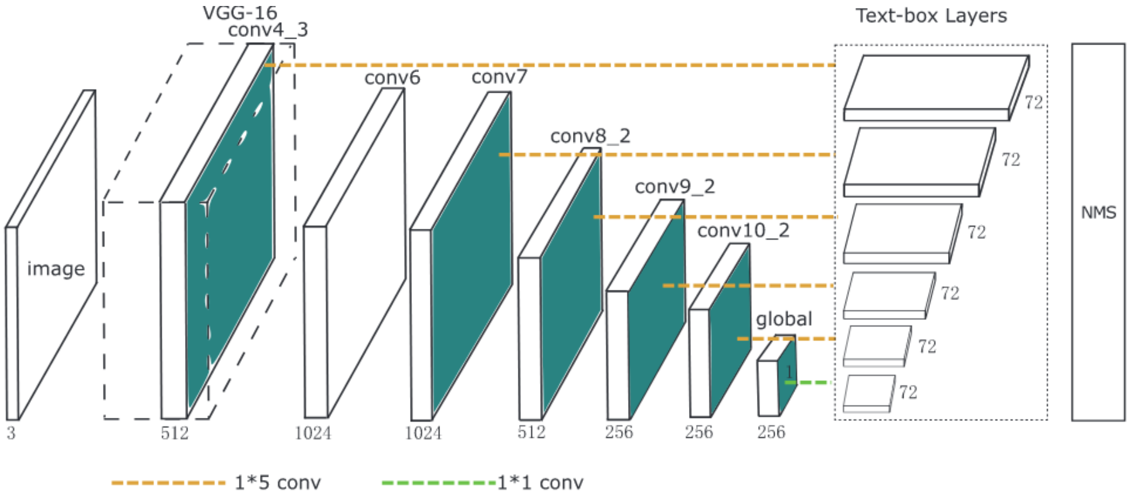


FIGURE 2.6 – Structure of the TextBox network, which inherits the first part of the popular VGG16 network and associates with the proposed Text-Box layer inspired by the Single Shot multibox Detector network (SSD)

The text detection network is depicted in Figure 2.6. It inherits the popular VGG-16 architecture [49], keeping layers from *conv1.1* through *conv4.3*. The last two fully connected layers are converted into convolutional layers and followed by a few extra convolutional and pooling layers, namely *conv6* to *conv11*. The textbox layer is integrated at the end of the network to regress text bounding boxes from network's default boxes and predict bounding box confidence scores. This last layer is inspired by the single shot multibox detector (SSD) [51] which is able to detect multi-scales objects with a single scale input image. Different from general objects, words have large aspect ratios. Therefore, large aspect ratio default boxes are included, containing 6 aspect ratios 1, 2, 3, 5, 7, and 10. Moreover, irregular 1×5 convolutional filters instead of the standard 3×3 ones are adopted, which fit better with larger aspect ratio words.

The text-box layer stacks up the last feature maps generated by the *conv11* layer with some other intermediate feature maps of the network to deal with the multi-scales problem. At each location in a feature map, it predicts bounding boxes confidence scores and regression values which is used to adjust associated default boxes to proper object bounding boxes. For example, at a location (i, j) in a feature map which associates with a default box (x_0, y_0, w_0, h_0), the output of the Text-box layer is a set of values as $(\delta_x, \delta_y, \delta_w, \delta_h, c)$, indicating that a box (x, y, w, h) is detected with a confident score c :

$$\begin{aligned} x &= x_0 + \delta_x \\ y &= y_0 + \delta_y \\ w &= w_0 exp(\delta_w) \\ h &= h_0 exp(\delta_h) \end{aligned}$$

The predicted bounding boxes at all collected feature maps are aggregated and undergo a non maximal suppression process to eliminate false positive proposals.

The TextBox network is first trained on the SynthText dataset [82] for 50k iterations then fine-tuned on the ICDAR2013 training set [2] for 2k iterations. The multi-tasks loss function is implemented as below :

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + L_{loc}(x, l, g)) \tag{2.10}$$

where N is the number of default boxes that match ground-truth boxes, x is the match indication matrix, c is the confidence, l is the predicted location, g is the

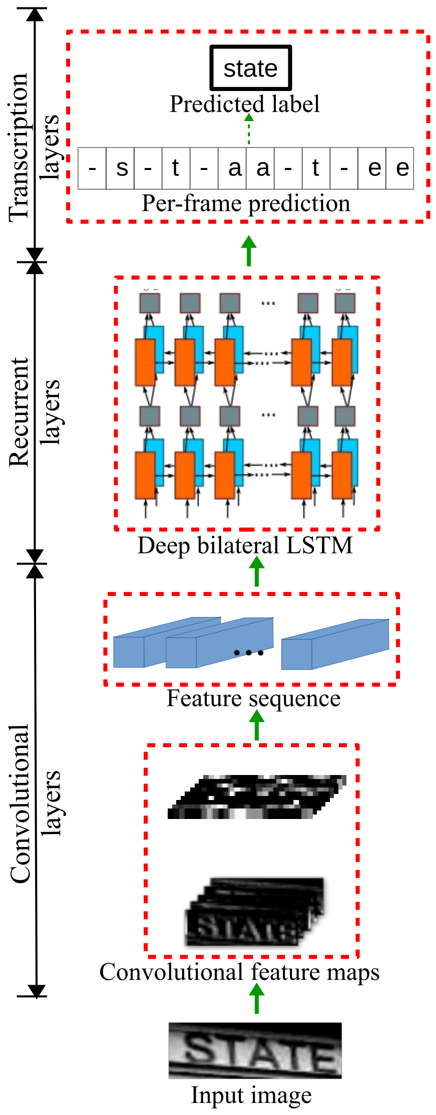


FIGURE 2.7 – A framework of the CRNN scene text recognition model. It consists of three major segments including a feature extraction based on convolution layers, a character distribution generation based on a recurrent neural network and a word generation based on a connectionist temporal classification.

ground-truth location, L_{loc} is a location loss function and L_{conf} is a detection loss function.

b. Scene text recognition

Figure 2.7 illustrates the structure of the scene text recognition model which consists of three segments, including a convolutional neural network (CNN) for sequential feature vectors extraction, a recurrent neural network (RNN) for predicting characters distribution of each feature vector in sequences and a transcription network for converting sequential characters distributions into readable words. The CNN based segment consists of convolution layers, max-pooling layers and element wise activation layers. A fixed size input image is converted into a feature map in which each column refers to a rectangle region in the input image. The column order in a feature map reflects the rectangle regions order in an input image. The RNN segment is built using bidirectional Long-Short Term Memory (LSTM). It takes sequences of columns as inputs and predicts labels distribution for each column. Labels are all Latin characters (36 characters) as well as a "blank" label denoted by "-". Label probability map therefore has size of 37-by-[the number of feature vectors in sequence]. Connectionist Temporal Classification (CTC) is adopted for mapping labels distribution map into predicted words including two steps. The first step is to map a labels distribution map ($\mathbf{y}$) into a labels sequence ($\mathbf{l}$), where each column is presented by a label which has the highest probability in the column. The second step is to map generated labels sequence $\mathbf{l}$ into a predicted word ($\mathbf{w}$) where repeated labels and "blank" labels are removed. For example, the second step will map a label sequence of "–hh-ee-l–lll-ooo—" into a word "hello". The conditional probability of predicted word $\mathbf{w}$ is defined as the sum of probabilities of characters in the corresponding labels sequence $\mathbf{l}$.

$$p(w|y) = \sum_i p(l_i|y) \tag{2.11}$$

where, $p(l_i|y)$ is defined as following equation :

$$p(l_i|y) = \prod_{t=0}^{T} l_i^t \tag{2.12}$$

with l_i^t is a probability of having label l_i at time stamp t.

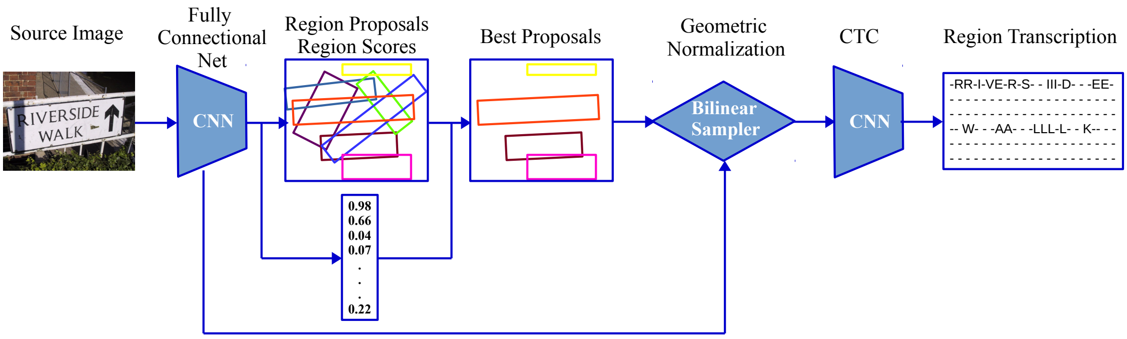


FIGURE 2.8 – Structure of a Deep Text Spotter network which successfully combines two single tasks of an automatic scene text reading system as a scene text detection and a scene text recognition into one trainable network

2.2.3 DeepTextSpotter

The novel idea in this state-of-the-art scene text reading system is an end-to-end trainable network for both a scene text detection task and a scene text recognition task. The proposed network is depicted in Figure 2.8.

In the scene text detection task, they adapted the YOLOv2 architecture [89] for developing a region proposal network. The first 18 convolutional layers and five maxpooling layers are inherited with proposed modifications : doubling the number of channels after every pooling steps, adding 1×1 filters to compress the representations between 3×3 filters, adding a bounding box rotation parameter into predicted proposals. At each point in an output feature map plane, proposal boxes are regressed from 14 different anchor boxes estimated from aggregated training images in the SynthText dataset [82]. Regression function is the logistic activation function presented in Equation 2.13, where, (x, y, w, h, θ) and $(r_x, r_y, r_w, r_h, r_\theta)$ are an actual bounding box position and a predicted bounding box position, respectively. The c_x, c_y are offset of the cell in the last convolution layer, and a_w, a_h are predefined height and width of the anchor box a. The rotation $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2})$ of a bounding box is predicted by r_θ. Text proposals which have confidence scores less than threshold $p_{min} = 0.1$ are eliminated. Remaining text proposals are then passed to the recognition network.

$$\begin{aligned} x &= \sigma(r_x) + c_x \\ y &= \sigma(r_y) + c_y \\ w &= a_w exp(r_w) \\ h &= a_h exp(r_h) \\ \theta &= r_\theta \end{aligned} \tag{2.13}$$

The scene text recognition network exploits a fully-convolutional network for predicting distribution of characters in time sequence. The network consists of four types of layers as a convolution layer, a max-pooling layer, a batch normalization layer and a recurrent convolution layer. Network input images have a size of $(H \times \bar{W})$, where the height H is fixed at 32 and the width $\bar{W}$ is flexible depending on image length. Network output shape is designed as $1 \times \frac{\bar{W}}{4} \times |\hat{A}|$ which is compressed into a 2D matrix with a shape of $\frac{\bar{W}}{4} \times |\hat{A}|$. The $|\hat{A}|$ is length of character set $\hat{A}$ including all English characters, ten numbers, and a symbol "$-$" representing a non-text class. At the end, the connectionist temporal classification (CTC) network is applied to transform network outputs into conditional probability distribution over label sequences which is later used to compute probability of a predicted word, please look at section 2.2.2 for a detail of how the CTC network is applied. The recognition network has been trained on the synthetic word dataset [78].

In order to join two above networks into a network, the bilinear sampling process is applied. For each region of interest (ROI) with a shape of $w \times h \times C$, it is mapped into a fixed-height tensor with a shape of $\frac{wH'}{h} \times H' \times C$ $(H' = 32)$ following Equation 2.14. The κ is a bilinear sampling kernel $\kappa(v) = max(0, 1 - |v|)$ and T is a point-wise coordinate transformation, which projects coordinates x' and y' of the fixed-sized tensor V to the coordinates x and y in the detected region features tensor U.

$$V^c_{x',y'} = \sum_{x=1}^{w} \sum_{y=1}^{h} U^c_{x,y} \kappa(x - T_x(x')) \kappa(y - T_y(y')) \tag{2.14}$$

In a training process, two networks are first trained separately to achieve their own task and a joined network is then trained on training images of the SynthText dataset, the Synthetic Word dataset, and the ICDAR datasets [1, 2, 81].

2.2.4 Discussion

Recent state-of-the-art scene text reading systems are developed using deep learning architecture. The TextBox system [50] consists of two deep learning models for two different task as scene text detection and scene text recognition. The Deep-TextSpotter [77] provides an advantage design that combines two independent tasks of scene text reading system into an end-to-end trainable deep learning model. In order to provide state-of-the-art performances, these techniques need to be trained on huge amount of data such as ICDAR, SVT, COCO Text, SynthText and so on. Our developed automatic scene text reading system adopts a framework presented in the EdgeBoxes based scene text reading system that exploits a scene text proposal technique for localizing texts in scene images and a scene text recognition model for both recognizing actual words in proposal regions and improving scene text detection performance. An advantage feature of our developed system is that we only used a top of 2000 proposals to pass to a recognition model instead of 10k proposals as mentioned in recent scene text proposals based systems [19, 20].

2.3 Conclusion

In this chapter, we reviewed state-of-the-art works relating to this thesis scope, including scene text proposal generation and automatic scene text reading systems in both hand-craft features based works and deep learning based works. Their performances are observed by their positions in the ranked list of algorithms in the robust reading competition website and scientific publications, which can be considered as proper benchmarks for validating our proposed techniques and systems. In the ensuing chapters, our proposed techniques and systems will be described and compared with these state-of-the-art works.

Chapter 3

Scene text proposal based on heuristic text specific features

Contents

Object proposal techniques well perform in localizing non-class objects. However, they fail to deal with scene texts. It is because scene texts have more diverse appearance than general objects due to not only texts' intra-class variations but also texts' distortions under environment impacts as discussed in section 1.2. There is only the EdgeBox [26] technique (EB) applicable for a scene text localization as described in the automatic scene text reading framework [20]. However, the EB has to integrate with another scene text proposal technique consisting of a Aggregate Channel Features (ACF) and an AdaBoost classifier to provide a proper set of proposals, which achieves a recall rate at 96% with 10000 proposals in the ICDAR2003 dataset [81]. In this technique, a large number of proposals is required to avoid loosing a certain number of challenging texts. However, it will influence efficiency of entire scene text detection/recognition systems when classification/recognition models have to repeat its processes on every proposals. Reducing a number of proposals while maintaining system performance therefore is an essential contribution. We suggest to integrate object proposal techniques with text specific features to filter out coarsely false positive proposals, as represented in recent developed scene text proposals [19, 21, 31].

A lot of text specific heuristic rules have been proposed to group atomic regions, such as edge pixels and MSER regions, into scene text proposals [44, 42, 31, 19, 32, 68]. For example, the stroke width transform (SWT) [42] proposes stroke width features measuring distances between opposite orientation edge pixels in edges connected components based on an assumption that text edge connected components have constant stroke width. In [68], a bunch of geographical text specific features are utilized for analysing MSER regions and eliminating false positive ones before moving on a grouping step.

Inspired by heuristic rules of text edges, we proposed two edge based text specific features for estimating text probability of edge connected components. These probabilities are also used to score our scene text proposals. True text proposals are supposed to achieve high scores and vice versa. This technique is developed based on the EdgeBox framework with our adapted grouping solution and scoring function, specifying for scene text objects.

3.1 Methodology

The proposed Text Edge Box (TEB) technique is designed to produce word-level text proposals in scenes. The framework is shown in Figure 3.1 including two main steps as a text edge map generation and a grouping-scoring-ranking proposal. Firstly, we exploit the Canny edge detector [90] to generate a binary edge map. A gradient map, an orientation map are also collected from the Canny's immediate

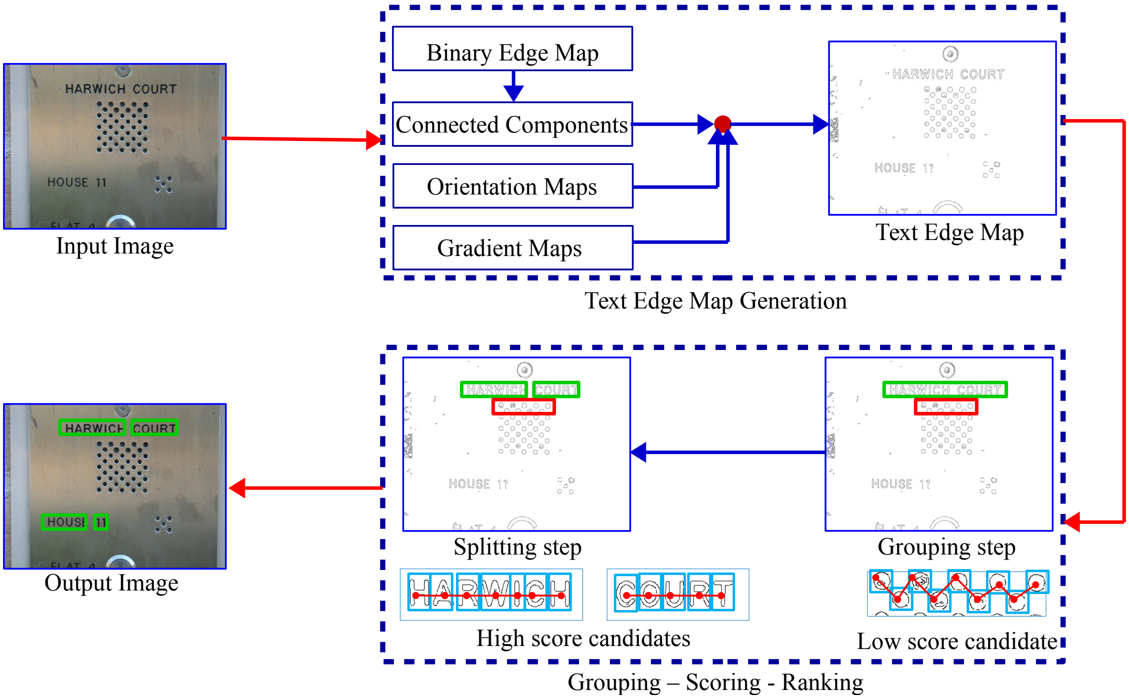


Figure 3.1 – A framework of the proposed Text-Edge-Box technique, including two main task as a text edge map generation and a grouping-scoring-ranking proposals. In the output image, only good proposals are presented for an illustration purpose

steps. Pixels in the orientation map are then normalized into the range of $[0,\pi]$. Connected components (CCs) are labelled within the binary map, which are further scored by a combination of two proposed low-cue text features including an edge pair feature (EP) and an edge variance feature (EV). They are estimated from the orientation and the gradient at the corresponding CC pixels, respectively. Text edge map is an image that has the same size with an original image and its pixels are assigned connected component scores. Secondly, the CCs are then merged together to produce word-level proposals. A proposal scoring function is designed, which computes probability of being a word of each word-level proposal by combining the scores of CCs belonging to the proposal and scores of their relationships (correlation in component scores, component sizes and links between pair of components). Finally, word-level proposals are sorted in the descending order, and those with high scores are superiorly identified as words.

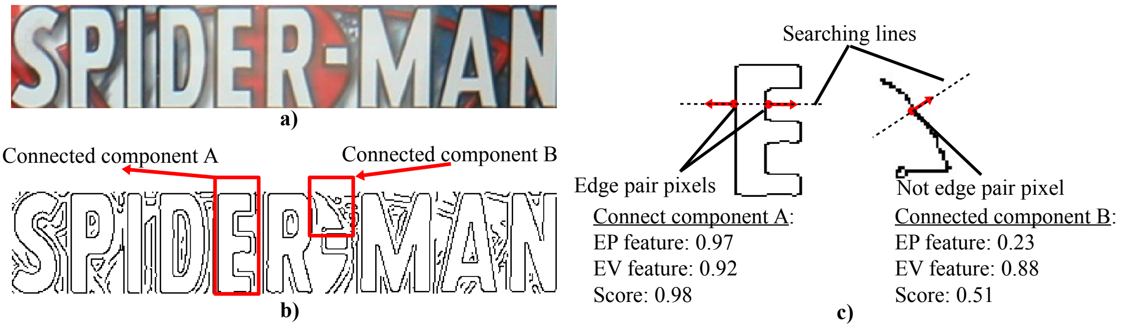


FIGURE 3.2 – The path a and b are respectively an example image and its own binary edge map. The path c shows examples of connected components including one for text connected component (connected component A) and one for non-text connected component (connected component B). The red arrows are orientations of the considered pixels in the connected components. The dash lines are searching lines corresponding to the orientations of pixels. The pixels in the couple pixels shown in the connected component A are defined as edge pair pixels. Obviously, a text connected component includes much more number of edge pair pixels than a non-text connected component.

3.1.1 Specific text edge features

In this research, we define two specific text edge features as an Edge Pair Feature (EP) and an Edge Variance Feature (EV). They are our key contributions which are proposed to clarify text edges from non-text edges and evaluate how possible connected components (CCs) are text CCs. The evaluated probabilities are also CCs' scores and they are stored in a Text Edge Map (TEM).

The edge pair feature

The first feature is an edge pair (EP) which is inspired by the Stroke Width Transform method [42]. It is developed based on a hypothesis that CCs of text objects are likely to contain the high portion of couples pixels that have opposite orientations, like the connected component A illustrated at the path c in Figure 3.2. From now, we name these pixels as edge pair pixels. In order to detect them, we start at each given pixel in a CC and its orientation is used to decide a searching line (dash lines in Figure 3.2.c). If an opposite orientation pixel in the same CC has been found in the searching line, the considered pixel and the searched one with an opposite orientation are defined as edge pair pixels. The EP feature of a given CC is defined as a fraction of edge pair pixels in the CC as follows :

$$EP(CC) = \frac{N_{pp}(CC)}{N_p(CC)} \tag{3.1}$$

Where $N_{pp}(CC)$ and $N_p(CC)$ denote the number of edge pair pixels and the number of edge pixels belonging to a CC under study, respectively. The value of this feature is in the range of [0, 1]. The CC having higher EP value is more likely to be a text CC.

The edge variance feature

The second feature is an edge variance (EV) that measures the variance of gradient magnitudes of pixels in a CC. This parameter is useful because the gradients of pixels in the boundary of an individual character (or boundaries of characters in a same word) are often monotonous. Therefore, their variances are expected to be small. We utilize an exponential function of the gradient variance in order to normalize these values into the range of [0, 1] and produce high values for text CCs as below :

$$EV(CC) = e^{-var(CC)} \tag{3.2}$$

Where the $var(CC)$ denotes the variance of the gradient of pixels in a CC.

The text edge map

The text edge map is a score map that shows the being-text probability of each CC. Pixels in a CC contain the value of the CC score, and other pixels have values of zero. The score of each CC is estimated by a weighted summation of its two text probability features as follows :

$$CCscore = \alpha EP + (1 - \alpha)EV \tag{3.3}$$

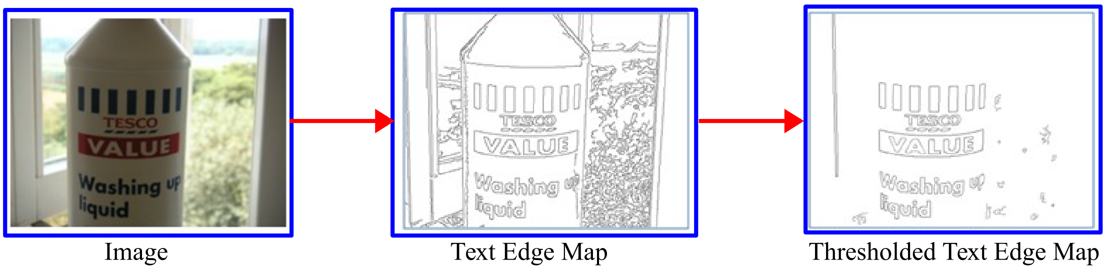


Figure 3.3 – Effectiveness of the two proposed heuristic text specific features on clarifying text edges from others by scoring them high score values. By applying threshold on a text edge map, we can perceive that most non-text edges are eliminated.

where α is in the range of [0, 1]. Its' value is determined through a tuning process which is described in Section 3.2.1. Since both features have their values in the range [0, 1], all pixels in the TEM are in the range of [0,1]. An example of the text edge map is shown in Figure 3.3. By applying threshold on the text edge map, we can perceive that most of non-text edges have been removed, meaning that text edges pixels have been scored higher values than others.

3.1.2 Scene text proposal generation

Generated CCs are then merged into text lines which are later split into small subgroups referring to word-level proposals. The proposed grouping strategy is illustrated in Figure 3.4.

As Figure 3.4 shows, starting with a given CC (called candidate A - a dark blue box), three properties of its bounding box (bb_A) are exploited including box height (h_A), box width (w_A) and box size (s_A). A corresponding search area is designed by expanding the bb_A, where the search area width (w_{search}) is equal to the image width, and the search area height (h_{search}) is γ times bigger than the h_A determined by expanding the h_A equally in both sides in the vertical direction (a red box). CC candidate B (a green box and its properties are bb_B, w_B, h_B, and s_B) is merged with the CC candidate A to form a group if the bb_B satisfies : (1) The ratio of intersection between the bb_B and the A's search space to the s_B is higher than τ_s, (2) the ratio

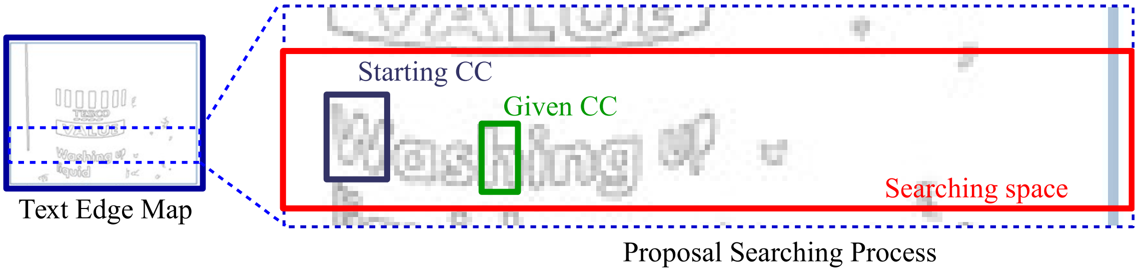


FIGURE 3.4 – Illustration of the proposed connected components grouping strategy. A searching space is estimated from a bounding box of a starting connected component (CC). A given CC in the searching space will be grouped with the starting CC if it satisfies certain grouping rules.

between $min(w_A, w_B)$ and $max(w_A, w_B)$ is higher than τ_w, (3) the ratio between $min(h_A, h_B)$ and $max(h_A, h_B)$ is higher than τ_h. The parameters that are γ and τ_s are sensitive to horizontal texts. The τ_w and τ_h are sensitive to size relationship between characters in a word. How to set values for these parameters is discussed in Section 3.2.1.

In order to divide text line proposals into small subgroups which refer to word-level proposals, an average of distances between adjacent CCs' boxes in the horizontal direction is estimated. Dividing positions are decided at the location where the distance is larger than the average value.

3.1.3 Scene text proposal ranking

This section elaborates a strategy to provide a ranked list of proposals in the decreasing priority order. Four measures are defined as S_a, S_c, S_h and S_o, which are all normalized in the range of [0,1]. The S_a is an averaging scores of CCs within a word-level proposal region, where the score of each included CC is defined in Equation 3.3. The S_c, S_h and S_o indicate affinity among grouped CCs. These measures are designed so that a proposal covering a word will have a high value. In particular, they are calculated from the variance of scores of grouped CCs, the variance of bounding box's height of CCs in a proposal region and the variance of angles between lines linking centroids of two neighbourhood CCs and the horizontal axis, respectively. These angles are adjusted in the range of $[0,\pi]$.

Generally, a proposal has a high likelihood of being word if it satisfies (1) its CCs should have similar scores, (2) the heights of CCs should be approximately stable and (3) the connection lines between the CCs should be appropriate in same direction, referring to text lines. Therefore, the variances of these measures are expected to be small for a word region proposal. In order to derive a high S_c, S_h, S_o value for a group of CCs which is likely to be a word, and also normalize these measures in the range of [0, 1], the *arctan* functions is implemented for each measure as follows :

$$S_x = \frac{2 \cdot arctan(k_x / var_x)}{\pi} \tag{3.4}$$

where the symbol "x" represents for the c, h, and o, and the var_x refers to the variances of CC's score, height, and angle respectively. The parameter k_x is set at the middle of each measure range, i.e. 0.5, a half of bounding box height and $\pi/2$, respectively for S_c, S_h, and S_o. The score function of a proposal region S_p is computed as :

$$S_p = S_a \cdot arctan(k_1 \prod S_x) \tag{3.5}$$

where, the function $arctan(k_1 \prod S_x)$ is used to control the relationship between the S_p and the S_a. If the S_x makes the $arctan(k_1 \prod S_x)$ value higher than 1, we say that the S_x has a supporting effect ($S_p > S_a$). If the S_x makes the function value smaller than 1, it has a penalizing effect ($S_p < S_a$). It means that : although a proposal has a high value of the S_a, it is unlikely to be a word proposal if its CCs provide low S_x values (referring to the penalizing effect, see the low score candidate at the grouping-scoring-ranking step in Figure 3.1). The parameter k_1 adjusts the role of the S_x measures. In particular, if the k_1 increases, the role of the S_x measures is reduced. In this research, we expect that if the $\prod S_x$ is higher than the middle value of its' range as 0.5, the function $arctan(k_1 \prod S_x)$ has the supporting effect and vice versa. So, the parameter k_1 is set at 3 as the $arctan(3 \cdot 0.5) \simeq 1$. Note that, the S_x are considered only when the number of CCs in a proposal is larger than 3. Otherwise, the S_p is calculated based on the S_a only.

3.1.4 TextEdgeBox implementation

The Text Edge Boxes (TEB) technique is implemented by two programming languages as Matlab and C++. Matlab programming language is used to read and pre-process images, generate a Canny binary edge map, a magnitude gradient map as well as an oriented gradient map, evaluate technique performance and save evaluated results. C++ programming language is used to implement major contributions of the TEB. The TEB class has been shown in Appendix. In the below TEB pseudo code, Matlab functions are highlighted in white backgrounds and C++ functions are highlighted in gray backgrounds.

Algorithm : Text Edge Box (TEB)

```
//Import image
Img = imread('Image address') ;
//Create a binary edge map (E), a gradient map (G), and an oriented gradient map (O)
[E,G,O] = CannyEdgeDetector(Img) ;
//Generate scene text proposals
ppb = TEB(E, G, O) ;
//Sort proposal boxes in descending order
ppb = boxsort(ppb) ;
//Evaluate scene text proposal performance. The gtb is ground truth of a corresponding image.
Recall = PPBEval(ppb,gtb) ;
//Save evaluated performance
save('Recall.mat','Recall') ;
```

A TEB technique is inspired by the EdgeBox technique [26] and two EdgeBox components have been adapted for scene text objects, including **ClusteringEdges** and **BoxesGenerator**. They have been rewritten by our implemented functions such as two edge based text specific feature generation (**EPCC** and **EVCC** functions), a connected component grouping (the **Group** function), a text line splitting (the **BoxesBreakDown** function), a bounding boxes generation (the **boxlist2Box** function). Before applying these two components, 8-connected components are searched on a Canny binary edge map.

The **ClusteringEdges** component focuses on scoring connected components using two proposed text specific features, and generating text edge map. In the generated text edge map, edge pixels of a connected component are assigned the same value which is a connected component score.

```
Function : ClusteringEdges

//Inputs of the ClusteringEdge function are a magnitude of gradient map (G), an
oriented gradient map (O), and a list of 8-connected components (CCs).
//Score connected component. The EPmap and EVmap contain connected com-
ponent scores calculated by using EP and EV features, respectively
EPmap = EPCC(CCs,O) ;
EVmap = EVCC(CCs,G) ;
//Generate text edge map (TEM)
TEM = alpha × EPmap + (1-alpha) × EVmap ;
 return(TEM) ;
```

The **BoxesGenerator** component is divided into three tasks : a connected components grouping, a group splitting, and a bounding box generation. The proposed grouping solution is implemented in the connected component grouping task (the **Group** function). Its outputs are hypothesized to be text lines and they are then divided into sub-groups targeted to be words (the **BoxesBreakDown** function). The bounding boxes generation task is applied to provide corresponding bounding boxes for text lines, words and single connected components. Concurrently proposal scores are also calculated using the proposal scoring function which is implemented inside the **boxlist2Box** function.

```
Function : BoxesGenerator
```

```
//Input of the BoxesGenerator function is a list of 8-connected component (CCs)
//Task 1 : Connected component grouping
TLGroupedCCsList = [ ] ;
for CC in CCs :
    SearchSpace = SearchSpaceInit(CC) ;
    LocalGroupedCCs = Group(SearchSpace,CCs) ;
    TLGroupedCCsList.push_back(LocalGroupedCCs) ;
//Task 2 : Split text lines into words
WGroupedCCsList = BoxBreakDown(TLGroupedCCsList) ;
//Task 3 : Bounding box generation. The boxlist2Box contains our proposal scoring function
ppb = [ ]
for CC in CCs :
    pp = boxlist2Box(CC) ;
    ppb.push_back(pp) ;
for GroupedCCs in TLGroupedCCsList :
    pp = boxlist2Box(GroupedCCs) ;
    ppb.push_back(pp) ;
for GroupedCCs in WGroupedCCsList :
    pp = boxlist2Box(GroupedCCs) ;
    ppb.push_back(pp) ;
 return(ppb) ;
```

Function : boxlist2Box

```
//An input of the boxList2Box function is a list of 8-connected components (CCs)
if CCs.size() < 4 :
    s = GetScoreMean(CCs) ;
else :
    sa = GetScoreMean(CCs) ;
    sc = GetSpaceVariance(CCs) ;
    sh = GetHeightVariance(CCs) ;
    so = GetOrientedLinkVariance(CCs) ;
    s = Score(sa, sc, sh, so) ;
[x, y, w, h] = GetBoundingBox(CCs) ;
return([x,y,w,h,s]) ;
```

3.2 Evaluation

In this section, we describe how we optimize the proposed technique and compare its performances to other state-of-the-arts. In order to improve the robustness of the proposed system under a wide diversity of text appearance, these five parameters ($\gamma, \tau_s, \tau_w, \tau_h$ and α) are determined based on the joined training sets of the ICDAR2013 dataset (for high contrast texts) [2] and the Street View Text (SVT) (for blur texts) [3]. An optimal combination of those parameters will be used to set-up the proposed system. The proposed technique has been compared to other state-of-the-art techniques including the simple text specific selective search (TP) [19], the Symmetry-Text Line (STL) [24], the DeepText (DT) [21], the EdgeBox (EB) [26], the Geodesic (GOP) [91], the RandomizedPrim (RP) [92] and the Multiscale Combination Grouping (MCG) [93] on the two scene text datasets by following scene text proposal evaluation framework as presented in Section 1.6.2.

3.2.1 Parameters tuning

We first focus on generating high quality set of proposals, which maximizes overlap with the ground truth, by varying the four parameters $\gamma, \tau_s, \tau_w, \tau_h$. The ranking step is ignored and all number of generated proposals used for evaluation. After obtaining good proposals, we then concentrate on scoring proposals to be able to shift likely-to-be-text proposals to the top of the list by arranging the found group in the descending order, so the proposed technique can perform well under a limitation of number of proposals.

In the first optimization step, the parameter γ is tuned in the range of [1, 2] with an internal step of 0.5 and other three parameters (τ_s, τ_w, τ_h) are tuned in the range of [0.5, 1], [0.1, 1], [0.5, 1] with an internal step of 0.1 respectively. All generated proposals were collected for evaluating detection rate. The best set of these values are found as $\gamma = 1.5, \tau_s = 0.7, \tau_w = 0.3, \tau_h = 0.7$.

In the second optimization step, the parameter α has been estimated to find the best sorting solution to be able to shift good proposals boxes to the front of the list. This parameter controls the contribution of two proposed features (EP and EV) which are reflected into values of S_a, S_c and S_p in the scoring function (Equation 3.4, 3.5). The number of proposals is set to maximum as 2000, the α value varies from 0 and 1 with an internal step of 0.1. The optimal performance on the joined training set under many thresholds of IoU is found at the $\alpha = 0.7$. When the α is 0 or 1, it means that connected components are scored based on only the EV feature or the EP feature, respectively. As the results presented in Table 3.1, when we remove the EP feature, performance of the TEB technique drops dramatically. In contrast, its' performance is just slightly lower than optimal performance when the EV feature is removed. Therefore, the EP feature seems more reliable than the EV features. In comparison with other state-of-the-art techniques, the α is set at 0.7.

Table 3.1 – The detection rate (in%) of the proposed technique with the variation of the α value from 0 to 1 with an inner step of 0.1, and the difference of the IoU threshold on the joined training sets of the two scene text datasets : ICDAR2013 and SVT. The maximum number of proposal regions is 2000.

α	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
$IoU = 0.5$	70.43	87.34	90.51	93.13	93.94	94.21	94.39	**94.76**	94.67	94.58	94.14
$IoU = 0.7$	55.61	72.69	75.95	81.74	82.73	83	82.91	**83.09**	82.82	82.64	82.82
$IoU = 0.8$	47.03	63.56	66.27	72.6	73.87	73.96	73.87	**73.96**	73.87	73.69	73.6

Finally, in evaluation and comparison to other state-of-the-art techniques, the five heuristic parameters of the proposed technique is set as :$\gamma = 1.5, \tau_s = 0.7, \tau_w = 0.3, \tau_h = 0.7$ and $alpha = 0.7$.

3.2.2 Experimental results

Figure 3.5 illustrates the performance of the proposed technique as well as the comparison with state-of-the-art techniques. In the left column, the detection rate

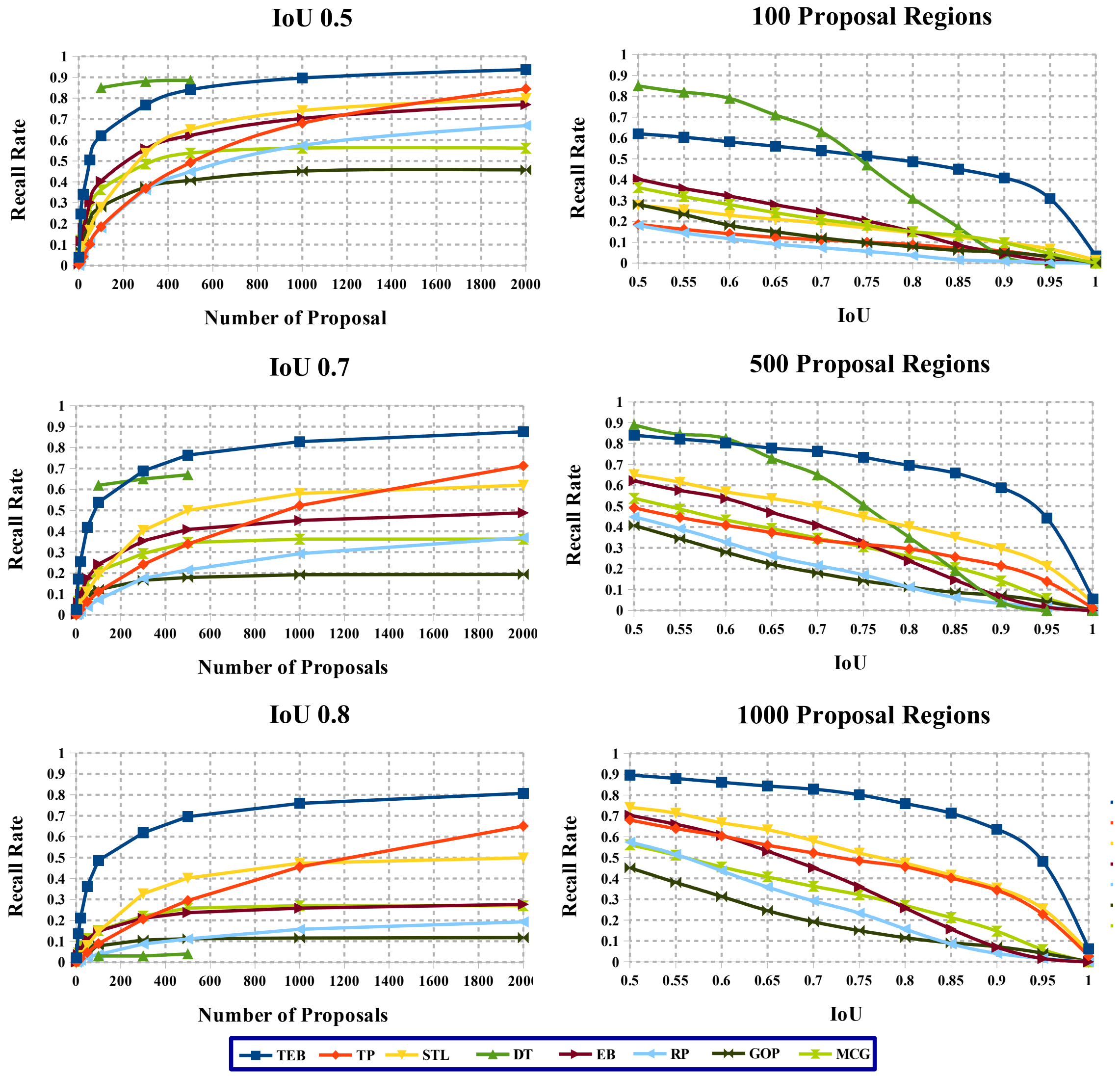


FIGURE 3.5 – The detection rate evaluation vs the number of proposals (left column) and the IoU threshold (right column) of the Text Edge Box (TEB) and other state-of-the-art algorithms, including the simple text specific selective search (TP) [19], the Symmetry-Text Line (STL) [24], the DeepText (DT) [21], the EdgeBox (EB) [26], the Geodesic (GOP) [91], the RandomizedPrim (RP) [92] and the Multiscale Combination Grouping (MCG) [93] on the ICDAR 2013 dataset

vs the number of proposals on the ICDAR 2013 dataset has been calculated under the three different IoU thresholds, i.e 0.5, 0.7 and 0.8. The TEB algorithm obviously outperforms other methods at the different IoU values when the number of propo-

Table 3.2 – Recall (%) and processing time (in second) of the proposed TEB and other state-of-the-art techniques under different IoU thresholds on the ICDAR2013 dataset. The Nppb denotes an average number of proposals that the techniques need to achieve its presented recalls.

IoU	0.5	0.7	0.8	Nppb	times (s)
TEB [25]	**94.25**	**87.95**	**81.55**	1777	5.4
TP [19]	84.47	71.32	65.11	1907	5.17
STL [24]	79.78	62.04	49.91	1034	361.3
DT [21]	88.5	67	4	**500**	–
EB [26]	76.93	48.81	27.67	1968	**1.02**
GOP [91]	45.68	19.39	11.76	1040	4.3
RP [92]	66.91	36.95	19.3	1917	10.07
MCG [93]	56.16	36.21	27.02	550	28.92

sals is larger than 1000. The DT leverages on the deep learning model for scoring proposal regions. Its' performance is therefore very competitive at the small number of proposals, especially with the IoU = 0.5. This is due to the deep learning model that has advantage in recognizing false positive regions and eliminating them from the generated proposal list. However, our proposed system localizes scene texts more successfully when the IoU threshold increases to 0.7 and 0.8, which is targeted for scene text objects as discussed at Section 1.6.2. The TP is the most competitive technique when a huge number of proposals are accepted. The EB shows better result than the TP when the number of proposals is smaller than 1000. On the other hand, its performance deteriorates when the number of proposals increases. The right column shows the second experiment that estimates the detection rate vs the IoU threshold for the different set of proposals : 100, 500 and 1000. The TEB outperforms other methods (excluding the DT) significantly under the different bunch of proposals. When the number of proposals increases and the IoU requirement is more constrained, the proposed TEB performs better than the DT. Note that the DT only provides maximally 500 proposals on an image.

In addition, we also track the average number of proposals that each method provides to achieve their best performance. Hosang [23] shows that this criterion

Table 3.3 – Recall (%) and processing time (in second) of the proposed TEB and other state-of-the-art techniques under different IoU thresholds on the SVT dataset. The Nppb denotes an average number of proposals that the techniques need to achieve its presented recalls.

IoU	0.5	0.7	0.8	Nppb	times (s)
TEB [25]	**87.64**	**47.91**	20.09	1890	2.2
TP [19]	74.65	42.5	21.33	1972	5.94
STL [24]	77.13	31.07	10.36	1358	433.82
EB [26]	76.35	47.45	**23.96**	2000	**1.28**
GOP [91]	52.09	18.24	6.8	1117	3.78
RP [92]	62.6	27.05	12.21	2000	8.02
MCG [93]	54.71	24.27	8.66	**557**	14.97

correlates well with the detection performance and it has been used to evaluate quality of the proposals in the TP [19]. Table 3.2 and 3.3 show the experimental results on the two datasets. On the ICDAR2013 dataset, the TEB algorithm provides less number of proposals than competitive techniques as TP and EB. On the SVT dataset, the TEB performs slightly lower in comparison with the EB algorithm, but better than other state-of-the-art algorithms. On the other hand, the average number of proposal regions required are clearly more than those for the ICDAR2013 dataset. Besides the poorer image quality in the SVT dataset, one important reason of the lower performance is due to the ground truth of the SVT dataset where the manually labelled bounding boxes are often much larger than the actual boxes. This is illustrated in Figure 3.6 where the ground truth boxes in the red color are clearly much larger than the boxes produced by the proposed TEB in the green color. The efficiency of the proposed technique is also evaluated based on the processing time. All above techniques are evaluated on the same computer and executed in one thread as the Xeon CPU E5-1650 v2 @ 3.5GHz. As presented in Table 3.2 and 3.3, the proposed TEB is comparable to the most efficient methods except the original EB method. However, the original EB method does not perform well in term of the number of proposals and the maximum recall obtained. For the DeepText method [21], the authors have not released their program yet, and we do not have a processing time report in our device. According to their report, their algorithm takes average 1.7 second for processing an image in the ICDAR2013 dataset in their device using the single GPU K40 which is much more powerful than what we used.

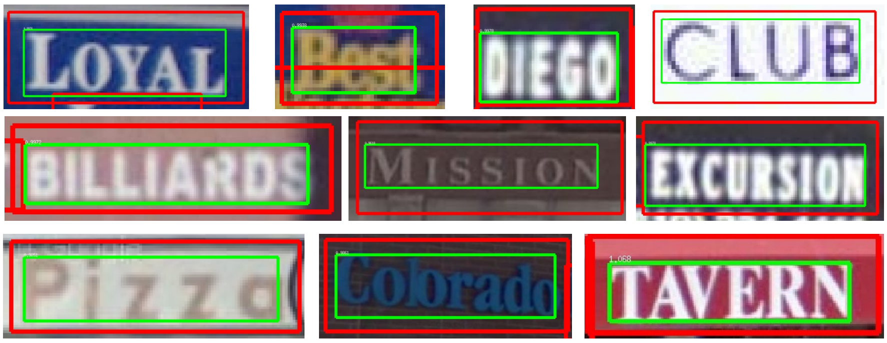


FIGURE 3.6 – There are some examples of the SVT ground truth boxes which our proposals cannot localize with the IoU threshold of 0.7. The red boxes are the ground truths and the green boxes are our proposals. The proposal boxes are much more smaller and closer to the scene text objects than the ground truth boxes.

Furthermore, the IoU based evaluation often has certain constraints where the proposals have small overlap with the ground truth boxes but cover entire objects as illustrated in Figure 3.6. We also adopted another evaluation that uses word recognition models to estimate the quality of proposals. The well-known word recognition model provided by Jaderberg [78] is implemented to perform this additional evaluation. A proposal is a correct localization if it overlaps with one of the ground truth boxes and provides enough information to help the recognition model to recognize correctly. The better proposal technique will achieve the higher F-score at the output of the recognition model. The quality of the recognition model is first estimated on the ground trust boxes of the testing sets in the two datasets. The F-scores of the model on the ICDAR2013 and the SVT dataset are 72.27 and 83.91 respectively. They are presented in Figure 3.7 as the RegModel's performances. This is the maximum performance that each proposal technique might obtain if they can provide good proposals that match perfectly to the ground truth boxes. As shown in Figure 3.7, the TEB method produces the largest number of good proposals which help the recognition model read contained words correctly. In addition, the performance of

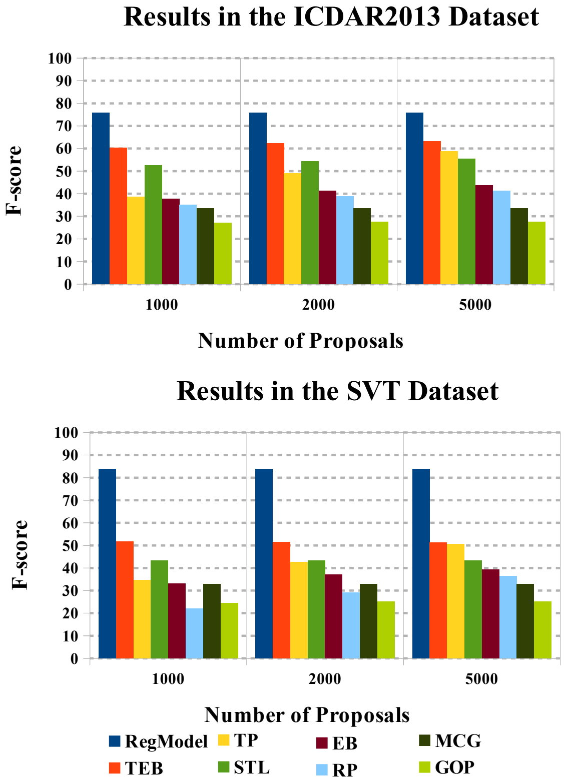
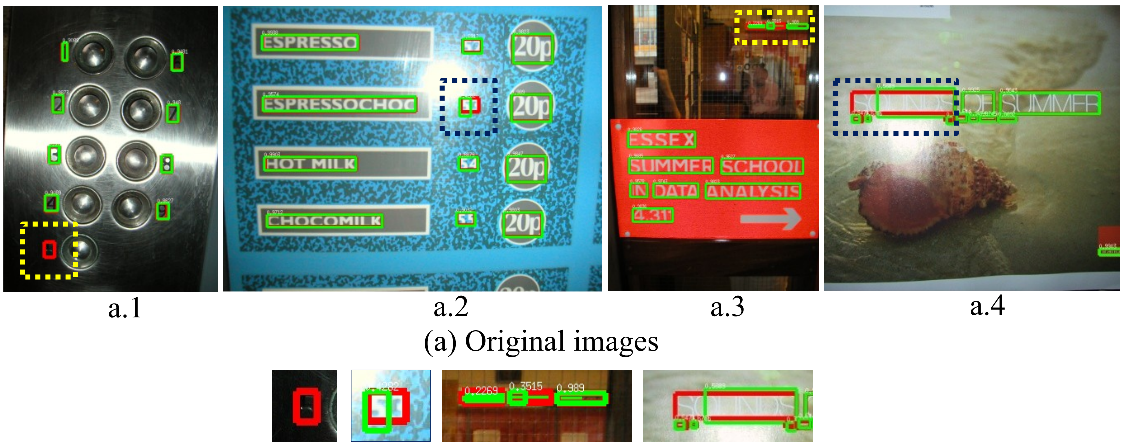


FIGURE 3.7 – The performance of the end-to-end word spotting systems which are constructed by the comparison techniques and the word recognition model [78]. The performance of RegModel here is the result of the word recognition model [78] tested on the ground truth of the testing sets of two scene text datasets as the ICDAR2013 and the SVT

the proposed TEB algorithm just changes slightly when we increase the number of proposals from 1000 to 5000 in both datasets. It proves that the proposed technique ranks proposals better than other techniques. Therefore, most good proposals have been ranked correctly at the top of the list.

Figure 3.8 illustrates several typical scenarios where our algorithm often fails to provide good proposals including ultra-low contrast (a.1), complex background (a.2), very small text size (a.3) and uneven illumination (a.2, a.4). In particular, the edges of texts in a complex background are often connected with edges of other objects where the edge pair feature may not be extracted reliably. Similarly, when the text objects are covered by shadow or uneven illumination, the forms of texts



(a) Original images

(b) Cropped regions in the dash rectangles

Figure 3.8 – Examples from the ICDAR2013 dataset that our algorithm failed to localize. The red boxes are ground truths and the green boxes are our proposal regions.

boundaries have been destroyed, where the text edge features may not be extracted properly either. In term of low contrast, the text edges could be missed by the Canny edge detector due to the ultra-low gradient magnitude.

One distinctive feature of the proposed TEB is that it does not rely on any classifier for eliminating false positive proposals (as implementation in the TP, DT). It simply uses the two proposed features and the geometric relationships among CCs to rank proposals. Nevertheless, very good performances were obtained on the two public scene text datasets, which demonstrate the effectiveness of the proposed text-specific proposal technique.

Besides, while testing on the SVT dataset that includes a certain amount of non-horizontal text lines, the TEB is still competitive as compared with the TP, which is designed without horizontal restriction. It is observed that the proposed algorithm can handle multi-orientation text lines. As discussion in Section 3.1.3, the horizontal text line assumption can be relaxed by increasing the parameter γ for a larger searching space and reducing the parameter τ_s for retaining more CCs. Then, the scoring function should be upgraded to handle a huge number of CCs in the merged groups.

3.3 Conclusion

In this chapter, we proposed a text-specific proposal algorithm to search text regions in scenes. Two text-specific features, namely, an edge pair and an edge variance, were designed to search for more likely text components. In order to measure the text likelihood of the proposal boxes, we designed a scoring function that computes word probability based on correlations of connected components in their score, height, and orientation of connections. The effectiveness of the proposed technique has been demonstrated by its superior performance as compared with other state-of-the-art algorithms.

Chapter 4

Scene text proposal based on max pooling technique

Contents

The Text-Edge-Box (TEB) technique and other existing scene text proposal techniques usually creates proposals by merging atomic elements of images (edge connected components, MSER regions) based on a bunch of heuristic rules with a lot of parameters [32, 44, 19, 42, 58]. Optimizing those parameters could cost huge amount of time. For example, the TEB technique has to optimize a combination of five parameters, including four parameters in a grouping step and one parameter in a scoring step. The TextFlow (TF) [32] has three parameters as a horizontal distance, a vertical distance and a size similarity which are needed extensive tests to search for optimal values. Generally, those parameters evaluated on geographical relation among atomic elements, and optimal values are thresholds applied on those parameters for providing grouping decisions. In this chapter, we are intent on providing a proper grouping solution that are independent of text heuristic parameters.

Studying on scene text geographic appearance, we observe that texts usually appear in words and text lines, and internal distances between characters are usually extracted as a feature and used for grouping atomic regions into text lines and splitting text lines into words [32, 25, 44]. Particular, internal distances between characters in a word are smaller than gaps between words and external distances from outer characters to non-text objects. Mean, variance and standard variance of the distances are usually estimated based on large cases of texts in scenes and used as thresholds for grouping and splitting. There is an idea that proposals can be generated from a dendrogram built based on distances, where the closest atomic regions are grouped first and followed by farther ones. In [19], this idea has been adopted, and the internal distances has been combined with other geographic features.

In this chapter, we propose a solution that can provide a dendrogram based on internal distances between atomic regions without calculating their distances. The idea is inspired by the process of a max-pooling layer in the deep learning network architecture. Two components have been adopted as a pooling window and strides. The stride what helps to shrink size of feature maps is adopted to shift atomic regions closer to each others. The pooling window is applied to select features for a new feature map and make grouping decisions concurrently. Atomic regions are grouped together if their pixels exist under a given pooling window. An iteration process is also employed to build a dendrogram grouping solution based on which scene text proposals have been generated.

4.1 Methodology

The proposed scene text proposal technique is developed following the general object proposal framework including two main steps as a proposal generation and a proposal scoring/ranking. In the proposal generation, texts are first localized base

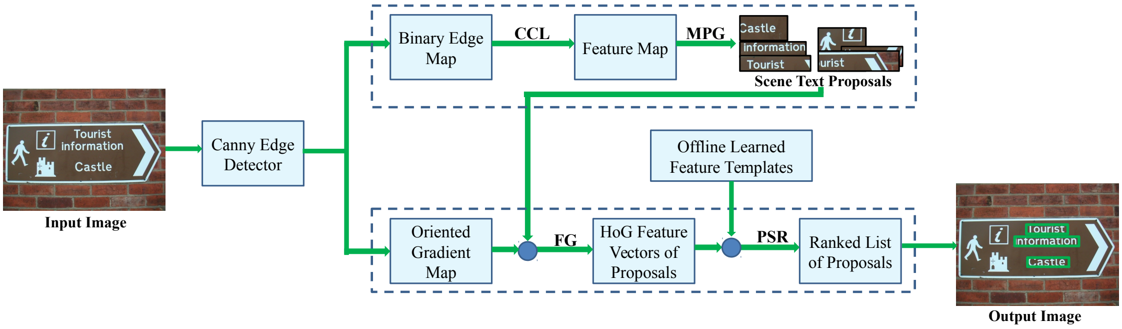


FIGURE 4.1 – The framework of the proposed scene text proposal technique, including a max-pooling based grouping strategy for scene text proposal generation and a low-level feature based proposal ranking (**CCL** : Connected Component Labelling, **MPG** : Max-Pooling based Grouping, **FG** : Feature Generation, **PSR** : Proposal Scoring Ranking)

on their edges. We adopted the Canny edge detection method [90] for generating a binary edge map. These edges could be a path of character, a character, a group of characters, a word, or text lines. Edges are then grouped together into proposals by an iterating max pooling process. Bounding box of each group is estimated by covering entire edges in a group. The proposal scoring/ranking strategy is proposed by using histogram of oriented gradients on edge pixels extracted from oriented gradient map as proposals' feature vectors and adopting Euclidean distance function for scoring. Scored proposals are then sorted in the descending order in which the high probability of text proposals are usually in the top of the list. Figure 4.1 depicts the proposed framework on an example image.

4.1.1 Max-pooling based scene text proposal generation

A max-pooling based grouping strategy is our novel contribution in this proposed scene text proposal technique. It is inspired by a pooling layer in a convolution neural network, which is used to eliminating insignificant features in a feature map while shrinking feature map size. In our framework, it is exploited on labelled edge maps which is generated from Canny binary edge maps by labelling connected components (CCs). Each CC is labelled by a number indicating when it has been found during

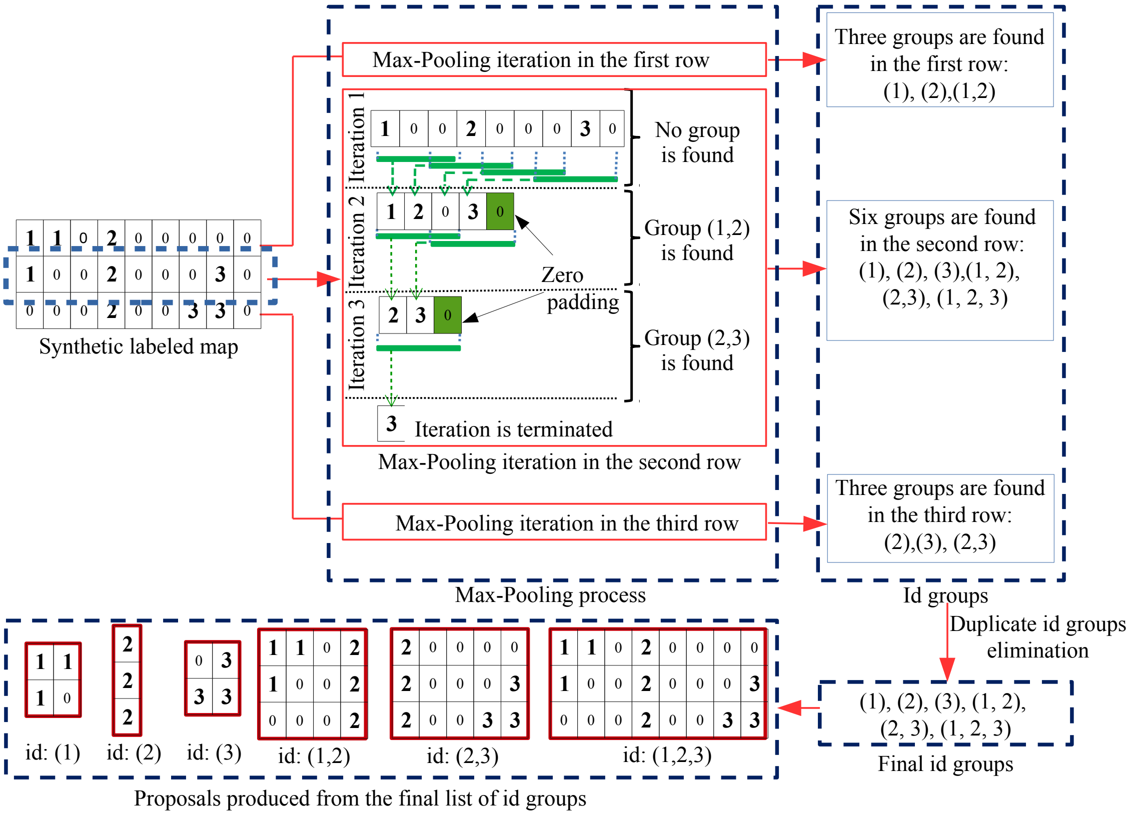
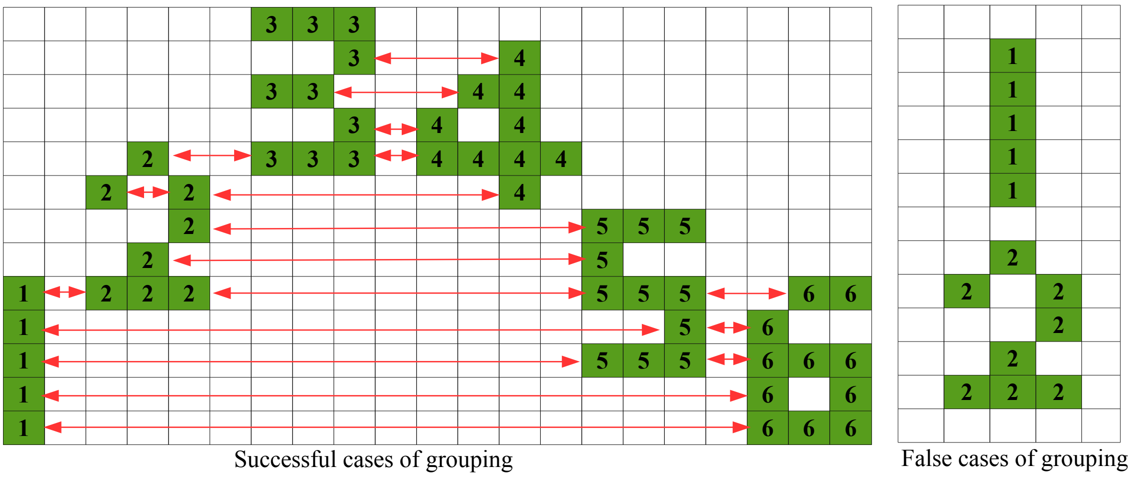

Max-Pooling iteration in the first row
Three groups are found in the first row: (1), (2),(1,2)
Iteration 1
No group is found
Iteration 2
Group (1,2) is found
Zero padding
Iteration 3
Group (2,3) is found
Six groups are found in the second row: (1), (2), (3),(1, 2), (2,3), (1, 2, 3)
Synthetic labeled map
Iteration is terminated
Max-Pooling iteration in the second row
Three groups are found in the third row: (2),(3), (2,3)
Max-Pooling iteration in the third row
Max-Pooling process
Id groups
Duplicate id groups elimination
(1), (2), (3), (1, 2), (2, 3), (1, 2, 3)
Final id groups
id: (1)
id: (2)
id: (3)
id: (1,2)
id: (2,3)
id: (1,2,3)
Proposals produced from the final list of id groups

Figure 4.2 – Illustration of the proposed max-pooling based scene text proposal generation with max-pooling window size of 1-by-3 and a stride of 1 in vertical direction and a stride of 2 in horizontal direction on the synthetic labelled map, detail in the second row. Zeros-padding was performed when the number of column is even. Duplicate proposals removed after the pooling process and six proposals have been generated, including three connected components itself and three found groups.

a CC searching process. For example, the first found CC is labelled by a number of 1. The number increases after finding a new CC, and the last found CC will be labelled by the highest number. In a labelled feature map, edge pixels in the same CC are assigned the same number as the label of that CC, and non-edge pixels are zeros. Under a given pooling window, only the highest labelled pixel is remained for generating a next labelled edge map, other lower labelled pixels are discarded, including zero-label pixels. Consequently, CCs are gradually shifted to each others. A pooling processes is exploited in multi-iteration. CCs that are neighbours to each



						3	3	3												
								3				4								
						3	3				4	4								
								3		4		4								
			2			3	3	3		4	4	4	4							
		2		2								4								
				2										5	5	5				
			2											5						
1		2	2	2										5	5	5			6	6
1																5		6		
1														5	5	5		6	6	6
1																		6		6
1																		6	6	6

Successful cases of grouping

		1		
		1		
		1		
		1		
		1		
		2		
	2		2	
			2	
		2		
	2	2	2	

False cases of grouping

FIGURE 4.3 – A synthetic example explains how horizontal pooling window and horizontal stride can group non-horizontal texts. The red two-direction arrows are links found in a max-pooling process to group connected components into proposals. This strategy is only false when connected components are not overlapping a part in vertical direction as shown in the right labelled map.

other are grouped first and further CCs are merged into groups at later pooling iterations. The iteration process terminates when there is no zero pixel existing in the labelled map, meaning that there is no more gap between CCs.

Figure 4.2 is a visualization of the max pooling process on a synthetic labelled edge map containing 3 CCs which are labelled by 1, 2, and 3. The zero-label pixels are non edge pixels and they are eliminated gradually after iterations. The closest CCs are grouped first and followed by the further CCs. For example, we have a group of CC1-CC2 at the second iteration and then a group of CC2-CC3 at the third iteration when we launch the pooling process on the second row of the labelled edge map. In order to make the max-pooling process work properly, we provide a zero-padding at the right most position when the number of columns is even. Since the max-pooling process finished, found groups are collected and duplicate ones are removed. Based on grouped CCs, bounding boxes are generated, which are the smallest rectangle boxes that cover whole edge pixels in corresponding groups.

Even-though a pooling and stride are in horizontal direction as 1-by-3 and [1,2] (row x column) respectively, the proposed max-pooling based grouping is able to capture non-horizontal texts that include CCs overlapping a path in the vertical direction. As the synthetic example is shown in Figure 4.3, a curve text (1-2-3-4-5-6) can be localized easily due to their vertical overlapping. CCs which do not contain vertical overlap also can be grouped by overlapping with other CCs proposal regions. For example, the CC1 and CC4 are grouped together because both CCs are grouped with the CC2. This strategy is only false when texts are totally vertical which is shown in the right synthetic labelled map.

4.1.2 Proposal ranking

Histogram of oriented gradient is a proper feature vector for object detection and classification [35, 68]. In the proposed ranking strategy, this concept has been adopted with a variance that an orientation histogram is estimated only on edge pixels, instead of whole pixels in the analysing areas. To clarify, we named it as Histogram of Oriented Gradient on edges (HoGe). Proposal scores are estimated based on correlation between their HoGe feature vector and text/nontext HoGe template feature vectors extracted from training sets, referring to Section 4.2 for a detail discussion how the templates were generated. The score function is shown in Equation 4.1.

$$s = \sum \frac{C_{i\_id} \cdot k^{C_{i\_idx}-1}}{D(C_i, F)} \tag{4.1}$$

$D(C_i, F)$ is a Euclidean distance between two vectors C_i and F, where C_i and F are a HoGe template vector i and a HoGe feature vector of a given proposal, respectively. The C_{i_id} is a class identification number of the template i. It is 1 for a text template and 0 for a non-text template. The C_{i_idx} is an index of the template i in the sorted templates list according to its Euclidean distance to the proposal feature vector F. The index are from N to 1 for templates that are from the farthest to the shortest distance to F, where N is the number of template vectors in both classes. The parameter k is a real number. It should be higher than 1 to ensure positive correlation between indexes and score. It has been set at 1.01 in the proposed technique. According to Equation 4.1, if the vector F is closer to text template vectors, the numerator will be larger and the denominator will be smaller.

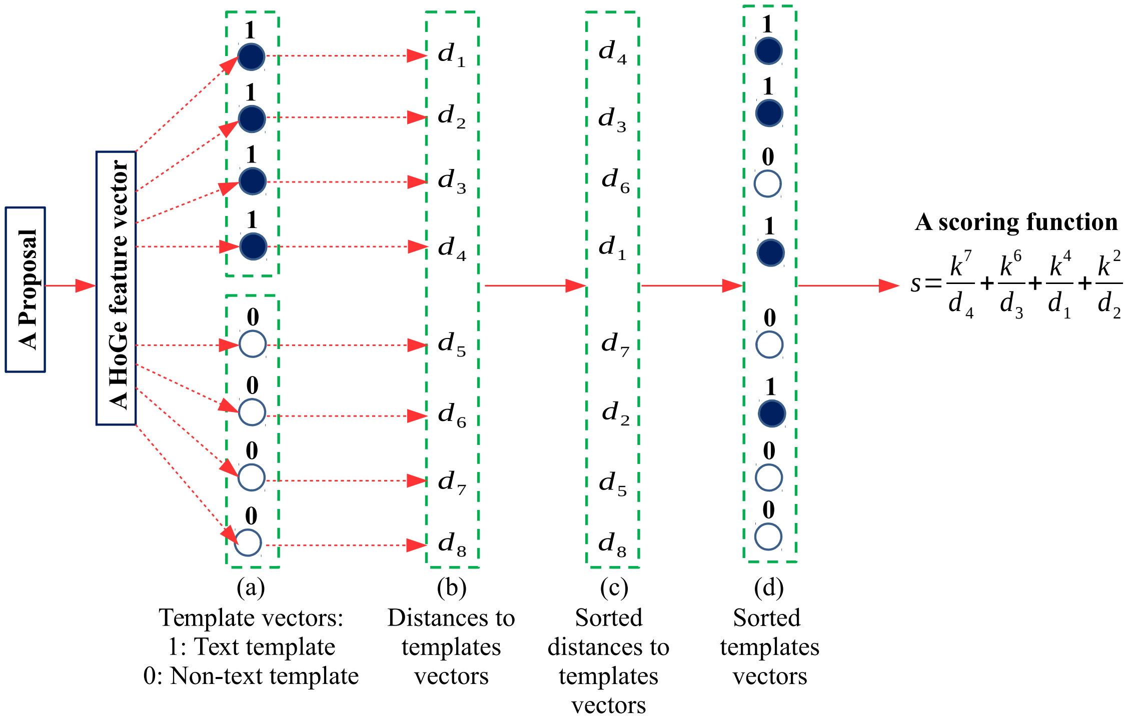


FIGURE 4.4 – Illustration of the designed scoring function. A proposal is ranked based on the distance between its feature vector and a set of pre-constructed text and non-text feature templates labelled by 1 (a text class) and 0 (a non-text class). These distances d_i are sorted in an increasing order (c) based on which the pre-constructed templates are then sorted (d)

It leads to a higher score to the evaluating proposal. A visualization of the proposed scoring function is shown in Figure 4.4 simplifying with 4 template vectors in each class.

4.1.3 Maxpooling based scene text proposal implementation

The MPT technique is implemented using Matlab for reading images, generating binary Canny edge maps, evaluating technique performances, and saving evaluated results, and using C++ for generating and scoring scene text proposals. Its main functions include the connected component grouping and the proposal boxes generator. The detail of the MPT class is depicted in Appendix. In the below MPT pseudo codes, Matlab based functions are highlighted in white backgrounds and C++ based functions are highlighted in gray backgrounds.

Algorithm : Max-pooling based scene text proposal (MPT)

```
//Import an image and text/non-text centroids
Img = imread('image address') ;
Centroids = load('centroids address') ;
//Create a binary edge map (E), an oriented gradient map (O)
[E,O,–] = CannyEdgeDetector(Img) ;
//Generate scene text proposals
 ppb = MPT(E,O,Centroids) ;
//Sort proposal boxes in the descending order
ppb = boxsort(ppb) ;
//Evaluate scene text proposal performance. The gtb is the ground truth of a corresponding image
Recall = PPBeval(ppb,gtb) ;
//Save evaluated performance
 save('Recall.mat','Recall') ;
```

Function : Max-pooling based grouping

```
//An input of the max-pooling grouping method is a labelled map (LabelledM) in which pixel is labelled as an ID of a connected component to which it belongs.
Maxdis = 1 ;
//Generate a list of pooling window
PWList = PoolingWindowGens(window_size, stride) ;
GroupedCCs = [ ] ;
While Maxdis > 0 :
    Maxdis = 0 ;
    //Generate TemM having a size calculated from a size of LabelledM and stride
va      -lues (Hstride, Vstride)
```

```
        TemM = MatrixInit(LabeledM.c/Hstride, LabeledM.h/Vstride) ;
        LocalGroupedCCs = [ ] ;
        for w in PWList :
                CropedM = Crop(LabeledM, w) ;
                [GCCs, localmaxdis, newpixels] = MaxPooling(CropedM) ;
                TemM[r, c] = newpixels ;
                LocalGroupedCCs.push_back(GCCs) ;
                Maxdis = max(Maxdis, localmaxdis) ;
        // Update LabelledM after max pooling process
        LabeledM = TemM ;
        //Eliminate duplicate groups of connected components
        ShortedList = Clean(LocalGroupedCCs) ;
        //Link connected components sharing the same connection
        LinkedList = Link(ShortedList) ;
        GroupedCCs.push_back(LinkedList) ;
 return(GroupedCCs) ;
```

Function : Proposal boxes generator

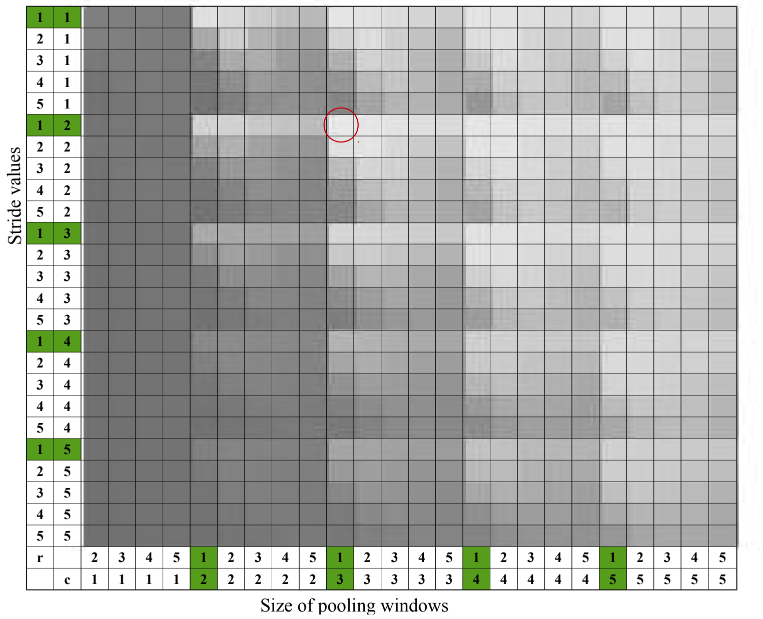
```
//Inputs of the proposal boxes generator function are a grouped connected component (GroupCCs) generated by the proposed grouping function, a labelled map (LabelledM), an oriented gradient map (O), and text/non-textt centroid (Centroids).
ppb = [ ] ;
for Group in GroupedCCs :
        HoGFV = HoGFeatureVectorExtract(Group, LabelledM, O) ;
        s = BoxScore(HoGFV, Centroids) ;
        [x, y, w, h] = BoxGens(Group) ;
        ppb.push_back([x, y, w, h, s]) ;
return(ppb) ;
```

4.2 Parameters optimization

There are four variable parameters in the proposed scene text proposal technique : size of pooling window, and stride values for the proposal generation, as well as a dimension of HoGe feature vectors and a number of templates in each class for the proposal ranking. They are essential parameters and impact significantly on performance of the proposed technique. The optimization process for these four parameters is simplified into two steps. The first is to optimize sizes of a pooling window and stride values to achieve the best recall rate without considering a number of proposals. The second is to optimize a number of dimensions of feature vectors and a number of templates to sort the best proposals to the top of a proposal list. Hence, the proposed technique can achieve the best performance under a limited number of proposals. It means that the first step focuses on providing as much as possible good proposals and the second step focusses on sorting proposals. Both steps are deployed on training images of the ICDAR2013 dataset [2] and the ICDAR2003 dataset [81]. The Street View Text (SVT) dataset [3] is not used because its ground truth boxes are not close enough to true text objects for the IoU based recall evaluation, as example images are shown in Figure 3.6. The following subsections discuss these two optimization processes in detail.

Proposal generation optimization

In this optimization process, a limitation on a number of proposals is relaxed. Whole generated proposals are collected for evaluation. The goal is to find the best combination between size of pooling window and strides that can provide the maximal number of good proposals. Since this process does not need a training phase, all training images in the two ICDAR datasets are utilized as a validation set, containing 479 images. The sizes of a pooling window and strides are varied from 1 to 5 in a row and a column. The pooling window size of 1×1 does not provide any grouping, hence this case is ignored. Therefore, there are (24x25=) 600 combinations evaluated. Figure 4.5 presents performance of the proposed proposal techniques in a heat-map, where columns present pooling window sizes, and rows present strides. Pixel values are average system's recall in three different IoU thresholds : 0.5, 0.7,



Stride values

1	1	87.38	89.34	89.19	88.84
1	2	81.68	**90.59**	89.53	88.33
1	3	66.12	83.74	87.95	86.73
1	4	53.75	70.29	81.05	84.79
1	5	50.54	64.49	75.83	81.43
r		1	1	1	1
	c	2	3	4	5

Size of pooling windows

IoU	0.5	0.7	0.8
Recall	98.1	91.77	81.9

Performance of the technique on training set with a stride of 1-by-2 and a pooling-window of 1-by-3

FIGURE 4.5 – The heat-map presents performances of the proposed proposal technique on the validation set constructed from training images of the two datasets : ICDAR2003 and ICDAR2013, at different combinations between sizes of pooling window and strides. The limitation on a number of proposals is relaxed in this evaluation. The heat-map shows that the optimal combination is a horizontal pooling window of 1×3 and a horizontal stride of [1,2].

and 0.8. As the illustration in Figure 4.5, peak values are always at the crossing positions of horizontal strides and horizontal pooling windows. We picked these pixels and filled in the top-right table. The best recall of 90.59% is found at the combination between the horizontal pooling window of 1×3, the horizontal stride of 2 and the vertical stride of 1. This is the optimal combination and is implemented in the proposed proposal technique. The recalls of the proposed technique at three different IoU thresholds on joined training images are presented in the bottom-right table in Figure 4.5.

Discussion : There are three essential factors in this technique affecting quality of generated proposals. The first is a pooling window size. It can refer to the smallest distance between connected components in a labelled map, which can be captured by the proposed technique. For example, a large pooling window such as 5×5 cannot find differences between distances of two, three, four pixels, while a small pooling window such as 2×2 can provide different groups for connected components having distances of one pixel first and later for connected components having distances of two pixels, three pixels and so on. The second is shape of pooling window and strides. The horizontal direction of both components is more productive than the vertical direction. It could be because most texts are in horizontal direction in causal scenes. As shown in the performance heat-map, the recalls of the proposed technique are extremely low at the vertical pooling window (the pooling windows have a width of one pixel). At fixed pooling windows width, the proposed technique's performance is usually decreasing following the increase of pooling window heights. These phenomena are similar to stride values. A recall rate is decreasing when a vertical stride turns larger than a horizontal stride and pooling windows are fixed. Furthermore, among horizontal pooling windows, larger-than-one-height windows can increase an ability to handle vertical texts which is a weakness of the proposed technique as mentioned in Section 4.1.1. However, it contemporarily merges text connected components with other non-text connected components in its vertical direction. It is conceivable to note that this capacity will pull recall rates down when there are a lot of non-text connected components around text connected components. It is the reason why one-height pooling windows usually provide the best recall rate at different window width. The third is the overlap between pooling windows in an iteration. The overlap is happen when stride values are smaller than pooling windows at corresponding directions. Usually, at a pooling widow size, the smaller stride provide the better recall rate, meaning that larger overlap between pooling windows provide the higher number of good proposals. In a case of non overlap, recall rates are extremely low in comparison with the best performance.

Proposal ranking optimization

This optimization step assumes that we already have the best proposal list and it tunes a number of dimensions of the HoGe feature vector and a number of templates in each class to search for optimal values that provide the best sorting solution giving the best recall rate under limited number of proposals. The training images are then

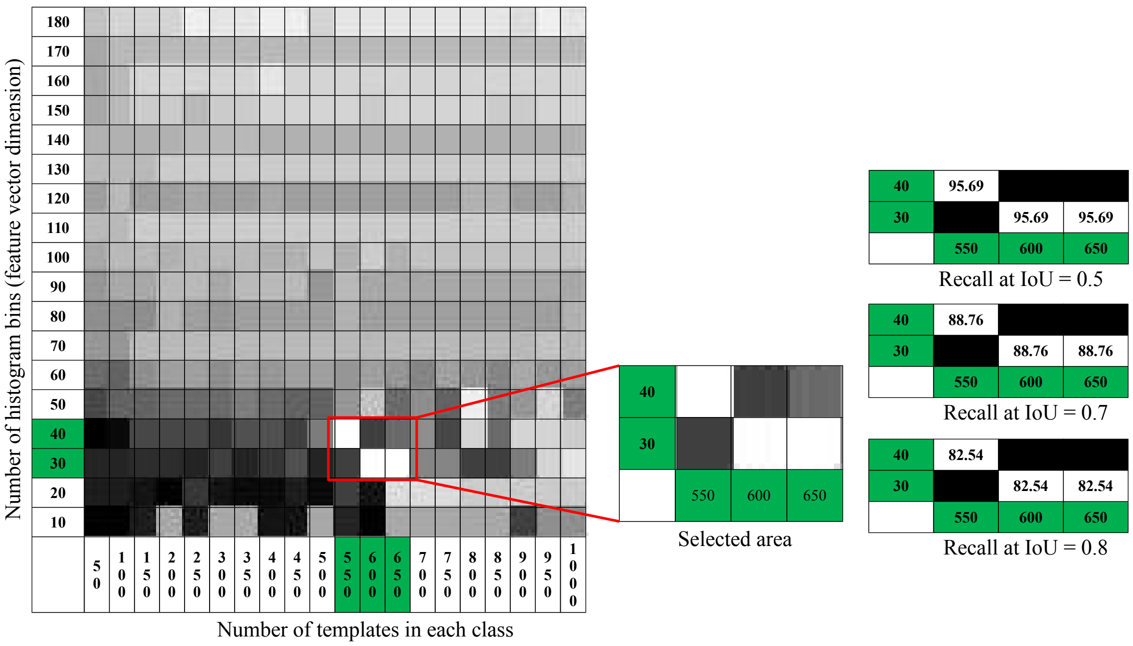


FIGURE 4.6 – The heat-map presents performances of the proposed technique on the evaluation set achieved from the ICDAR2013 and ICDAR2003 dataset on different combinations of a number of templates and a number of HoGe bins. Its contrast has been enhanced by subtracting pixel values by 0.9 and normalizing into a range of [0,1]. The best performance has been found when a number of templates and a number of HoGe bins are (600, 30), (650, 30), and (550, 40). Their recall rates at different IoU thresholds are presented in detail in the selected area.

divided into two sets. One is a training set which contains 80% number of training images. Remained training images are in a validation set. True text regions are located by ground truth boxes and good proposal boxes, which are generated by the proposed scene text proposal technique and have more than 50% overlapping with ground truth boxes. Non-text regions are proposal boxes that do not overlap with any ground truth box. From the training set, we create a training regions set. K-mean algorithm is applied on HoGe feature vectors of the training regions set to generate K centroids referring to feature vector templates. At a given validating image, proposal boxes are generated. Their HoGe feature vectors are extracted and distances to generated templates are measured to calculate their text confidence scores. Only 2000 proposal boxes in a sorted list are utilized to evaluate performance.

A number of bins in a histogram refers to a number of dimensions of the HoGe feature vector, and it is varied from 10 to 180 with an internal step of 10. A number of templates in each class is varied from 50 to 1000 with an internal step of 50. At each combination of two parameters, templates of each class are generated based on the training regions set and then performance of the proposed technique is validated in the validation set using recall rates at different IoU thresholds as 0.5, 0.7 and 0.8. Figure 4.6 shows a heat-map which is generated by system performances at these different combinations. Note that the heat-map has been enhanced contrast for better visualization by subtracting pixel values by 0.9 and normalizing into a range of [0,1]. Each pixel is an average performance of the proposed technique at three difference IoU thresholds. The whiter pixels present the better performances, and they are found at three combinations (600, 30), (650, 30), and (550, 40). The first number is a number of templates and the second number is a dimension of feature vectors. The combination of (600, 30) is selected to implement in the proposed technique due to the smallest number of computations.

4.3 Experiments and results

4.3.1 Evaluation set-up

The proposed max-pooling based scene text proposal technique is evaluated and compared to state-of-the-art techniques on the two public scene text datasets including the robust reading competition 2013 (ICDAR2013) [2] and the street view text (SVT) [3] which are described in Section 1.6.1. During our evaluation, there is only 2000 proposals selected due to a trade-off between system efficiency and accuracy mentioned in [79]. According to optimization in Section 4.2, a pooling window size of 1×3 and a stride of 1 and 2 in horizontal-vertical directions are set-up to generate proposals, and a feature dimension of 30 and a number of template of 600 are set-up to score proposals, in the proposed technique.

4.3.2 Comparing with state-of-the-art object proposal methods

Performance of the proposed technique is compared to state-of-the-art scene text proposal techniques, including Text Edge Box (TEB), Simple Selective Search for Text Proposal (TP) [19], Symmetry Text Line (STL) [24], DeepText (DT) [21], and Weakly Supervised Text Attention Network (WeakCNN) [31]. In addition, we also compare to state-of-the-art generic object proposal methods that are EdgeBox (EB) [26], Geodesic (GOP) [91], Randomized Prime (RP) [92], and Multiscale Combination Grouping (MCG) [93]. Parameters of those comparison techniques are adopted from their published codes and papers. Most of compared techniques is programmed in Matlab, excluding TP and STL which are implemented in C++. All techniques are executed in a HP workstation computer with Intel Xeon 3.5GHzx12 CPU, and 32 GB Ram.

Figure 4.7 shows comparisons in the variations of the number of proposals at different IoU threshold at 0.5, 0.7 and 0.8 (the left-column graphs) and the variations of an IoU threshold at a limitation on a number of proposals at 100, 500, and 1000 (the right-column graphs) on the ICDAR2013 dataset. In the left-column graphs, the proposed technique outperforms other techniques, excluding DT at IoU of 0.5, which was built on a deep learning frame work. In the right-column graphs, the DT technique is a competitive technique when the IoU threshold is in a range of (0.5, 0.6). However, its performance drops dramatically following an increasing IoU threshold. The TP, STL, and WeakCNN are more stable than the DT because of their included hand-craft text specific features. Nevertheless, their performances are lower than the proposed technique until the IoU threshold of 0.9, which is unusually used in real applications. The MPT also outperforms our previous technique as the TEB in different cases of number of proposals and IoU thresholds. Note that the DT technique only provides maximally 500 proposals per image. In those graphs, there is no generic object proposal technique exceeding performances of the proposed MPT technique.

We also evaluate and compare scene text proposal techniques based on an average number of proposals and processing time on the two testing datasets. Those results

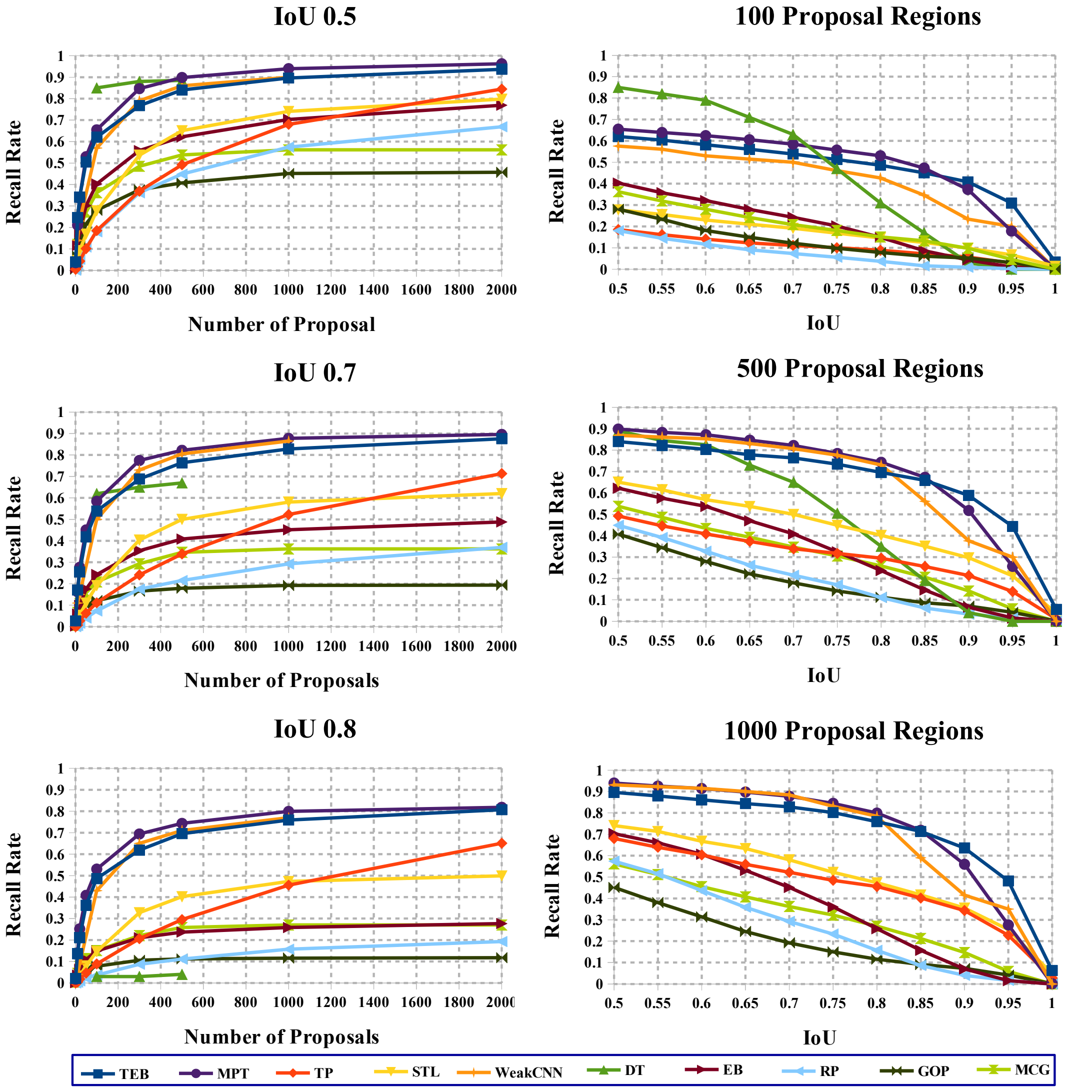


FIGURE 4.7 – Comparison of the proposed Max-pooling based scene text proposal technique (MPT) to other state-of-the-art techniques on the ICDAR2013 dataset [2], in difference of a number of proposals and IoU thresholds. Comparable techniques include both scene tex proposal techniques : Text Edge Boxes (TEB), Simple Selective Search for Scene Text Proposal (TP) [19], Symmetry Text Line (STL) [24], Deep Text (DT) [21], and Weakly Supervised Text Attention Network (WeakCNN) [31] ; and generic object proposal techniques : EdgeBoxes (EB) [26], Geodesic (GOP) [91], Randomized Prime (RP) [92], and Multiscale Combination Grouping (MCG) [93]

Table 4.1 – Recall (%) and processing time (in second) of the proposed MPT and other state-of-the-art techniques under different IoU thresholds on the ICDAR2013 dataset. The Nppb denotes an average number of proposals that the techniques need to achieve its presented recalls.

IoU	0.5	0.7	0.8	Nppb	times (s)
MPT	**96.16**	**88.77**	**82.1**	1450	4.49
TEB [25]	94.25	87.95	81.55	1777	5.4
TP [19]	84.47	71.32	65.11	1907	5.17
STL [24]	79.78	62.04	49.91	1034	361.3
DT [21]	88.5	67	4	**500**	–
WeakCNN [31]	90	86.5	77	1000	–
EB [26]	76.93	48.81	27.67	1968	**1.02**
GOP [91]	45.68	19.39	11.76	1040	4.3
RP [92]	66.91	36.95	19.3	1917	10.07
MCG [93]	56.16	36.21	27.02	550	28.92

are shown in Tables 4.1 and 4.2. The proposed MPT technique always achieves the best recall rate in most cases of IoU threshold on the two datasets. The TP is very competitive technique ; however, it requests a larger number of proposals to be comparable and processes slower than the proposed technique. The EB is the most efficient technique that need around a second to process an image. The DT and MCG provide the smallest number of proposals as around 500. Nevertheless, their performances are much lower than the MPT's performance.

Performances of the proposed technique and two other state-of-the-art scene text proposals on some example images from the SVT and ICDAR2013 datasets have been illustrated in Figure 4.8. In various challenges of text distortions such as text size variation (the first column), certain image blurs (the third column - the fourth column), uneven illumination (the second column), and non-horizontal orientation (the fifth column), the proposed technique always handles better than other competitive techniques. It may fails when text objects are ultra-low contrast, strong blur as showed in the right most image. These texts could be lost due to a edge detector capacities. Indexes of good proposals, which overlap with corresponding ground truth boxes over 80%, in the sorted proposal list are tracked and presented

Table 4.2 – Recall (%) and processing time (in second) of the proposed MPT and other state-of-the-art techniques under different IoU thresholds on the SVT dataset. The Nppb denotes an average number of proposals that the techniques need to achieve its presented recalls.

IoU	0.5	0.7	0.8	Nppb	times (s)
MPT	**88.41**	46.99	21.48	1769	3.59
TEB [25]	87.64	**47.91**	20.09	1890	2.2
TP [19]	74.65	42.5	21.33	1972	5.94
STL [24]	77.13	31.07	10.36	1358	433.82
EB [26]	76.35	47.45	**23.96**	2000	**1.28**
GOP [91]	52.09	18.24	6.8	1117	3.78
RP [92]	62.6	27.05	12.21	2000	8.02
MCG [93]	54.71	24.27	8.66	**557**	14.97

under each image. Note that there are many good proposals for a given ground truth box, and only the smallest index proposals are presented. In comparison with other techniques, the proposed technique usually sorts good proposals at the smallest order.

The most attribution of this impressive achieved performance is a grouping strategy that focuses on popular text appearance in scenes : characters usually stay in a group to form words or sentences, and internal distances between characters are usually smaller than distances between outer characters to other objects round texts. The proposed grouping strategy also supports linking broken connected components of the same text object. In comparison to another edge based proposal technique as EB [26], the two techniques have two similar processes that connected components are searched on an edge map, and bounding boxes are generated based on outer edge pixels of each group. The difference is only a grouping strategy and this difference provides a huge gap performance between two techniques. Due to a simple implementation that does not contain either distance calculation or threshold comparison, the proposal generation executes efficiently. The HoGe based proposal ranking participates to shift good proposals to the top of a sorted list. It is also a significant attribution for the state-of-the-art performance of the proposed technique when a

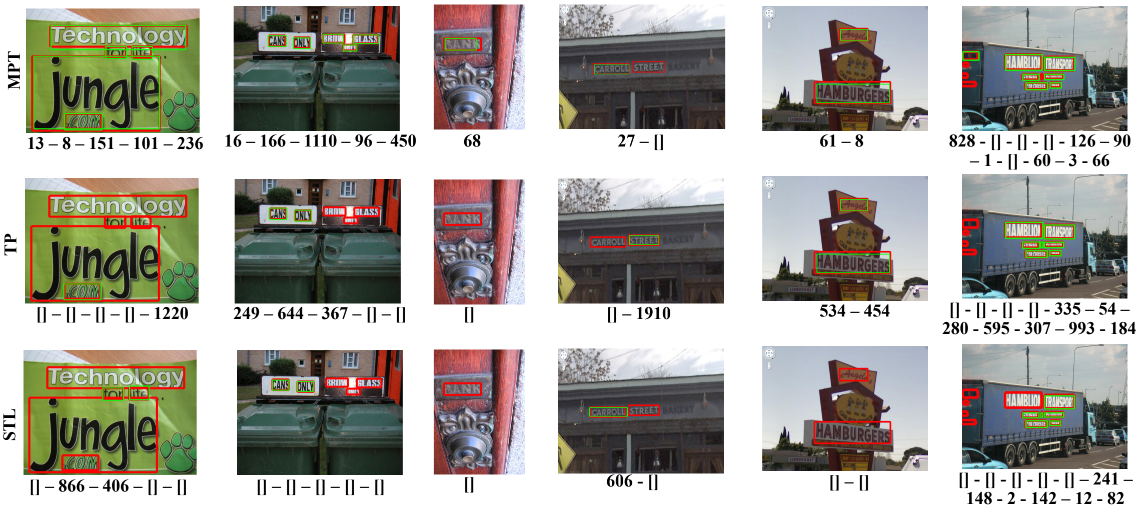


FIGURE 4.8 – Performance of the proposed technique MPT and two competitive techniques STL and TP on some images in the two generic scene text datasets. The numbers under each image are the smallest orders where good proposals are found in the list respecting to each ground truth. The good proposals are proposals that overlap with ground truth boxes over 80%. Noted that there are many good proposals respecting to a ground truth in the proposal list. However, only the top proposal respecting to each ground truth box in each image is shown because of illustration purpose.

number of proposals is limited.

4.4 Conclusion

This chapter presents a max-pooling based scene text proposal technique. The proposed technique is inspired by the CNN max-pooling layer, which is capable of grouping image edges into words and text lines accurately and efficiently. A novel score function is also designed, which is capable of ranking proposals according to their probabilities of being text and accordingly helps to reduce the number of false positive proposals greatly. Further, the proposed proposal technique does not rely on those heuristic thresholds/parameters such as text sizes, inter-character distances, and so on, which are widely used in many existing techniques. Extensive experiments show that the max-pooling based proposal technique achieves superior performance as compared with state-of-the-arts, including techniques developed based on the deep learning framework.

Chapter 5

Automatic scene text reading systems and applications

Contents

In the two preceding chapters, we proposed two state-of-the-art scene text proposal techniques including the Text Edge Box (TEB) and the Max Pooling Text (MPT), which are able to localize texts appearing in different fonts, sizes, colors and under different distortions due to environment conditions such as uneven light, occlusion, perspective, and so on. In this chapter, we are going to develop end-to-end scene text reading systems based on those proposed scene text proposal techniques. Further, we design an online scene text spotting system aiming to assist people in searching their keywords in given images which are captured or uploaded by their own devices.

A traditional framework of a scene text reading system consists of two main tasks as a scene text detection and a scene text recognition as described in Section 1.3. At least, two machine learning models are needed to handle these two tasks sequentially. It will consumes a huge processing time and computing resources. In recent years, a hybrid framework is a raising interest, and it is proposed to overcome this weakness [20, 19, 24], in which texts have been detected and recognized concurrently. Inspired by this advantage, the hybrid framework has been adopted to develop our scene text reading systems. State-of-the-art scene text recognition models are employed and built on top of scene text proposal techniques to clarify true text proposals from the rests and recognize actual words in proposals.

Designing an automatic scene text reading system is not only to evaluate influence of scene text proposal performances on performances of end-to-end scene text reading systems but also targeted to archive state-of-the-art scene text reading performance in competition with other existing systems. In fact, the proposed system outperforms other recent systems including hands-craft features based [20] and deep learning based [50, 76, 77] on public scene text datasets.

The online scene text searching application utilizes the proposed automatic scene text reading system as a back-end program. The interface between users and the back-end program has been designed using web design language. The application allows users to capture images using integrated cameras on their smart devices as input images and to provide keywords which they want to search on the captured images. In contextualization of searching targeted keywords, the proposed online scene text searching application is compared with the google translate application [5]. As comparison results, the proposed application outperforms the google application ; however, its processing time is less efficient than the application of google.

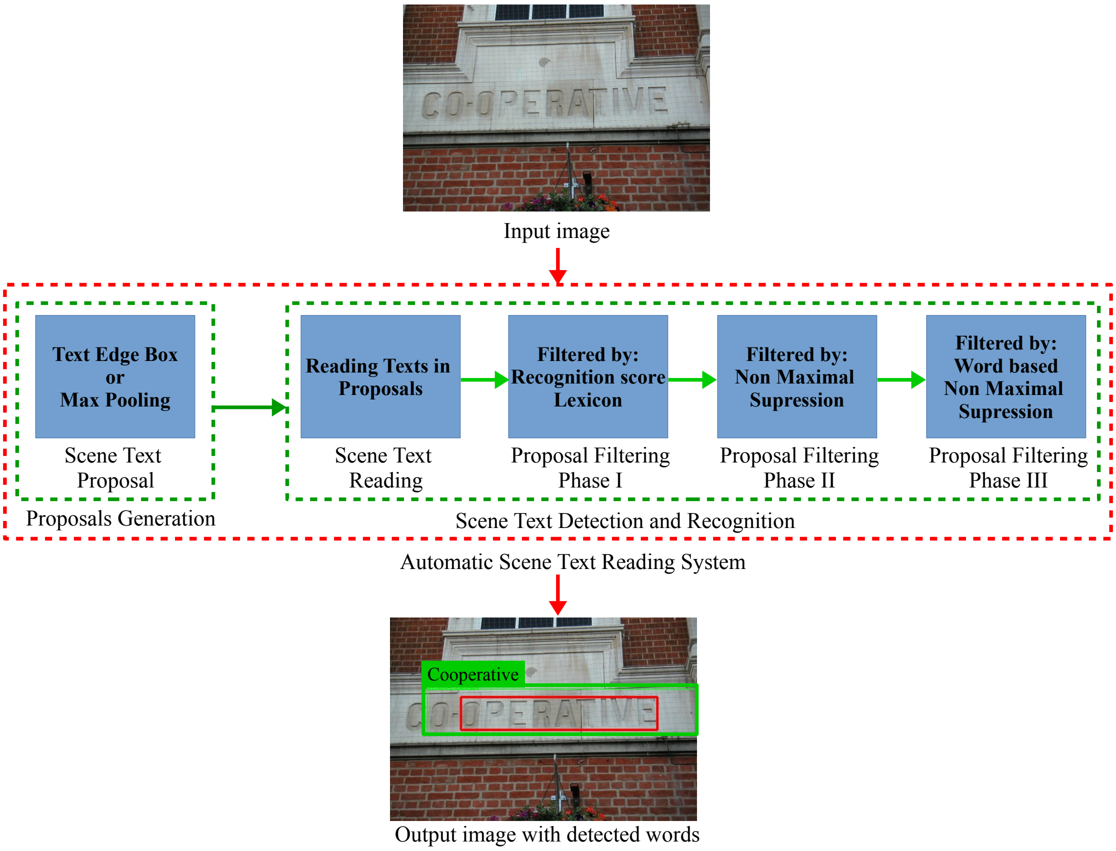

Input image
Text Edge Box or Max Pooling
Scene Text Proposal
Proposals Generation
Reading Texts in Proposals
Scene Text Reading
Filtered by: Recognition score Lexicon
Proposal Filtering Phase I
Filtered by: Non Maximal Supression
Proposal Filtering Phase II
Filtered by: Word based Non Maximal Supression
Proposal Filtering Phase III
Scene Text Detection and Recognition
Automatic Scene Text Reading System
Cooperative
Output image with detected words

Figure 5.1 – A framework of the proposed automatic scene text reading system consists of proposed scene text proposal techniques for searching text locations in scene images and the scene text detection/recognition for eliminating false positive proposals and reading actual words in proposals.

5.1 Automatic scene text reading system

5.1.1 System framework

In this section, we present our two proposed automatic scene text reading systems which are developed base on two proposed scene text proposal techniques. The state-of-the-art scene text recognition model provided by Jaderberg [78] is adopted to deal with both a text/non-text classification and a scene text recognition. The proposed framework has been shown in Figure 5.1. An input image is first fed into the proposed scene text proposals for searching potential text regions that have high probabilities of being text. Top n proposals in a sorted proposal list are collected and the rests are discarded. Collected proposals are then passed to the scene text recognition model for achieving recognized words and recognition scores. The proposal filter process includes three phases. In the first phase, non-text proposals are filtered out by using recognition score thresholds and lexicon lists, which are accompanied with images. Particularly, proposals are discarded if their recognition scores are lower than a threshold or their recognized words are not in the corresponding lexicon lists. In term of unavailable lexicon list, proposals are discarded using a recognition threshold only. In the second phase, the non-maximum suppression algorithm (NMS) is adopted to remove low recognition score proposals when they overlap with the higher recognition score ones larger than the Intersection over Union threshold (IoU). The smaller IoU threshold will let the proposed system remove the larger number of proposals. However, too small IoU threshold can degrade system performance because some true text proposals also can be eliminated when they overlap with other higher score true text proposals. In the proposed system, the IoU threshold is implemented at 0.1 for removing a vast number of non-text proposals while being able to maintain true text proposals. We also proposed a word-based NMS algorithm as the third phase of the proposal filter process. It considers proposals that overlap each others and contain the same recognized words, and a proposal that has the highest recognized score is maintained.

Table 5.1 – Recognition performance comparison between the Jaderderg's model and other deep learning based models [62, 64, 65] on three scene text datasets : SVT, IC03, and IC13 in different modes including a lexicon mode and a non-lexicon mode (xx-None). This comparison shows that the Jaderberg's model outperforms other models in the non-lexicon mode where recognized words are not corrected by lexicons.

Model	SVT-50	SVT-None	IC03-50	IC03-50k	IC03-None	IC13-None
Jar's model [66]	95.4	80.7	**98.7**	93.3	**93.1**	**90.8**
CRNN model [62]	96.4	80.7	98.7	95.5	89.4	86.7
TextAttCNN[65]	**97.4**	**82.7**	**98.7**	**96.7**	89.2	88
IrrTReader[64]	95.2	-	97.7	-	-	-

In designing this framework, we also consider to adopt other state-of-the-art scene text recognition models including the CRNN model [62], the IrregularTextReader [64], the CNN Text Attention model [65]. These models are developed base on the deep learning framework consisting of several convolution layers for converting input images into feature sequences, and bilateral Recurrent Neural Network (RNN) [64] or Long Short Term Memories (LSTM) [62] or another CNN model [65] for predicting a label distribution of each feature vector. Building on top of these models' architecture is a transcription layer which adopts the Connectionist Temporal Classification (CTC) network to convert sequences of label distributions into predicted words. In comparison with the Jaderber's recognition model, these models have an advantage that they can predict strange words which do not exist in the 90k-word dictionary on which they are trained because they treat an input image as a sequence of characters. However, this ability leads these models to be false in recognizing word due to wrong recognition, miss or false recognition of one or some characters in a word. It could be a reason why they achieve inferior performance to the Jardeberg's model in none-lexicon mode while provides superior performance in lexicon mode, as shown in Table 5.1. Therefore, the Jaderberg's model has been selected as a recognition model of the proposed end-to-end scene text reading system.

5.1.2 Scene text reading system implementation

The automatic scene text reading systems consists of a proposed scene text proposal technique (TEB or MPT) and the state-of-the-art scene text recognition models proposed by Jaderberg [66]. The system is almost implemented in Matlab, excluding the scene text proposal technique which was already developed in C++. The pseudo code of the proposed automatic scene text reading system is exposed below. **Lexicon** is a list of suggested words which are provided along with each image and used to guide the scene text recognition model to predict words.

```
System : An automatic scene text reading system
```

*//Inputs of the automatic scene text reading system are an image (**Img**) and a lexicon list (**Lexicon**) when it is available.*

```
Scale = [1, 0.5, 0.25, 0.125] ;
RecognizedThreshold = 0.7 ;
Net = GetRecognitionModel('Net address') ;
 Centroids = Load('Centroid address') ;
ppb = [ ] ;
for sc in Scale :
    S_Img = imresize(Img,sc) ;
    for k from 1 to 4 :
        if k < 4 :
            I = S_Img( :, :,k) ;
        else :
            I = rgb2gray(S_Img) ;
        [E,G,O] = CannyEdgeDetector(I) ;
        //Proposal generation
        if TEB :
            boxes = TEB(E, G, O) ;
        if MPT :
            boxes = MPT(E,O,Centroids) ;
        ppb = [ppb ;boxes] ;
ppb = boxsort(ppb) ;
//Get top 2000 proposal boxes
 ppb = ppb(min(2000,size(ppb,1)), :) ;
//Proposal recognition
PredictedWords = {} ;
Recognizedppb = [ ] ;
for box in ppb :
    CropedImg = ImageCrop(Img,box) ;
    [s, predictedword] = Net(CropedImg) ;
    Recognizedppb = [Recognizedppb ;[box[1 :4] s]] ;
    PredictedWords{end+1} = predictedwords ;
//Eliminate non-text proposals
```

```
Keptppb = [ ] ;
KeptWords = { } ;
for id from 1 to length(PredictedWords) :
    word = PredictedWords{id} ;
    box = Recognizedppb[id, :] ;
    if Lexicon is not None :
        if word is in Lexicon and box[5]>RecognizedThreshold :
            KeptWords{end+1} = word ;
            Keptppb = [Keeppb ;box] ;
    else :
        if box[5]>RecognizedThreshold :
            KeptWords{end+1} = word ;
             Keptppb = [Keeppb ;box] ;

//Non maximal suppression (NMS)
[Keptppb,KeptWords] = NMS(Keptppb,KeptWords) ;
[Keptppb,KeptWords] = WordBasedNMS(Keptppb,KeptWords) ;
return(Keptppb,KeptWords) ;
```

5.1.3 Evaluation

Variants of the proposed systems

The recognition score threshold is an essential parameter of the proposed automatic scene text reading system. The higher threshold is better to filter out false positive proposals, however, it will degrade system performance in localizing challenging texts in scenes. We take this parameter in our consideration and evaluate its contribution to performance of our two proposed scene text reading systems, including a TEB based and a MPT based.

We vary the threshold of recognition score in a range of (0.1, 0.9) with an internal step of 0.1. Performances of two scene text proposal based scene text reading systems are evaluated on the ICDAR2013 dataset and presented in Figure 5.2. The F-measurement is calculated base on system precision and system recall as shown

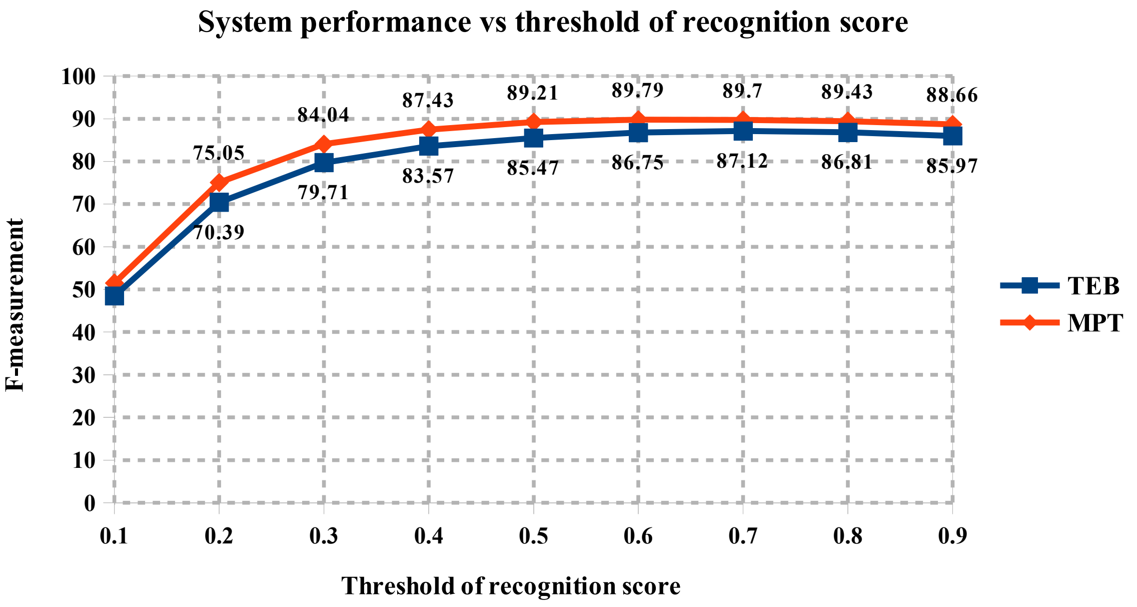


Figure 5.2 – Variants of the proposed scene text reading system under different thresholds of recognition score and scene text proposal techniques. System performances are evaluated on the ICDAR2013 dataset. The best variant is the system developed based on max-pooling scene text proposal (MPT-based) at the threshold of 0.7.

in equation 1.1. The best performance is found around the threshold of 0.7. The increment of system performance in the range of (0.1,0.7) is because that system precision improvement is much faster than system recall degradation. On the other hand, system performance decreases in the range of (0.7, 0.9) due to effect of recall degradation. In the full range of the recognition score threshold, the MPT based scene text reading system always outperforms the TEB based system. Therefore, in comparisons with recent state-of-the-art systems, the MPT based system with the threshold of recognition score of 0.7 is utilized.

Comparison with state-of-the-art systems

In order to analyse the influence of scene text proposal performance on the end-to-end scene text reading performance, we provided a series of scene text proposals based systems which are constructed by applying our proposed automatic scene text reading framework on the state-of-the-art scene text proposal techniques including

MPT, TEB, TP, STL, EB. Their performances are evaluated by using the ICDAR competition contextualizations including a word spotting evaluation and an end-to-end evaluation as discuss in Section 1.6.3. These performances are presented in Tables 5.2 and 5.3 for a word spotting systems and an end-to-end systems respectively. In this comparison, scene text proposal based systems are named following the template ***[Scene Text Proposal method]_Sys***.

Table 5.2 – Word spotting performance of the proposed automatic scene text reading system MPT-based and other scene text proposals based systems on Robust Reading Competition 2013 Dataset (ICDAR2013) in the two contextualizations.

	Strong contextualization			Weak contextualization		
	Recall	Precision	F-score	Recall	Precision	F-score
EB_sys	55.26	66.81	60.49	54.79	57.48	56.10
STL_sys	61.8	85.32	71.68	61.45	81.30	69.99
TP_sys	66.47	89.47	79.27	65.07	82.40	72.72
TEB_sys	84.46	96.4	90.03	82.94	91.96	87.22
MPT_sys	**88.20**	**97.42**	**92.58**	**87.85**	**95.31**	**91.43**

Table 5.3 – End-to-End performance of the proposed automatic scene text reading system MPT_based and other scene text proposals based systems on Robust Reading Competition 2013 Dataset (ICDAR2013) in the two contextualizations.

	Strong contextualization			Weak contextualization		
	Recall	Precision	F-score	Recall	Precision	F-score
EB_sys	52.67	65.80	58.51	52.24	56.69	54.37
STL_sys	59.65	85.70	70.13	59.43	81.10	68.60
TP_sys	63.90	88.12	74.08	62.70	81.21	70.77
TEB_sys	80.80	94.51	87.12	79.50	90	84.42
MPT_sys	**84.08**	**96.13**	**89.70**	**83.86**	**93.89**	**88.59**

According performances shown in the two tables, the two systems developed based on the two proposed scene text proposal techniques (MPT and TEB) outperform other scene text proposal based systems. Recalling scene text proposal performances in Section 4.3.2, there is a correlation between scene text proposal performances

Table 5.4 – Word spotting performance of the proposed automatic scene text reading system (MPT_based) and other state-of-the-art systems on the two scene text datasets including the Street View Text with 50-word lexicon per an image (SVT-50) and Robust Reading Competition 2013 Dataset (ICDAR2013) in the two contextualizations.

	SVT-50	Strong contextualization			Weak contextualization		
		Recall	Precision	F-score	Recall	Precision	F-score
JarE2E [20]	68	86.68	94.64	90.49	-	-	-
ConvLSTM [76]	-	84.93	**98.91**	91.39	84	**97.29**	90.16
DTSpotter [77]	-	-	-	92.89	-	-	89
TextBoxes [50]	84	**90.77**	97.25	**93.90**	87.38	97.02	**91.95**
MPT_sys	**84.63**	88.20	97.42	92.58	**87.85**	95.31	91.43

and scene text reading performances where the better scene text proposal technique usually provides the better automated scene text reading system.

The best proposed automatic scene text reading system as the MPT based system is then compared with state-of-the-art systems including the convLSTM [76], the DeepTextSpotter [77], the TextBoxes [50] and the Jar_E2E [20]. The TextBoxes system is the most state-of-the-art system in the scene text reading competition 2017 (http://rrc.cvc.uab.es/ ?ch=2&com=evaluation&task=4). It is a cascade of two deep learning models : one is a scene text localization model inspired by the SSD network [51] and one is a scene text recognition model developed by combining a convolution neural network, a long-short term memory model and a connectionist temporal classification network. The convLSTM system applies a CNN based character recognition model on multi-scale input images to generate character saliency maps based on which licence plates are detected and its' bounding boxes are refined based on edge information. Characters in detected plates are finally recognized by a CNN model consists of CNN layers for feature extraction and LSTM layers for sequence recognition. The DeepTextSpotter is a trainable scene text reading system which combines a scene text detection and a scene text recognition into a single CNN network. The Jar_E2E system has been developed following the traditional scheme that includes a scene text detection task and a scene text recognition task. The scene

TABLE 5.5 – End-to-end performance of the proposed automatic scene text reading system (MPT_based) and other state-of-the-art systems on the Robust Reading Competition 2013 Dataset (ICDAR2013) in the two contextualizations.

	Strong contextualization			Weak contextualization		
	Recall	Precision	F-score	Recall	Precision	F-score
JarE2E [20]	82.12	91.05	86.35	-	-	-
ConvLSTM [76]	79.39	**96.68**	87.19	79.28	94.91	86.69
DTSpotter[77]	-	-	89	-	-	86
TextBoxes [50]	**87.68**	95.83	**91.57**	**84.51**	**95.44**	**89.65**
MPT_sys	84.08	96.13	89.70	83.86	93.89	88.59

text detection is developed by combining two scene text proposal techniques including EdgeBoxes [26] and a Aggregate Channel Features based scene text proposal for providing potential text regions. Histogram of gradient features are extracted at each region and the random forest classifier is adopted for the text/non-text classification. Two deep learning based scene text recognition models are finally applied, one for regressing bounding boxes and one for recognizing words in detected areas. Please refer to section 2.2 for more details about these state-of-the-art systems.

As shown performance in two Tables 5.4 and 5.5, the proposed scene text reading system provides competitive performances to state-of-the-art systems, even provides superior performance to CNN based systems as DeepTextSpotter and convLSTM in F-score measurement in two datasets. The TextBoxes is the most competitive system where it provides better performance in almost measurements. However, it provides less recall rate than the proposed system in the weak contextualization scenario of the word spotting evaluation. Note that CNN based systems are trained on a large number of images including 800.000 images in SynthText [82], 90k word images in the Synthetic text dataset [78], and training images in the ICDAR2011, the ICDAR2013, and the Street View Text, while the proposed system is only optimized using 479 images in the ICDAR2003 and ICDAR2013 datasets.

Figure 5.3 shows a number of sample images that illustrate the performance of our developed end-to-end scene text reading system. As Figure 5.3 shows, the proposed technique is capable of detecting and recognizing challenging texts with

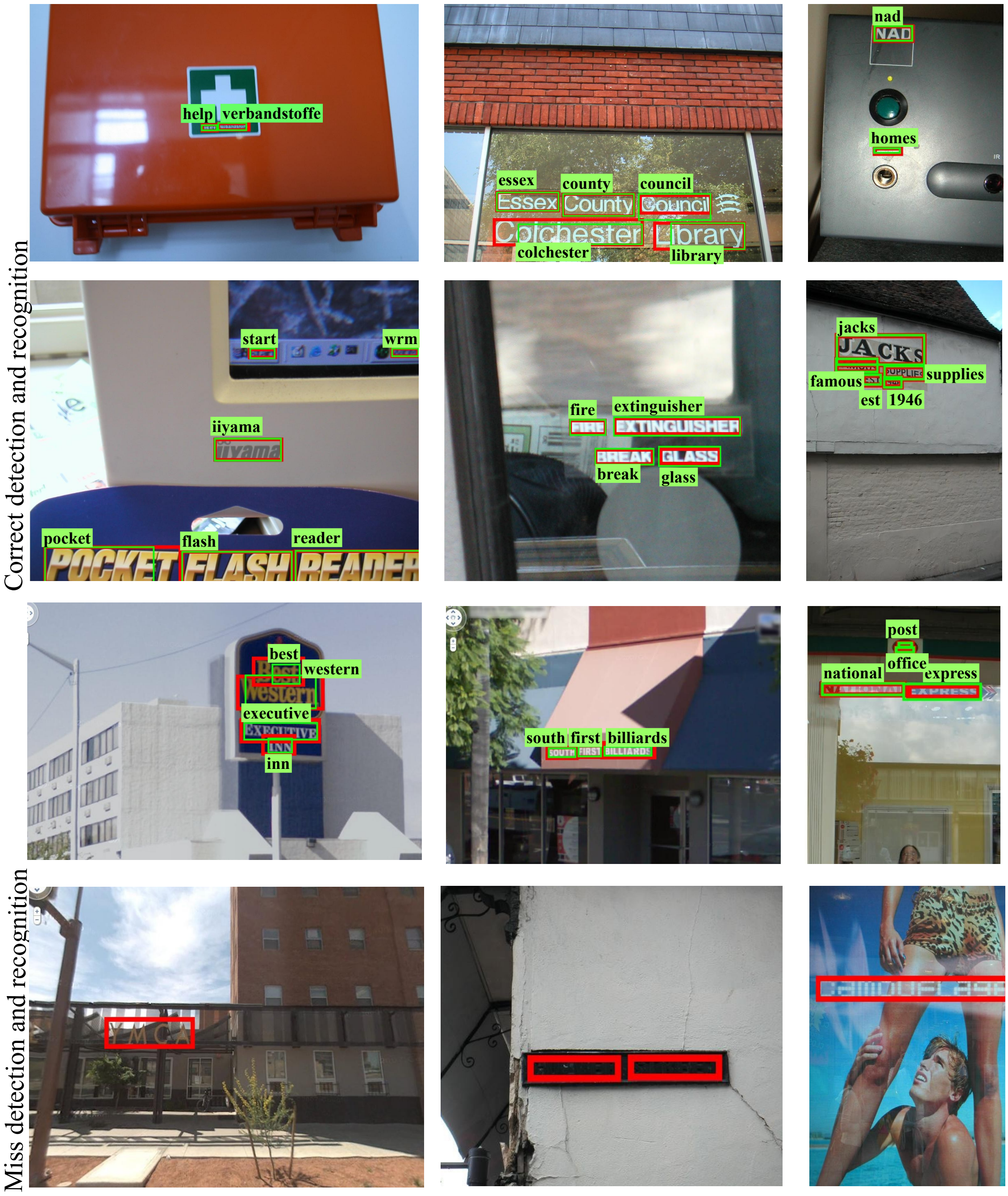
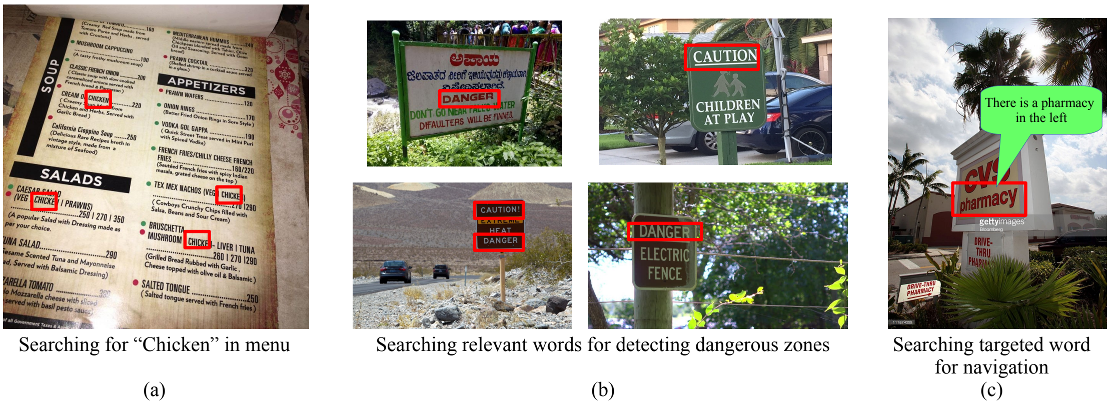


FIGURE 5.3 – Several scene text detection and recognition examples where the proposed scene text reading system succeeds (the first three rows) and fails (the last row) : The red boxes are ground truth and the green boxes are detection boxes provided by our proposed technique. The boxes with green-color background give the recognition results (words containing less than three characters are ignored)



Searching for “Chicken” in menu

(a)

Searching relevant words for detecting dangerous zones

(b)

Searching targeted word for navigation

(c)

Figure 5.4 – Several meaningful applications for a scene text searching application including a searching keywords in menus (a), a danger alarm (b), and a navigation (c).

small text size (the first image in the first row), poor illumination and motion blur (the second images in the first and second rows), perspective distortion (the second image in the third row and the third image in second row). The superior scene text reading performance is largely due to the robustness of the proposed scene text proposal technique and the integrated scene text recognition model. Note that the proposed technique may fail when scene texts have ultra-low contrast or are printed in certain odd styles as illustrated in the sample images in the last row.

5.2 Online scene text searching application

Scene texts based applications have been taken in consideration for aiming human life activities as discussions in Section 1.1 due to containing rich semantic context information. The Orcam product is developed to support visual impairment patients in reading books, magazines, menus, and so on. Google and Microsoft translators are very useful for tourists in understanding local or unlearned languages.

Inspired by the usefulness of texts searching tools in localizing keywords in documents, which supports users in finding quickly relevant paragraphs in documents, we are intent on developing a similar text searching tool but it is applied for scene texts instead. It could be useful for quickly pointing out relevant regions in images, from which related information can be extracted and applied for semantic analysis systems such as scene understanding, navigation, and so on. For example, when customers are reading a restaurant menu and looking for specific foods such as chickens, the developed system will capture an image of the menu, search for the word "chicken", then localize where word "chicken" exist. Users then can find other relevant information such as dish names, food recipe as well as its price (Figure 5.4 (a)). For another example, we need a system that can detect dangerous zones and notify users in advance. "Dangerous", "Caution", "Stop", and so on will be relevant keywords which a danger detection system has to search in captured images. This system will be useful for people who need special cares such as dementia patients or elderly (Figure 5.4(b)). The scene text searching system is also useful for navigation which can be used to estimate direction based on detected texts and GPS data (Figure 5.4(c))

In this Section, we will describe our proposed scene text searching system developed base on our proposed automated scene text reading system, which is used as a back-end program. It is evaluated based on images captured by users and compared with the Google translator application in contextualization of spotting keywords in captured scenes. The application can be accessed by a web browser application in any smart devices such as smart phones, tablets, and computers at the link : *dinh.ubismart.org:27790*. The developed application can be further integrated in another meaningful application, for example one has been described in Section 6.2.3 (Future Work).

5.2.1 System framework

Nowadays, smart devices such as mobile phones, tablets which have built-in cameras become more and more popular and people is easy to capture their own photos. In addition, internet network is more and more easy to access. Based on this infrastructure, we adopt the server/client framework to deploy our proposed scene text searching application. It is because the back-end program requires a high computing system which mobile device hardware is not satisfied. The framework of our proposed application is shown in Figure 5.5. Scene images are captured by

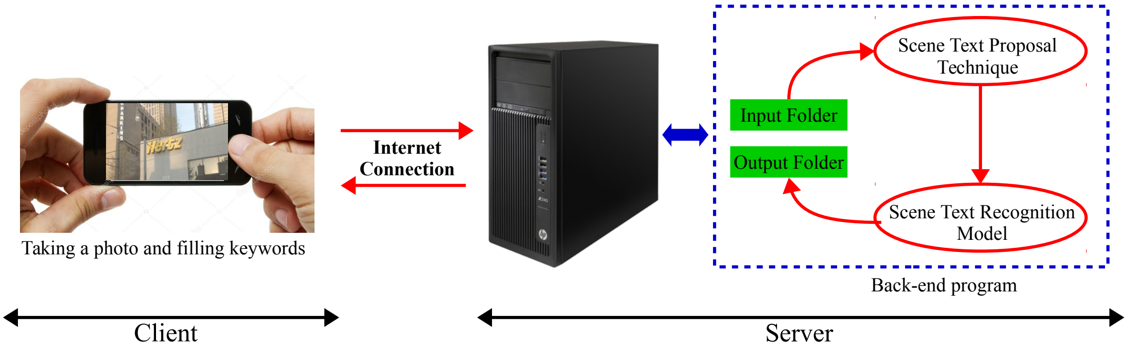
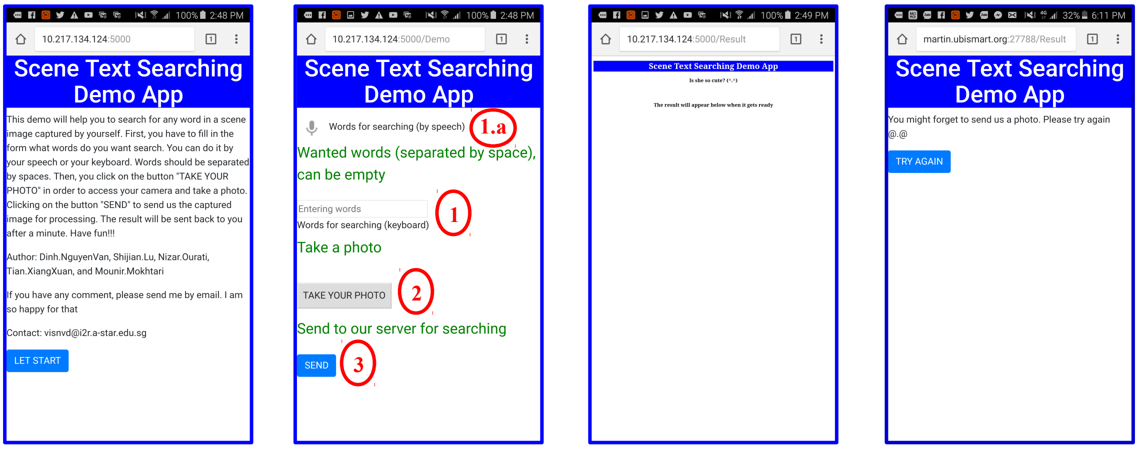

Taking a photo and filling keywords
Internet Connection
Input Folder
Output Folder
Scene Text Proposal Technique
Scene Text Recognition Model
Back-end program
Client
Server

FIGURE 5.5 – A framework of the proposed scene text searching application based on a client/server architecture. Clients can use a web browser application installed in their smart devices to access our server for sending images and keywords and receiving results. The proposed automatic scene text reading system is implemented on a server as a back-end program to process clients' requests.

using built-in cameras in client devices and keywords are provided by filling in text-boxes or speaking to micro-phones. Captured images and keywords are then sent to our server where the back-end program is executed. Detected texts are drawn on captured images and returned to customers. The proposed application is developed base on the web design platform and can be accessed by using web browsers. There is advantages that it becomes independent of operation systems installed on customers' devices such as window, ubuntu, android, iOS, and so on, and customers do not need to install additional packages on their devices.

Web interface

The interface pages of the proposed application is presented in Figure 5.6 including four pages : an introduction page which explains how the proposed application work, a data acquisition page where users process to take a photo and provide keywords, a result page that shows system outputs, and an error page that gives suggestions of errors. In order to provide keywords using speech, web-browsers in customers' devices have to support the google speech recognition system. In fact, they have to use the chrome web-browser with a version over 25 or the safari web-browser (for devices running iOS). Customers can activate this function by pressing



10.217.134.124:5000

Scene Text Searching Demo App

This demo will help you to search for any word in a scene image captured by yourself. First, you have to fill in the form what words do you want search. You can do it by your speech or your keyboard. Words should be separated by spaces. Then, you click on the button "TAKE YOUR PHOTO" in order to access your camera and take a photo. Clicking on the button "SEND" to send us the captured image for processing. The result will be sent back to you after a minute. Have fun!!!

Author: Dinh.NguyenVan, Shijian.Lu, Nizar.Ourati, Tian.XiangXuan, and Mounir.Mokhtari

If you have any comment, please send me by email. I am so happy for that

Contact: visnvd@i2r.a-star.edu.sg

(a)
Introduction page

10.217.134.124:5000/Demo

Scene Text Searching Demo App

Words for searching (by speech)

Wanted words (separated by space), can be empty

Entering words

Words for searching (keyboard)

Take a photo

Send to our server for searching

(b)
Data acquisition page

10.217.134.124:5000/Result

Scene Text Searching Demo App

Is she so cute? (^.^)

The result will appear below when it gets ready

(c)
Result page

martin.ubismart.org:27788/Result

Scene Text Searching Demo App

You might forget to send us a photo. Please try again @.@

(d)
Error page

FIGURE 5.6 – Web interface of the proposed application including four pages : an introduction page, a data collection page, a result page and an error page. Button 1.a is to activate the speech recognition function. Text box 1 is to fill keywords or to correct recognized keywords from the speech recognition function. Button 2 and 3 are to take a photo and send request to server. The error page is used to notify users that they forgot to select a photo before sending and the button "TRY AGAIN" helps them to navigate back to the data collection page.

on the micro-phone button (Figure 5.6b(1.a)) in the data collection page. Recognized keywords from speeches are automatically filled in the keywords text-box. Users also can modify those recognized words by touching on the keywords text-box for activating a keyboard which is used to rewrite/provide keywords (Figure 5.6b(1)). For capturing an image, users can process through the "TAKE YOUR PHOTO" button which support accessing cameras functions in their devices (Figure 5.6b(2)). For their convenience, image orientation has also been recorded for refining image later. Users do not need to have too much experience in capturing fine images. When they touch on the "SEND" button (Figure 5.6b(3)), all required data including an image, keywords, and an image orientation are packed into a form and sent to server. Concurrently, web browser switches to the result page and waits for the returned image (Figure 5.6c). If users forget to take a photo, the error page will be shown up to remind them that they need to provide an input image (Figure 5.6d). The

"TRY AGAIN" button is provided to help them go back to the data collection page. If users forget to provide keywords, the proposed application still works properly ; however, it will shown up all detected words in upload images.

Back-end program

The back-end system of the proposed scene text searching application is the proposed automatic scene text reading system which has been built by integrating our proposed scene text proposal technique with an existing scene text recognition system. The RCNN scene text recognition model [62] which is presented in Section 2.2.2 is adopted for this back-end system instead of the Jaderberg's model [78]. Despite this change can degrade system performance of finding common words (which are in 90k-word dictionary used to train scene text recognition models), it widens a chance of finding untrained words, especially non English words in Latin-character based languages such as French, German, and so on because it treats word images as character sequences. The conditional probability of a predicted word is considered as a recognition score, and the rest steps of the proposed scene text reading system including a recognition score based proposal filter, a non maximum suppression, and a word based non maximum suppression are maintained as described in Chapter 5

5.2.2 Scene text searching application implementation

The online scene text searching application is implemented in Python and C++ programming languages. It consists of two main parts : a web browser interface and a back-end scene text spotting system. The web browser interface provides users a convenient method to access our application. The flask python package which can interact with html, css and Java-script languages is adopted to design our web pages including a welcome page, a data collection page, a result page, and a fault alert page. This package also handles a submitting action in which users can send to our server their keywords and captured images. The pseudo codes of the web browser interface are presented below. The back-end scene text spotting system is implemented similarly to the proposed automatic scene text reading system, but in Python programming language instead of Matlab language.

System : Web interface of the online scene text searching application

```
//Call a welcome page. It is executed when users access the application link
def welcome() :
        return render_template('Welcome.html')
//Call a data collected page. It is executed when users press a button in the welcome
// page
def Demo() :
        return render_template('Demo.html')
//Call a data collected page. It is executed when users press a button in the fault
//alert page
def Comeback() :
        return render_template('Demo.html')
//Call a uploading task. It is executed when users press the button "SEND" in the
// data collection page
def upload() :
       try :
              //'ImageOrientation' is a parameter generated by the data collection page
              ImageOrientation = request.form('ImageOrientation')
              //Write the image orientation to a txt file which later is accessed by the
              // back-end program
              txtWrite(ImageOrientation,'Orientation.txt')
              //'Keywords' is a parameter generated by the data collection page contai
              // -ning keywords provided by users
              keywords = request.form('Keywords')
              //Write the keywords to a txt file which later is accessed by the back-end
              // program
              txtWrite(Keywords,'Keywords.txt')
              //Get an uploaded image
              Images = request.files.getlist('file') :
```

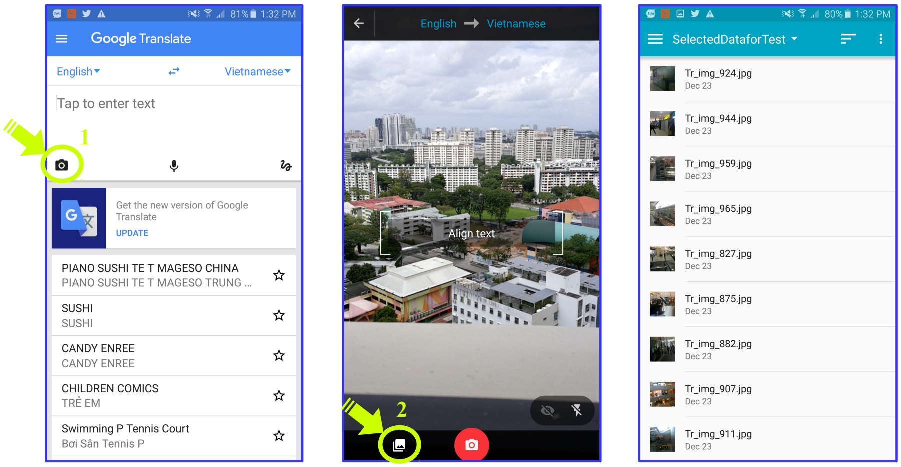
```
        for I in Images :
            //Save images to files. They are later accessed by the back-end pro
            //gram
             I.save('File name')

        //Call a result page in which the back-end program outputs will be shown
        return render_template('Result.html')
    except :
        //Call a fault alert page giving a suggestion that users forgot to provide
        //an image
        return render_template('Error.html')
```

5.2.3 Evaluation

In this evaluation, the proposed application has been evaluated on a set of images collected from the incidental scene text dataset [1], which are captured by using a google glass without focussing on scene text objects. In evaluation, we simulate human behaviour when they are going to search for specific words in natural scenes. Each image is considered as a captured image taken by users and corresponding keywords are provided. A fraction of keywords which are found successfully is used to evaluate application performance. Following this evaluation scenario, the proposed application is compared with the GoogleTrans application [5] which has been developed to read texts in scene images and released onto the Google Play store and the Apple store.

From the incidental scene text dataset [1] which contains 1500 images including training images and testing images, 142 images are selected to build a dataset for this evaluation. Selected images contain texts suffering a lot of degradation types such as motion blur, non-horizontal orientation, decoration, perspective, illumination and so on. Contained texts are mostly shops and restaurants name, traffic panels as well as words referring dangerous and cautious situations which are useful for navigation or danger detection applications. 217 keywords have been assigned for this dataset provided by volunteers who look at images and give keywords they are interested



PIANO SUSHI TE T MAGESO CHINA
PIANO SUSHI TE T MAGESO TRUNG ...

SUSHI
SUSHI

CANDY ENREE
CANDY ENREE

CHILDREN COMICS
TRẺ EM

Swimming P Tennis Court
Bơi Sân Tennis P

Tr_img_924.jpg	Dec 23
Tr_img_944.jpg	Dec 23
Tr_img_959.jpg	Dec 23
Tr_img_965.jpg	Dec 23
Tr_img_827.jpg	Dec 23
Tr_img_875.jpg	Dec 23
Tr_img_882.jpg	Dec 23
Tr_img_907.jpg	Dec 23
Tr_img_911.jpg	Dec 23

FIGURE 5.7 – GoogleTrans interface and a solution to evaluate GoogleTrans performance on our dataset. Step-by-step from 1 to 2 is a path to upload collected images to the google search engine.

in.

In evaluation, images and keywords are provided to our proposed application by using our data collection page. To evaluate the google translation application, images are uploaded to the google scene text reading engine using the image upload option as shown in Figure 5.7 following step 1 and 2 sequentially. For each image, corresponding keywords are decided as successful search if and only if the keywords are exist in detected words provided by this app.

TABLE 5.6 – Scene text searching performance and processing time of the proposed application and the GoogleTrans app.

	Proposed app	GoogleTrans
Performance	48.7%	39.48%
Time processing(second)	17.98	2.34

Table 5.6 presents performances of these two applications. As the results shown in Table, the proposed scene text searching application performs better than GoogleTrans application with 8.22% improvement. However, it is less efficient than the GoogleTrans application. The main reason is because our proposed application ana-

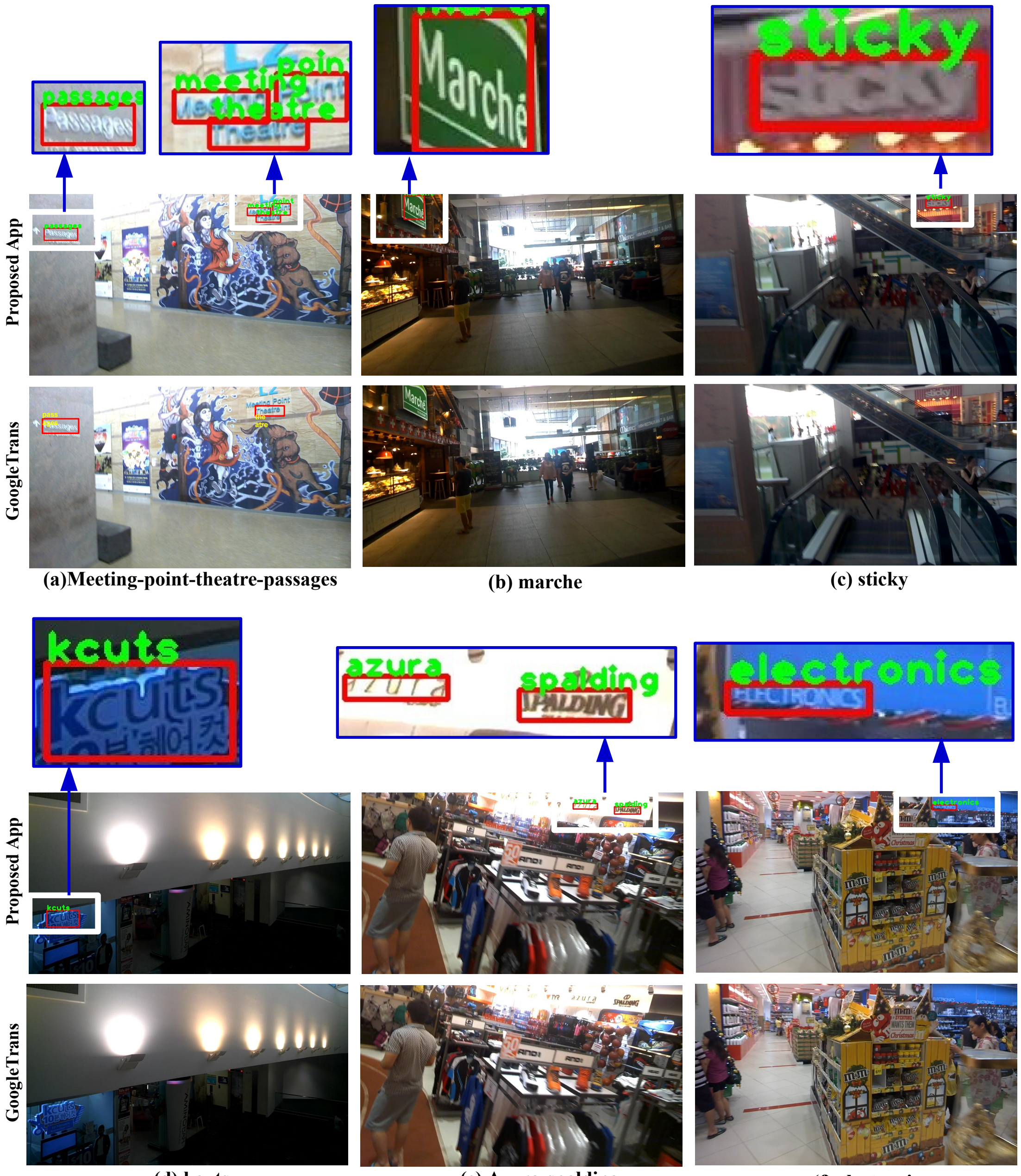
lyses scene text proposals sequentially. In addition, computing system is also a reason where the google scene text reading engine executes on a high computing system while the proposed application launches on a personal desktop computer.

Three figures as Figure 5.8, Figure 5.9 and Figure 5.10 present performance of two scene text searching applications on several example images in the evaluation dataset. Figure 5.8 shows images where the proposed application outperforms the GoogleTrans. In contrast, Figure 5.9 shows images where the GoogleTrans performs better than our proposed application. Figure 5.10 illustrates images where both applications fail to detect and recognize keywords. In these figures, words presented under images are keywords corresponding to images. As performance shown in Figure 5.8, the proposed application performs better than the GoogleTrans on texts degraded by illumination (image (a) and (e)), non-horizontal texts (image (b)), blur and decorated texts (image (c) and (d)). In contrast, the GoogleTrans is more reliable than the proposed application when scene texts are printed in standard fonts as images in Figure 5.9, even ultra-low contrast texts as texts in Figure 5.9(f). Note that two scene text searching applications are totally false when texts are printed using art fonts (Figure 5.10(c,e)), under affected by both illumination and perspective (Figure 5.10(b,d,f)). These weaknesses are still open challenges for future developments.

5.3 Conclusion

In this chapter, an automatic scene text reading system has been proposed. It has been developed based on our proposed scene text proposal techniques and the Jaderberg's scene text recognition model. Due to robustness of the two models, the developed scene text reading system can detect and recognize texts under different degradations. In comparison to state-of-the-art systems evaluated on two public scene text datasets, it provides competitive performances in both an end-to-end and a word-spotting contextualization evaluation.

A scene text searching application is also developed by using the proposed automatic scene text reading system as a back-end program. The web based interface has been designed aiming for help users interact conveniently with the back-end program. Due to using web based scheme, our application can be easily accessed by different devices such as smart phones, tablets, computers, in different operation systems (Window, Linux, Android, and iOS). It also integrates with a speech recognition engine developed by Google to provide a convenient data entry solution. In comparison with a released application as GoogleTrans, the proposed application provides better performances in term of searching for targeted keywords.



(a)Meeting-point-theatre-passages

(b) marche

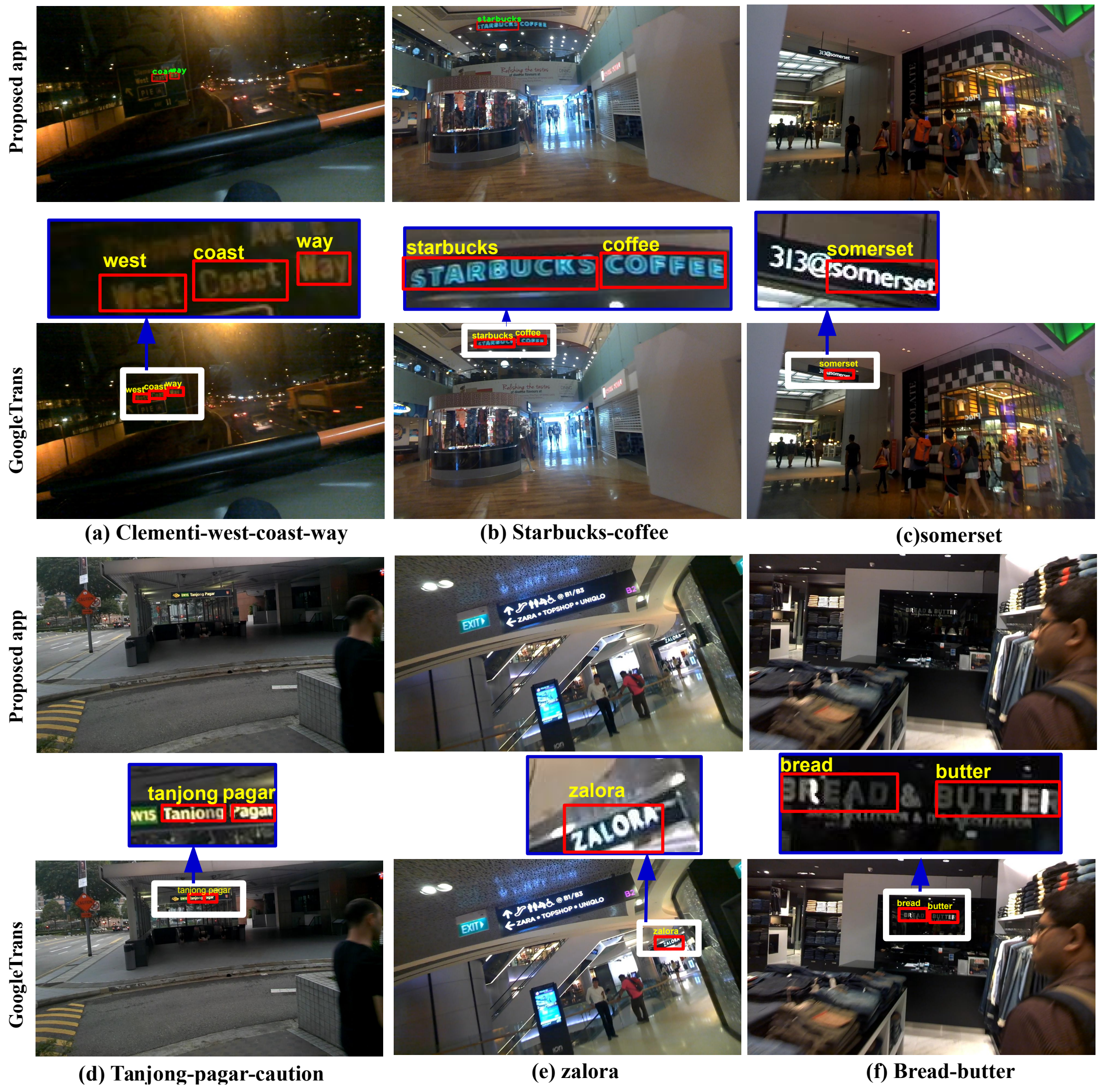
(c) sticky

(d) kcuts

(e) Azura-spalding

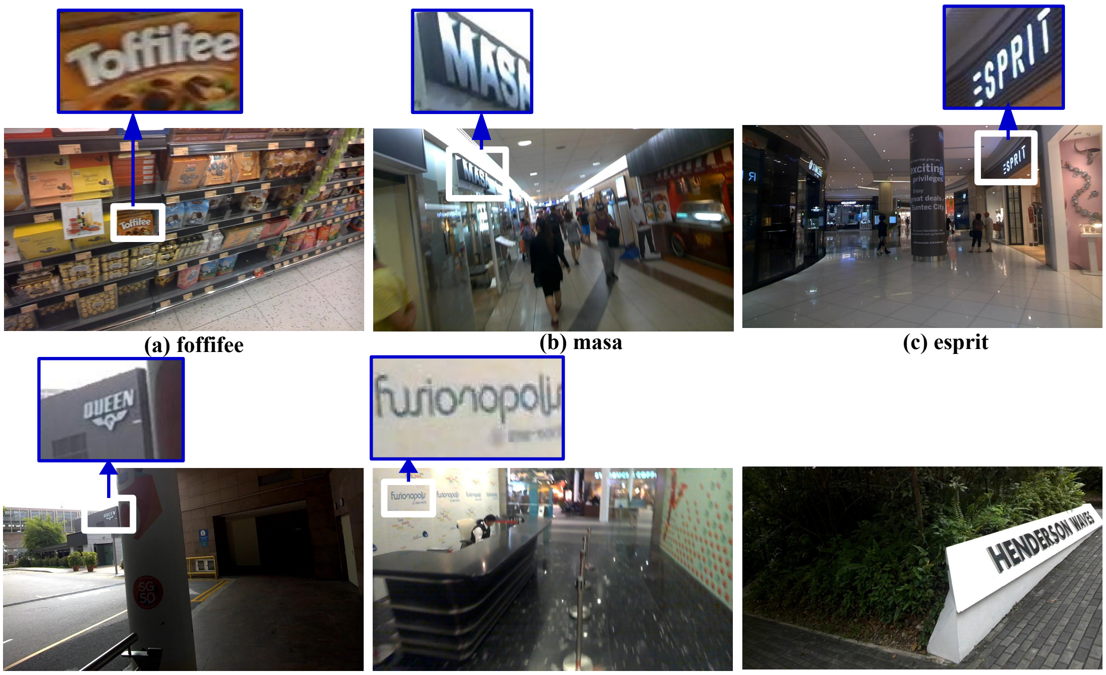
(f) electronics

Figure 5.8 – Several image examples on which the proposed scene text searching application outperforms the GoogleTrans app. Detected boxes are drawn on images, the green recognized words are from our proposed app and the yellow recognized words are from the GoogleTrans app. The white boxes describe from where images are cropped.



(a) Clementi-west-coast-way
(b) Starbucks-coffee
(c)somerset
(d) Tanjong-pagar-caution
(e) zalora
(f) Bread-butter

FIGURE 5.9 – Example images on which the GoogleTrans app provides better performance than the proposed app. The green recognized words are generated by the proposed app and the yellow recognized words are generated by the GoogleTrans. The white boxes describe from where images are cropped.



(a) foffifee (b) masa (c) esprit

(d) dueen (e) fusionopolis (f) Henderson waves

Figure 5.10 – Example images on which both proposed scene text searching app and GoogleTrans app fail to detect and recognize words. Those words are degraded uneven illumination, decoration purpose, and perspective. The white boxes describe from where images are cropped.

Chapter 6

Conclusion and future works

Contents

6.1 Conclusion

In this thesis, we have contributed several techniques and systems on the scene text detection and the scene text reading. Extensive evaluations on several scene text datasets show that the proposed works outperform recent state-of-the-art works.

In Chapter 3, the proposed scene text proposal technique has been described, which is adapted from the EdgeBox object proposal framework for scene text objects. Three major contributions are applied, including text specific edge-based low-cue features for discriminating text edges from non-text edges, a text geographical relationship based edge grouping solution for generating text proposals, and a text geographical relationship based scoring function for ranking proposals. The proposed scoring function is more simple than a Adaboost classifier and a deep learning classifier which are used in the simple selective search for scene text proposal and the DeepText technique, respectively. However, it performs better and sorts almost true text proposals in the top of the ranked list. This advantage is witnessed by superior recall rates of the proposed technique under a limited number of proposals at 2000.

In Chapter 4, the max-pooling based scene text proposal (MPT) has been proposed. A novel edge grouping solution and a proposal scoring function are developed. The proposed grouping solution is inspired by a max-pooling process in the deep learning framework, and it merges connected components searched on binary Canny edge maps into scene text proposals. Influences of pooling window size and stride values are studied. An unique feature of the max-pooling grouping is that it does not rely on any text specific heuristic rules for merging. A novel proposal scoring function is built up based on the Histogram of Oriented Gradient on edge pixels and the K-mean cluster algorithm. The MPT technique outperforms state-of-the-art scene text proposal techniques, including our developed technique as the Text-Edge-Box in both recall rates and efficiency.

In Chapter 5, we designed an automatic scene text reading system, which is intent on localizing and recognizing texts in scene images, and a scene text searching application. A state-of-the-art scene text recognition model is adopted for both scene text classification and scene text recognition, and it is built on top of our proposed scene text proposal techniques which are the TEB and the MPT. False positive proposals are eliminated based on a recognition confidence score threshold, lexicons, and non-maximum suppression. In the scene text reading competition, the proposed system provides state-of-the-art performance in two different evaluation contextualizations including an end-to-end scene text reading and a scene text spotting. Utilizing the developed automatic scene text reading system as a back-end program and adopting a server-client communication framework, the scene text searching application is developed. Users can access the application by using web browsers on their smart devices. Input images are captured by client cameras and the proposed application returns locations of provided keywords in output images if keywords are exist in images. The proposed application is compared with the Google Translation application in a contextualization of scene text searching and it provides better searching performance while its efficiency needs to be improved.

In the ensuing section, we will present future works which can be developed based on proposed techniques and systems. A YoLo based scene text proposal technique is proposed, which adopts the YoLo network to provide scene text proposals. Suggestions for improving performance of proposed techniques and systems are also presented, including a new system design and a quadrilateral bounding boxes generation. In addition, we proposed a relevant scene text searching based application for aiming elderly in their outdoor activities, including a navigation assistance, a danger alert, a scene text searching and a direction estimation.

6.2 Future works

6.2.1 Deep learning based scene text proposal

Deep learning framework (DL) has been applied on several computer vision applications such as object detection, object recognition, scene text detection, scene text recognition, and so on. These systems provided attractive performance in comparison with traditional frameworks developed base on hand-craft features. In recent years, this framework is also adopted for scene text proposals [21, 22, 30, 31]. They usually adapt the R-CNN framework for proposal generation by modifying the network to provide much more number of detection boxes at each cell in a square grid over image space. Proposal scores predicted by DL networks are seen as boxes confidence scores. An advantage of this solution is that it processes more efficient than traditional solutions. In this section, we propose a DL based scene text proposal technique which is constructed base on the YoLo9000 network architecture [89]. Note that the developed network in this section is just a baseline version and it needs further improvement as discussed in Section 6.2.1 to reach as well as outperform recent state-of-the-art scene text proposal techniques.

YoLo network

Yolo is you-only-look-one. As its name, the model is developed to localize and recognize objects with only one time scanning input images. An unique character of YoLo network among most recent deep learning frameworks such as R-CNN [79],

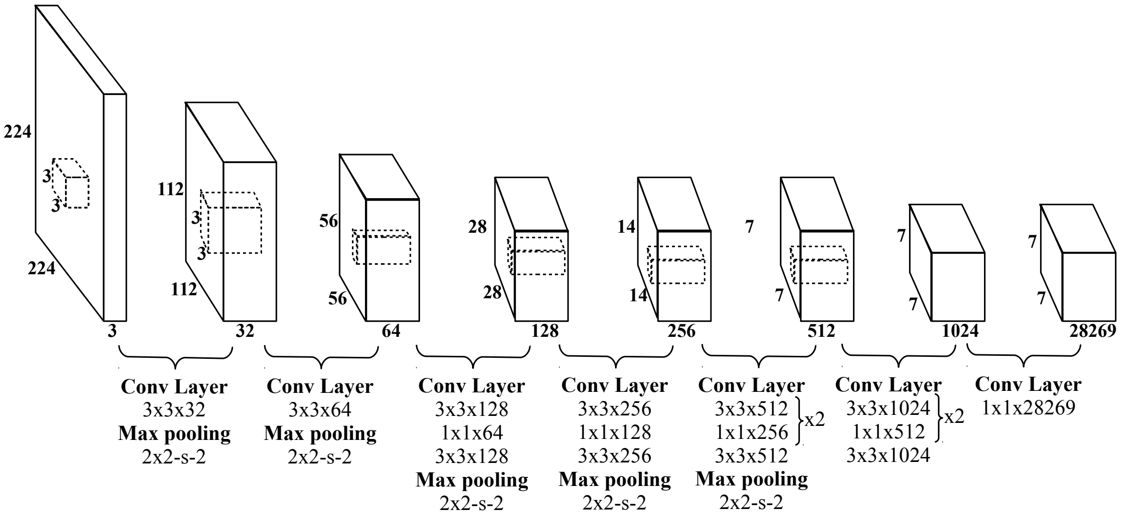


FIGURE 6.1 – YoLo framework consists of 19 convolution layers. Image space is divided into 49 cells by a square grid of 7×7. At each cell, three bounding boxes are predicted. Each box contains its coordinates (four parameters), its confidence score (one parameter), and probability distribution of 9418 objects (9418 parameters). It forms 28269 parameters as the third dimension of the output layer.

Fast and Faster R-CNN [80, 29] is that it performs coincidently both object localization and object recognition in the final prediction layers. Other frameworks mostly separate these two tasks into two different layers including proposal generation layers (forms a region proposal network (RPN)) and object recognition layers. This unique architecture brings YoLo several benefits including an extremely fast process, a global reasoning over image space for generating and scoring bounding boxes. Its architecture is shown in Figure 6.1

This architecture is inspired by the GoogleLeNet model [86] with a difference that inception modules are replaced by proposed blocks in which a 1×1 reduction layer is followed by a 3×3 convolution layer. The output layer has a size of $7 \times 7 \times 28269$, where the 7×7 space presents a square grid covering over image space. At each grid cell, the network predicts three bounding boxes. Each box contains five parameters as x,y,w,h,s, where [x,y,w,h] is box coordinates and s is a box confident score, and 9418-class probability distribution. They form the third dimension of the output layer as (3x(5+9418)=)28269 parameters.

The YoLo model has been trained on the join dataset including the COCO dataset [94] and the ImageNet dataset [95] to classify among 9000 classes. Multi-scale training is also implemented for improving the network robustness to object size variant. This flexibility is gained by converting last two fully connected layers in the older YoLo version [96] into two convolution layers. Due to combining the object localization task and the object detection task, a multi-part loss function is applied for training and it is shown in Equation 6.1.

$$\begin{aligned} L = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} ¶_{ij}^{obj} (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \\ + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} ¶_{ij}^{obj} (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \\ + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} ¶_{ij}^{obj} (C_i - \hat{C}_i)^2 \\ + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} ¶_{ij}^{noobj} (C_i - \hat{C}_i)^2 \\ + \sum_{i=0}^{S^2} ¶_{i}^{obj} \sum_{c \in Classes} (p_i(c) - \hat{p}_i(c))^2 \end{aligned} \tag{6.1}$$

Where $¶_{i}^{obj}$ denotes if object appears in cell i and $¶_{ij}^{obj}$ denotes that the jth bounding box predictor in the cell i is "responsible" for that prediction.

There are two weaknesses of the YoLo model which can be considered for future developments. Firstly, the YoLo model struggles in detecting small objects that stay nearby each other and inside a grid cell, such as folks of birds. It is because there is only one object detected in one grid cell. Secondly, a weakness comes from the training process where it treats equally errors in small bounding boxes and in large bounding boxes. In fact, small errors in large bounding boxes are generally benign while small errors in small bounding boxes are great effect on the Intersection over Union (IoU) loss.

YoLo based scene text proposal

In this proposed idea, the YoLo architecture is adopted to generate scene text proposals. Three issues are addressed including a training process, model modification, and multi-scale prediction. Training process is to adjust pre-trained YoLo weights from general objects sensitivity to scene text objects sensitivity. Model modification focuses on changing the output layer for providing a larger number of proposal boxes and being suitable for two classes classification as text/non-text instead of 9418 classes. Multi-scale prediction is implemented to support a model in handling text size variants. Network training process is implemented using the Darknet library developed by authors of YoLo network, published at the link : *https://pjreddie.com/darknet/*

Due to the output layer of YoLo network is designed to predict three bounding boxes at each cell, the original YoLo network provides maximally 147(=7x7x3) bounding boxes because some of generated boxes are eliminated by a confidence score threshold as 0.5 and a non maximal suppression with IoU of 0.4. In order to adapt this layer to generate scene text proposals, two modifications have been addressed. First, we change the number of classes from 9418 to 2 referring two classes as text and non-text. Second, a number of predicted boxes at each grid cell is increased from 3 to n and it is set at 10 in the baseline model implementation. Therefore, the third dimension of the output layer is ((5+2)*10 =)70 parameters. A confidence score threshold and a non maximum suppression IoU threshold are also changed, which are decreased from 0.5 to 0.1, and increased from 0.4 to 0.95 respectively.

In the training process, the developed model is first trained on a join dataset consisting of training images in the ICDAR2013 dataset [2] and training images in the Synthetic data [82]. Then, it is fine tuned again on the ICDAR2013 training images. In order to enrich training datasets, which is very useful for training a deep learning network, training images are also modified by adding noise, random cropping, flipping, and rotating before forwarding to the network. Resolution of input images is also varied to improve model's robustness to different text sizes. However, the input image size has to be a multiple of 32. Training hyper-parameters are adopted from the YoLo training configuration, including batch size of 1, momentum of 0.9, weight decay of 0.0005, initial learning rate of 0.01, learning rate decay of 0.1 after constant epochs, maximum number of epochs of 10^8.

In evaluation, we also apply a multi-scale input image solution to get a larger

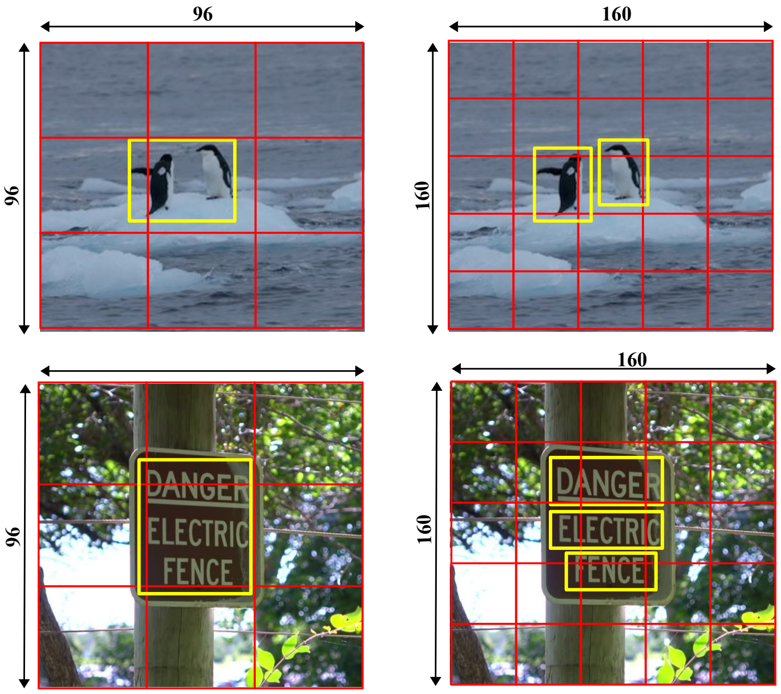


FIGURE 6.2 – Two example images with different scales and according grid sizes. With a size of 96 × 96, two images in the left will be downscaled into a grid size of 3 × 3 after going through the YoLo network. Similarly, two images in the right are downscaled to a grid size of 5 × 5. Square grids are presented with red lines. For a grid size of 3 × 3, objects in two images are located inside a grid and can not be detected separately. In contrast, a grid size of 5 × 5 can split them into single detection.

chance of locating texts in scene images even though the YoLo model is already designed to handle multi-scale object detection in single scale input images with state-of-the-art performances. By changing input image resolutions, we implicitly control a grid size. In fact, an image is downscaled 32 times when it goes through the network. A grid size therefore is a result of image resolution divided by 32. For example, if an input image has a resolution of 32 × 32 or 64 × 64, a grid size will be 1 × 1 or 2 × 2, respectively. In consequence, this solution addresses the weakness of the original YoLo model in detecting small objects when they are nearby and inside a grid cell. Figure 6.2 shows an example where a multi-scale input image can address a YoLo's problem. The first row presents a problem in general object detection and the second row presents a problem in scene text object detection. As images shown in Figure 6.2, small objects like birds (in the first row image) or small texts (in the second row image) can be false to detect with small grid size (left images in the two rows) due to the YoLo weakness. By increasing input image resolution, cell grid becomes more dense and it can separate small objects into single cells (right images in the two rows).

Evaluation

The baseline of YoLo based scene text proposal is evaluated on two public scene text datasets as the ICDAR2013 dataset [2] and Street View Text (SVT) dataset [3]. Baseline system performances are compared with state-of-the-art scene text proposal techniques in both the scene text proposal evaluation and the scene text reading evaluation. In these evaluations, two versions of the baseline system are introduced including one is the original YoLo set-up, which provides three predicted bounding boxes at each grid cell and sets parameters of detection threshold and nms threshold at 0.5 and 0.4 respectively, and one is the modified YoLo set-up for scene text proposal. In ensuing paragraphs, the original YoLo set-up version is named as O_YoloModel, and the modified YoLo set-up version is named as M_YoloModel.

Figure 6.3 presents performances of two versions of Yolo based scene text proposals and three state-of-the-art techniques including the Text-Edge-Box (TEB), the Maxpooling base Text Proposal (MPT) and the DeepText (DT) [21], using the scene text proposal evaluation framework which considers a system recall rate under different number of proposals and different IoU thresholds. As results shown in Figure 6.3, these YoLo based techniques outperform state-of-the-art techniques when a number of proposals and an IoU threshold are small. In the left column, performances of YoLo based techniques improve dramatically when the number of proposals increases within a range of (0,300) at IoU of 0.5 and 0.7. When IoU is 0.8, state-of-the-art techniques, which are TEB and MPT, provide superior performances in a full range of a number of proposals. In the right column, recall rates are evaluated under difference of IoU at three numbers of proposals as 100, 500, 1000. YoLo based techniques usually provide the best performance when IoU thresholds

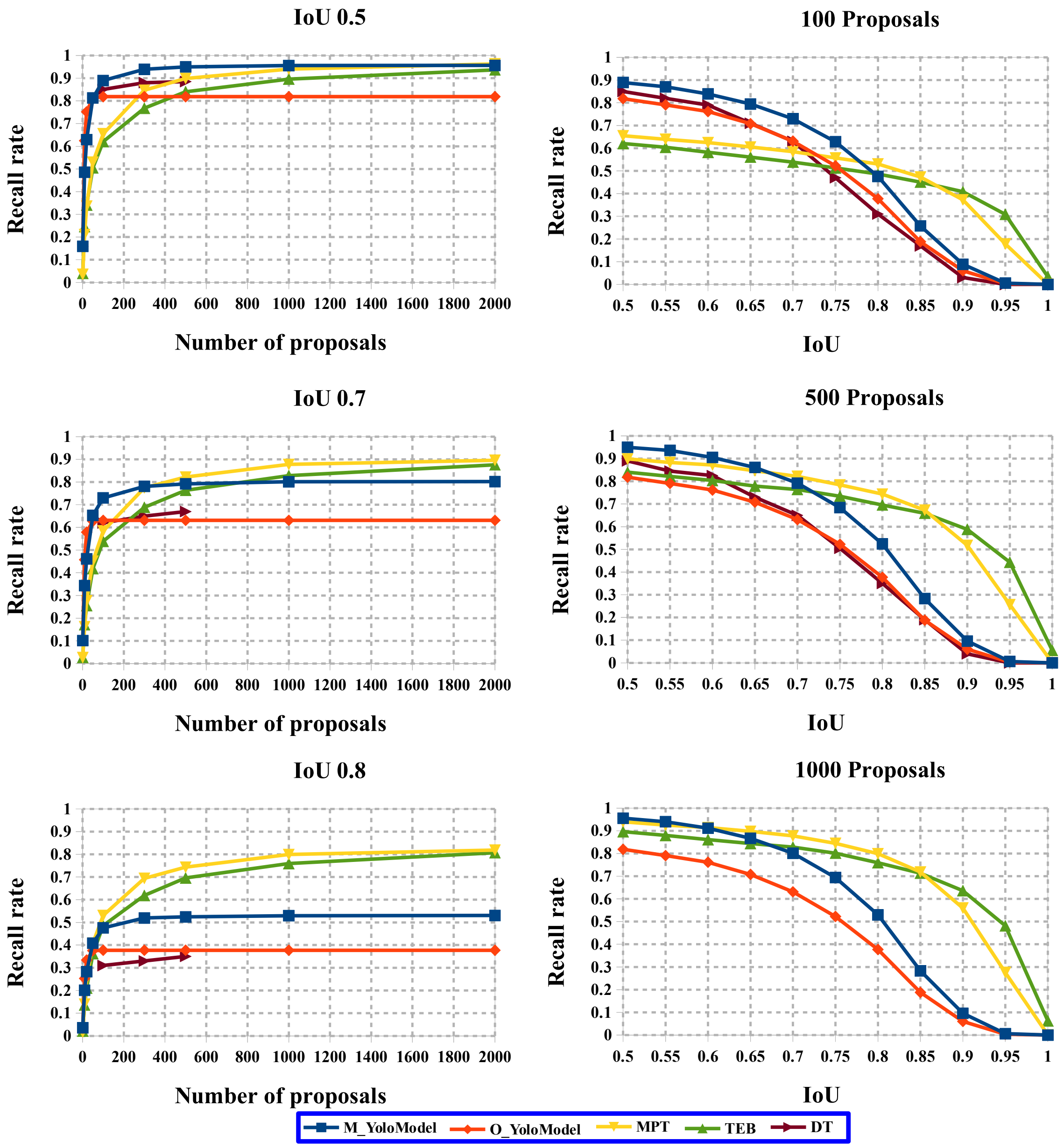


FIGURE 6.3 – Comparison of the baseline of YoLo based scene text proposal techniques (M_YoLoModel, and O_YoLoModel) to other state-of-the-art techniques on the ICDAR2013 dataset [2], in difference number of proposals and IoU thresholds. Comparable techniques include the Maxpooling based scene text proposal (MPT), the Text-Edge-Box (TEB), and the DeepText (DT) [21]

are small, less than 0.75, 0.65, and 0.55 for 100, 500, 1000 proposals, respectively. It also performs better than DeepText, which is developed based on the R-CNN neuron network. When the IoU threshold increases higher than 0.75, performance of deep

Table 6.1 – Recall (%) and processing time (in second) of two versions of YoLo based technique and other state-of-the-art techniques under different IoU thresholds on the ICDAR2013 dataset. The Nppb denotes an average number of proposals that the techniques need to achieve its presented recalls.

IoU	0.5	0.7	0.8	Nppb	times (s)
M_YoloModel	95.62	80.18	53.06	1171	1.49
O_YoloModel	81.83	63.11	37.72	**16**	**0.34**
MPT	**96.16**	**88.77**	**82.1**	1450	4.49
TEB [25]	94.25	87.95	81.55	1777	5.4
DT [21]	88.5	67	4	500	–

learning based techniques including YoLo based systems and DT are degrade dramatically. Between two YoLo based techniques, the modified one (M_YoloModel) usually provides approximately 10% better than the original one (O_YoloModel) in different combinations of a number of proposals and an IoU threshold. This improvement is because the modified model provides a larger number of proposals which widen a change of localizing text regions in images.

The second evaluation, two YoLo based baseline techniques are compared with state-of-the-art techniques in term of processing time and an average number of proposals each technique needs to provide competitive performances. This evaluation is shown in two Tables 6.1 and 6.2. As the shown results, YoLo based techniques provide competitive performance on the ICDAR2013 dataset and outperform other state-of-the-art techniques on the SVT dataset. Even though the O_YoloModel needs much less number of proposals than DT (16 comparing to 500), it performs much better than DT under IoU of 0.8 (37.72% comparing to 4%). Both YoLo based techniques are the best efficient techniques when they need less than two seconds in average to process an image, especially around 0.3 seconds by the O_YoloModel. Due to resizing images into constraint sizes, average processing time of both YoLo based techniques are similar on both datasets.

On the SVT dataset, YoLo based scene text proposal techniques provide attractive results in a full range of IoU from 0.5 to 0.8. Note that we do not evaluate

TABLE 6.2 – Recall (%) and processing time (in second) of two versions of YoLo based technique and other state-of-the-art techniques under different IoU thresholds on the SVT dataset. The Nppb denotes an average number of proposals that the techniques need to achieve its presented recalls.

IoU	0.5	0.7	0.8	Nppb	times (s)
M_YoloModel	**97.37**	**82.53**	**53.01**	1369	1.58
O_YoloModel	78.67	56.11	32.46	**18**	**0.34**
MPT	88.41	46.99	21.48	1769	3.59
TEB [25]	87.64	47.91	20.09	1890	2.2

techniques performance at the higher IoU threshold due to uncommonly used in real applications. The main reason is because proposal bounding boxes of YoLo based techniques are not close to text edges which are usually considered by human during labelling ground truth boxes. Yolo proposal boxes are therefore fine match with ground truth boxes in the SVT dataset where human labels are not strongly following text edges, and it fails to outperform state-of-the-art techniques in the ICDAR2013 dataset where human labels are required to be as close as possible to text edge pixels. YoLo proposal boxes are also more precise than DeepText proposal boxes because YoLo proposal boxes are estimated based on entire image contents which combines both content features inside a studied cell and content features in other neighbour cells. In contrast, DeepText boxes are predicted base on content features inside a studied cell only.

The effects of YoLo based scene text proposals to scene text reading system performance are also evaluated by embedding them into our propose scene text reading framework. Proposal boxes are then forwarded to the scene text recognition model for recognition and classification. False positive proposals are eliminated by a recognition score threshold, lexicons, and non maximum suppression. These systems performances are then compared with other developed systems, including TEB_based system and MPT_based system which achieve state-of-the-art performances in the robust reading competition in 2017. These results are shown in two Tables 6.3 and 6.4. Even though YoLo based scene text proposal techniques process

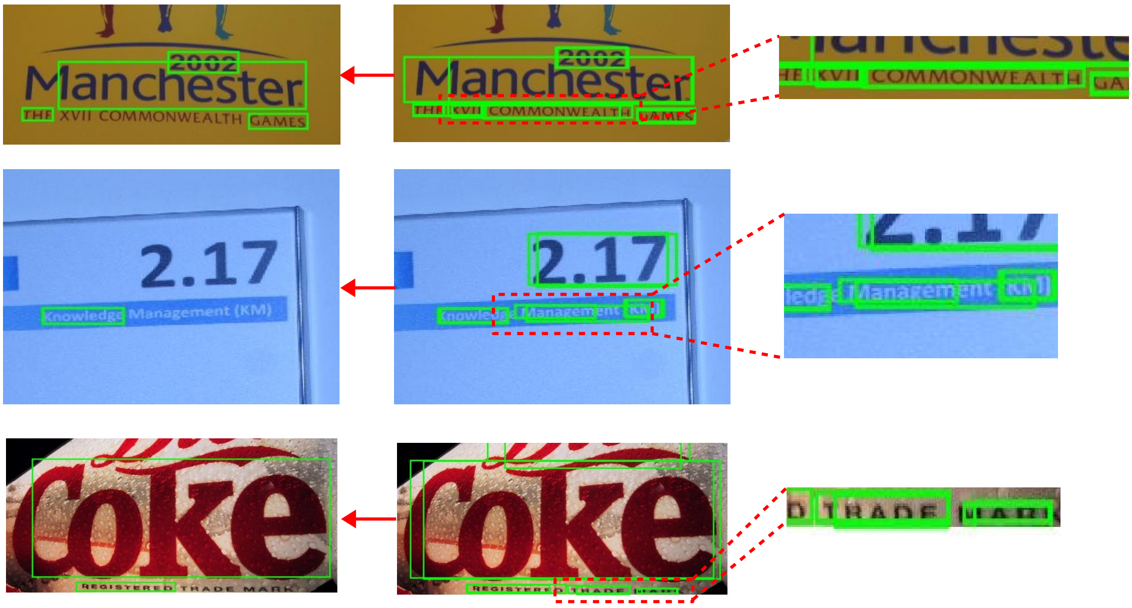
Table 6.3 – Word spotting performance of the YoLo based scene text reading systems (M_YoloModel_sys and O_YoloModel_sys) and other proposed systems (MPT_sys and TEB_sys) on the Robust Reading Competition 2013 Dataset (ICDAR2013) in the two contextualizations.

	Strong contextualization			Weak contextualization		
	Recall	Precision	F-score	Recall	Precision	F-score
M_YoloModel_sys	80.84	84.18	82.47	82.12	88.76	85.31
O_YoloModel_sys	79.79	97	87.34	79.43	**97**	87.34
TEB_sys	84.46	96.4	90.03	82.94	91.96	87.22
MPT_sys	**88.20**	**97.42**	**92.58**	**87.85**	95.31	**91.43**

Table 6.4 – End-to-end performance of the YoLo based scene text reading systems (M_YoloModel_sys and O_YoloModel_sys) and other proposed systems (MPT_sys and TEB_sys) on the Robust Reading Competition 2013 Dataset (ICDAR2013) in the two contextualizations.

	Strong contextualization			Weak contextualization		
	Recall	Precision	F-score	Recall	Precision	F-score
M_YoloModel_sys	78.95	87.65	83.07	77.86	83.02	80.36
O_YoloModel_sys	76.33	**97.35**	85.57	76.11	**96.11**	84.91
TEB_sys	80.8	94.51	87.12	79.49	90	84.42
MPT_sys	**84.08**	96.13	**89.70**	**83.86**	93.89	**88.59**

faster and provide smaller number of proposals, automatic scene text reading systems built on them still perform competitive results. Besides missing in detecting scene text objects, non-close-to-text-edge bounding boxes is also a reason that makes YoLo based scene text reading system miss detected texts due to false recognitions. Figure 6.4 presents examples of images in which detected texts have been removed by the text recognition model.



Output of scene text reading system Output of YoLo based scene text proposal

FIGURE 6.4 – Example images where texts are detected by YoLo_sys techniques and eliminated by the recognition model.

Suggestions for improving scene text proposal's performance

According to performances shown in the left column of Figure 6.3 and two Tables 6.1 and 6.2, deep learning based scene text proposal techniques can provide the best recall rate when a small number of proposals is required and they are the most efficient techniques. However, there is a disadvantage that their performances drop dramatically when Intersection over Union (IoU) threshold increases higher than 0.7 observed by the results shown in the right column of Figure 6.3. Therefore, in future works of improving performances of YoLo based scene text proposal techniques while system's efficiency is still inherited, we are going to integrate it with our proposed state-of-the-art technique as the Max-pooling based scene text proposal (MPT).

The first suggestion is to use a YoLo network to boost up performance of MPT technique in term of a small number of proposals such as 100, 200 by adapting the YoLo net for scoring proposal boxes. A solution is to use a YoLo network to generate a text heat-map where text pixels are assigned higher values than other non-text pixels. Generated MPT proposals are then scored based on the heat-map using simple equations such as mean of pixel values inside proposal boxes or a fraction of

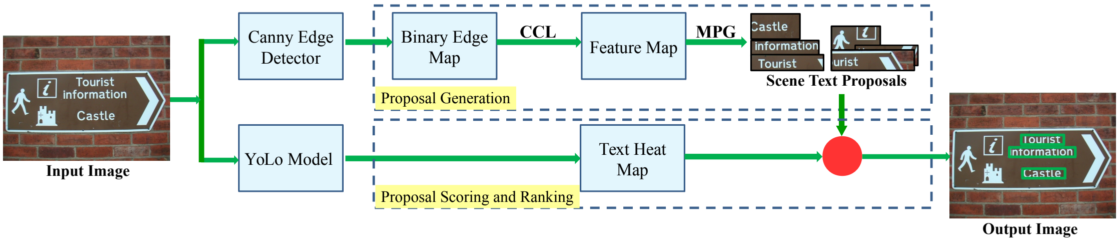


FIGURE 6.5 – The proposed framework for future development of maxpooling based scene text proposal by adopting a YoLo model for scoring proposals. **CCL** is connected component labelling and **MPG** is maxpooling based grouping.

text pixels in area covered by scene text proposal boxes, and so on. The framework of this suggestion is shown in Figure 6.5. By applying this solution, we expect to improve the proposed MPT technique in two aspects including processing time and technique performance under a limited number of proposals. For the processing time, replacing the current scoring function by a YoLo model, we can reduce a processing time of scoring-ranking proposals. In fact, the developed scoring function in the MPT technique consists a lot of exponential and square root operations which are parts of the Euclidean distance calculation utilized in Equation 4.1. These calculations also have to be repeated along with a number of proposals, and it could consume a lot processing time. When we apply the MPT technique on an 2592x3880 image, it needs 15.537 seconds for generating proposals and 49.78 seconds for scoring and ranking proposals. For processing the same size image, YoLo network, instead, needs only 0.943 seconds when it processes on a GPU 1080Ti. For technique performance under a limited number of proposals, replacing the current developed scoring function by a YoLo based text heat-map, we expect to have a more precise scoring solution, proposals are therefore ranked better and more text boxes are shifted to the front of ranked list, while a slope of MPT performance under different IoU thresholds is still maintained.

The second suggestion is to improve YoLo based proposal technique performance by utilizing the MPT scene text proposals as anchor boxes. This idea is inspired by the weakness of the recent anchor box initialization which contains hand-craft features and causes of missing objects in object detection systems. Particularly, recent

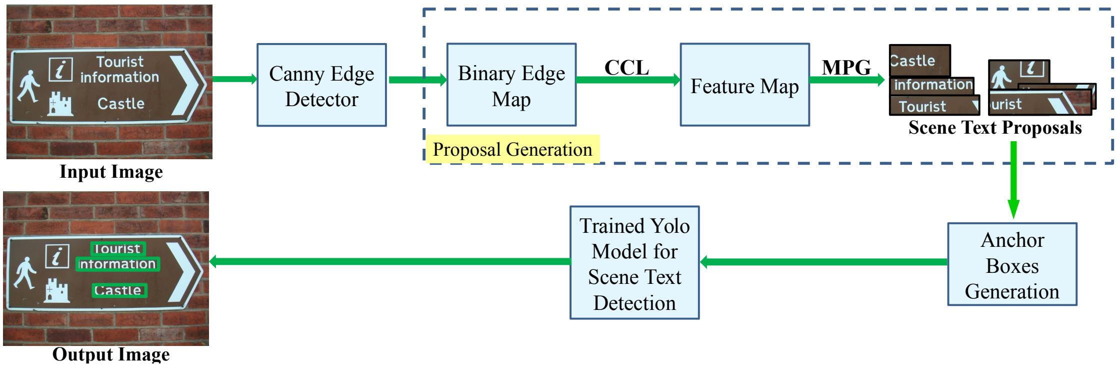


FIGURE 6.6 – The proposed framework for future development of YoLo based network by adopting the MPT scene text proposal technique as an anchor box generation. **CCL** is connected component labelling and **MPG** is maxpooling based grouping.

R-CNN neural networks developed for general-object/scene-text-object detection [80, 29, 96, 51, 21, 22, 30, 50] divide an image into grid cells and assume that there is only one object in a cell if it exists. At each grid cell, anchor boxes are initialized with different sizes and aspect ratios. R-CNN neural networks are trained to score those boxes and regress their ratio and size to cover objects better. In this framework, initial anchor boxes need to have a large range of sizes and aspect ratios to be robust to objects' shapes variants. Otherwise, they will be false. For example, the SSD network [51], which is one of state-of-the-art models in detecting general objects, is false when its original design is applied for detecting scene text objects. The reason is because scene text objects including words, text lines have straight aspect ratios which are not initialized in SSD's anchor boxes. By redesigning anchor boxes with more suitable aspect ratios, the SSD network performs better and achieves state-of-the-art performances in scene text detection [50]. By applying MPT as an anchor box initialization, anchor box size and aspect ratio are flexible and they can be self-adaptive to different image contents. Figure 6.6 illustrates this proposed framework, which uses the MPT technique as an anchor box initialization and the YoLo network as a regression and detection engine for refining and evaluating those initialized boxes.

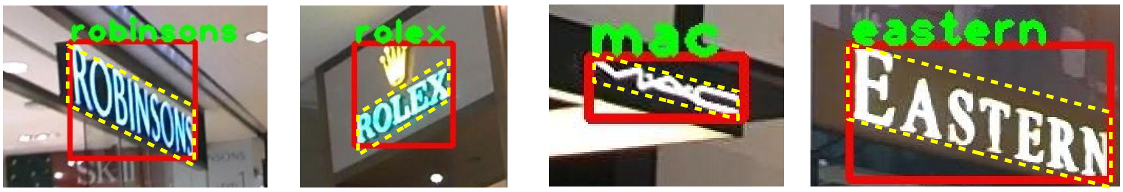
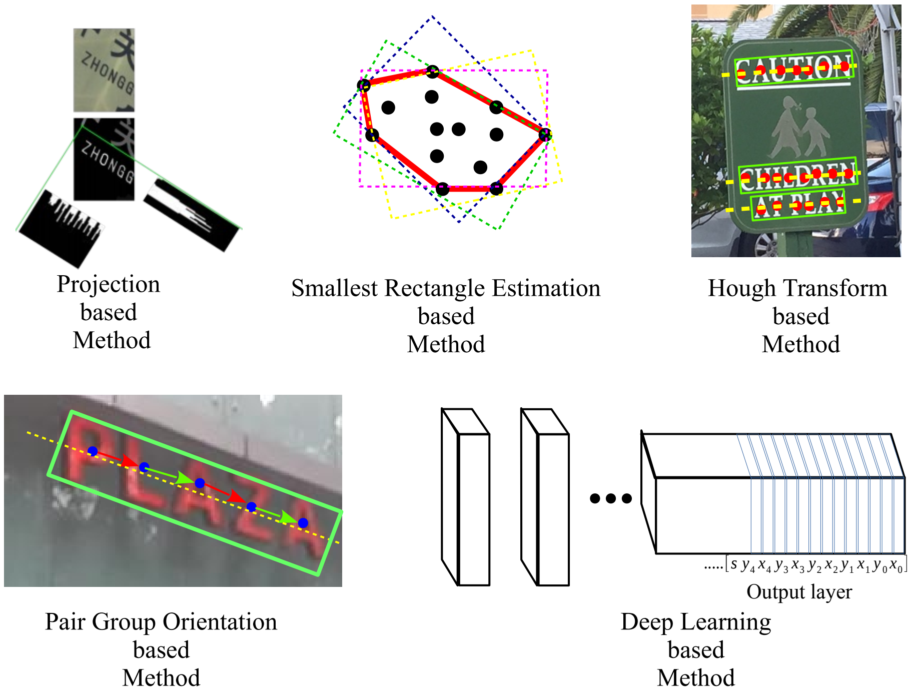


FIGURE 6.7 – Some examples of non-horizontal texts which have been detected successfully by the MPT scene text proposal technique while decided as miss detection in the IoU based evaluation. They are cropped out from images in the incidental scene text dataset. The red boxes are boxes generated by the MPT technique and the dash yellow boxes are ground truth bounding boxes.

6.2.2 Quadrilateral bounding boxes generation for orientated scene text detection

Even though the proposed scene text proposal techniques can detect non-horizontal texts or even curved text as discussion in Section 4.1.1, its detections are considered as false detections with non-horizontal texts due to a low IoU measurement with provided ground truth boxes, as some example images shown in Figure 6.7. A solution for addressing this problem is to provide quadrilateral bounding boxes for groups of connected components generated by our proposed scene text proposal techniques. There are many quadrilateral bounding boxes generation methods proposed, including hand-craft solutions : a multi-orientation projection [47, 39], a smallest rectangle estimation [40], a Hough transform based [97], an internal pair group orientations based [36, 45] as well as a deep learning based [46, 98, 99, 54, 52]. These methods are illustrated in Figure 6.8.

Three solutions including a multi-orientation projection, a Hough transform based and an internal pair group orientations based, quadrilateral bounding boxes are generated based on text line orientation. The multi-orientation projection solution searches for text line orientations by projecting a group of points into different orientations from -90 to 85 degree with an interval of 5 degree. Text lines should be elongated and its projection profile should have the highest variance. In the Hough transform based solution, orientated text lines are estimated by the Hough trans-


ZHONGG
CAUTION
CHILDREN
AT PLAY
PLAZA
.....[$s\ y_4\ x_4\ y_3\ x_3\ y_2\ x_2\ y_1\ x_1\ y_0\ x_0$]
Output layer

Projection based Method

Smallest Rectangle Estimation based Method

Hough Transform based Method

Pair Group Orientation based Method

Deep Learning based Method

FIGURE 6.8 – Several methods have been studied and applied to generate quadrilateral bounding boxes for better enclosing to oriented scene texts

form applied on centroid points of characters inside a studying group. In the internal pair group orientation based estimation, text line orientation is estimated from the most orientations of internal links of pairs of characters inside a group. On the other hand, the polygon method [100] is applied to find the smallest rectangle bounding box of a group of points [40]. In the MPT technique, proposal boxes contain connected components which have not been defined as text/non-text yet. Solutions which consider characters as key points for estimating text line orientation therefore could not be suitable to apply. Instead, if we consider connected components as groups of pixel points, proposal boxes therefore are considered as groups of points. Methods such as the multi-orientation projection and the smallest rectangle estimation are suitable options to upgrade the MPT technique.

Adapting deep learning networks for predicting quadrilateral bounding boxes is proposed in recent years with a lot of solutions. Note that deep learning based region proposal methods usually divide image space into a grid. In [98], deep learning model is designed and trained to predict both four 2D coordinators of quadrilateral boxes (8 parameters) and geometry values of rotated horizontal bounding boxes (top-left coordinators $[x, y]$, height, height, and angle) for each cell of an image. In [46, 54], text regions and orientated links between text region bounding boxes are predicted based on which words and text lines are formed by merging text regions together. Quadrilateral bounding boxes are then estimated on groups of text region bounding boxes as the system post-processes. In [99], two deep learning networks are deployed following a cause-to-fine strategy where a cause network provides a text attention map and a fine network processes on cropped text regions to generate a text line heat-map and a text line area heat-map. Two maps are then binarized by threshold of 0.5 and quadrilateral bounding boxes are estimated base on detected regions using the minimum area rectangle estimation method [100]. In the other hand, quadrilateral text regions are searched by using quadrilateral anchor boxes and a regression deep learning network [52]. Those strategies can be applied on the developed YoLo based scene text proposal system which is discussed in Section 6.2.1. The proposed solution consists of a MPT technique based anchor initialization and a modification of the YoLo output layer for estimating quadrilateral coordinators. The model can be trained for regressing MPT proposal boxes into quadrilateral boxes.

6.2.3 A navigation application for aiming elderly in their outdoor activities

The world population is ageing due to life quality improvement with a lot of caring services in healthy and safety. In 2017, there is an estimation of 962 million people aged 60 or above in the world, comprising 13 percent of the global population, and that age population is growing at a rate of about 3 percent per year. By 2050 all regions of the world except Africa are predicted to have nearly a quarter or more of their populations at ages 60 and above. According to age-in-place project [101], nearly 90% of people over age 65 want to stay at their home as long as possible. They also prefer to show up that they have abilities to stay safe and independent. However, staying alone for elderly, especially people with ageing diseases such as parkinson and dementia is unsafe due to a lot of risks, such as fail in walking, eating, using kitchen tools, and so on. Allowing their family members and caregivers to keep taking care of their activities while giving them a feel of independence will be a hard deal but it is an interesting and high demand mission. In fact, many proposed systems have been investigated, and most of applications are developed for assisting elderly in indoor activities which are known as nursing house projects including fall detection systems [102], tasks reminder applications, and so on[103].

In our observation, outdoor activities are also very helpful for the elderly as they aim to reduce elderly cognitive weakness by increasing interactions with social activities and communities. However in an outdoor environment, elderly have to deal with much more dangerous problems such as getting loss, falling in the street, being subject to crimes, being hit by vehicles, and so on. To avoid these incidents, the market demand is high. However, the number of recent applications supporting elderly in their outdoor safety is small. There is only the Global Position System (GPS) based elderly tracker which is deployed to support caregivers and family members in localizing elderly position and tracking their activities base on their movements in maps [104]. In this section, we are going to propose an application scenario based on our developed scene text searching system and the GPS system to assist elderly in their daily outdoor activities.

Proposed application framework

The GPS based navigation system is currently adopted for many tracking systems and the most favourite application is the google map which helps people to track their location and search directions. The elderly tracker system is also developed base on this google map API and supports in tracking elderly locations, arranging safety zone, elderly moving speed. On the other hand, this system does not provide extensive information regarding environment around elderly, which could be very useful for recognizing their situations and providing supporting suggestions. In this future work proposal, we are intent on integrating the GPS tracking module with our developed scene text searching system to enrich capability of our system. Besides an elderly tracking function, the system can support the elderly in finding their direction, navigating their way to their target destinations, searching for specific words, alerting dangerous conditions, and so on. Figure 6.9 depicts an infrastructure of the proposal system, including client wearable devices, which consists of a camera, microphone and a GPS module, and a server in which our back-end program and a database of local areas where elderly is living are executed and stored, respectively.

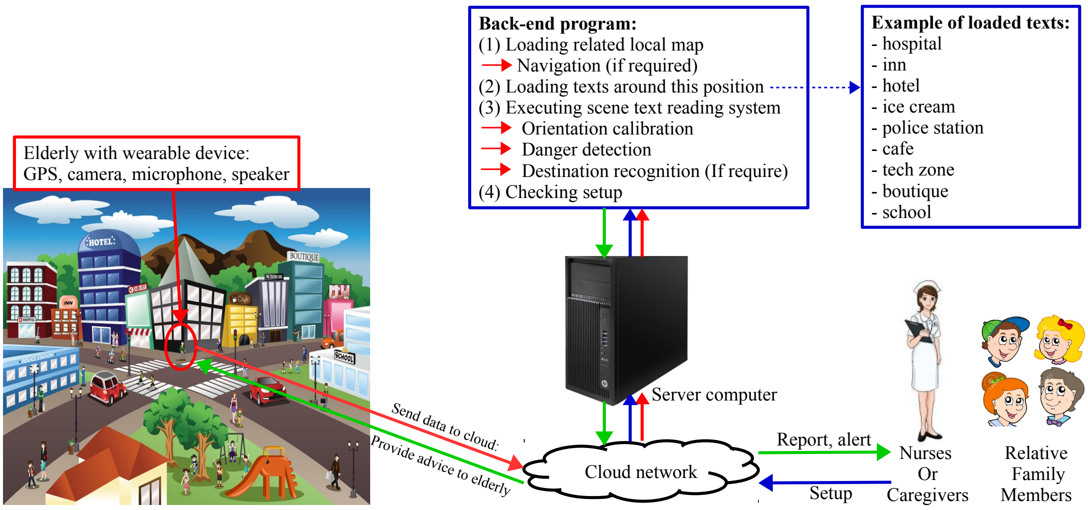


FIGURE 6.9 – A proposed application framework aiming to help elderly in their outdoor activities, including a direction estimation, a specific address searching, a danger detection, a navigation, and so on. The system also connects to related people of elderly such as their nurses, caregivers, family members

In order to provide a navigation function, the system needs to know a current location and a target location. The current location can be estimated from the GPS device, and the target location is provided by users using their microphone or keyboard. After receiving required information, additional information related to two locations is queried from the system database such as a local map, local scene texts around two locations. The shortest route is defined in the local map and local scene texts are used for correcting users' direction. Note that directions estimated from the GPS device are also collected. For example, in Figure 6.10 when users' locations are figured out, their local texts are loaded and used as keywords. When one of texts in the keyword list appears in images captured by wearable cameras, user's direction is calibrated so that the system can provide more precise advices. In a case that target destination contains texts, these texts are also used to guide elderly reach their destinations.

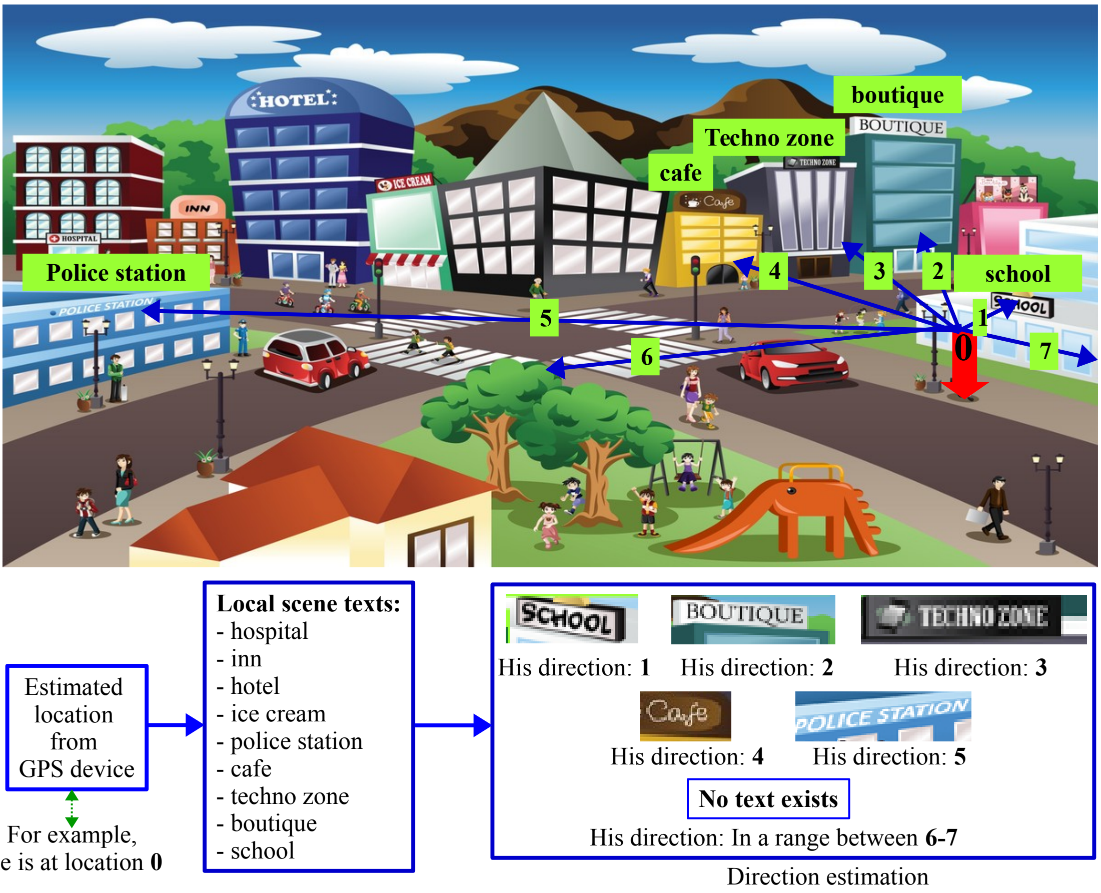


Figure 6.10 – An example solution for observing users direction using scene text objects which can appear in captured images taken by users wearable cameras.

The proposed system also employs a danger alert function. Specific keywords which are usually used to present dangerous situations in modern cities such as "construction in process", "electric shock", "dangerous", "caution", "work ahead", and so on are added to the system keyword list. When those words appear in captured images, a danger alert function is activated and it could notice to users as well as according people to let them take a look on users. Depending on where those words are in images, and word sizes, different levels of alert could be generated. For example, if those words have a large size and at the centre of images, users might stay very close to dangerous areas. A high lever alert should be activated.

Appendix

In this appendix, we are intent on showing up our implemented scene text proposal generator classes for the TextEdgeBoxes technique and the Maxpooling based Text Proposal Technique and related codes to utilize this class.

TextEdgeBoxes class :

The parameter **_alpha** is a parameter of the Text Edge Map generation function, which is used to control participation of the two edge based text specific features (EP and EV) in scoring connected components.

The parameter **_arexp** is used in a searching space generation. The 2×_arexp is a ratio of a searching space height to a bounding box height of the study connected component.

The parameters **_wthr, _oathr, _hthr** are parameters of the proposed grouping solution of the Text Edge Box technique.

C++ codes

```
class TextEdgeBoxGenerator{
  public :
  //method parameters (can be manually set)
  float _alpha, _wthr, _arexp, _oathr, _hthr ;
  //A main function of the TEB class. It takes a binary edge map (BW), a gra
  //-dient map (E), and an oriented gradient map (O) as inputs and provides a list
  //of proposal bounding boxes (boxes)
  void generate( Boxes &boxes, arrayf &E, arrayf &O, arrayi &BW ) ;
  private :
  //Edge segment information (utilized in clusterEdges)
  int h, w ;//Input image dimensions
```

```
    int __segCnt;//A number of connected components
    arrayi __segIds;//Segmented edge pixels ids (-1/0 means no segment)
    arrayf __EPsegIds;//A EdgePair score map
    arrayf __EVsegIds;//A EdgeVariance score map
    arrayf __TEMsegIds;//A TextEdgeMap
    vector<vectori> __csIds, __rsIds;//x,y coordinates of edge pixels
    //internal functions (our contributions) :
    //Searching for 8-neighbourhood connected components
    void CCfound(vectori &cs, vectori &rs, vectori ids);
    //Clustering edge pixels based on their score visualized by a TextEdgeMap
    void clusterEdges(arrayf &E, arrayf &O, arrayi &BW );
    //Generating proposals, estimating proposal bounding boxes and scoring proposals
    //it including many sub-functions
    void BoxesGenerator(Boxes &boxes);
    //Splitting proposals into word-level proposals
    void BoxesBreakDown(Boxes &boxesout,Boxes &boxesin);
    //Estimating boxes covering whole small boxes
    void boxlist2box(Box &b, Boxes &boxes);
    //Scoring proposal boxes in which our proposed scoring function is implemented
    void scoreBox( Box &box, arrayf &V );
    //Two edge based text specific features estimation functions
    void EPCC(float &ep,vectori &cs,vectori &rs,arrayf &O);
    void EVCC(float &ev,vectori &cs,vectori &rs);
};
```

Utilizing the TEB class :

C++ codes

```
//Initializing a TEB class
TextEdgeBoxGenerator TEBGen;
//Initializing vector to store generated proposal boxes. Each element as a vectori is
// a box
vector<vectori> boxes;
```

```
//Manually set class parameters
TEBGen._alpha = par0 ;
TEBGen._wthr = par1 ;
TEBGen._arexp = par2 ;
TEBGen._oathr = par3 ;
TEBGen._hthr = par4 ;
TEBGen._sc = par5 ;
//Text proposals generation. E, O, BW are a gradient map, an orientation map and
// a binary edge map respectively
 TEBGen.generate(boxes,E,O,BW) ;
```

MaxPoolingProposalText class :

C++ codes

```
class MPTGenerator{
   public :
   //"O" and "BW" are an oriented gradient map and a binary edge map
   //"sc" and "psinfo" are a scale of input image size to original size and a group of
   //pooling window size and stride values
   //"boxes" is the class output which are generated text proposal boxes
   void ProposalBoxes(Boxes &boxes, arrayf &O, arrayb &BW, arrayf &sc, arrayf
                        &psinfo) ;
   private :
   // Class parameters which are set in the public ProposalBoxes function
   int h, w ;
   int __segCnt ;// Total segment count
   float __scale ;
   arrayi __segIds ;// Segmented edge pixels map (-1/0 means no segment)
   arrayf __psinfo ;//Pooling window size and stride values
   vector<vectori> __csIds, __rsIds ;//x,y coordinates of edge pixels
   vector<vectorf> __hogIds ;//Histograms of gradient of connected components
   arrayf __cens ;//Pre-trained text/non-text centroids
```

```
    vector<vectori> __GList ;//A list of groups of connected components
    //Internal functions which are our contributions
    void clusterEdges(arrayf &O, arrayb &BW, float thrs ) ;
    void BoxesGenerator(Boxes &boxes, arrayf &sc) ;
    //Searching for 8-neighbourhood connected components
    void CCfound(vectori &cs, vectori &rs, vectori ids) ;
    void Grouping() ;//The proposed max-pooling based grouping solution is imple
    //mented
    void BoxesScore(float &s, vectori &idlist) ;//The proposed scoring function is
    //implemented
    //Proposal boxes generation functions
    void BoxesFromList(Boxes &boxes, vector<vectori> selectedList) ;
    void CleanList(vector<vectori> &cleanedList, vector<vectori> selectedList) ;
    void LinkList(vector<vectori> &linkedList, vector<vectori> selectedList) ;
    void List2Boxes(Boxes &boxes, vector<vectori> selectedList) ;
} ;
```

Utilizing a MPT class

```
//Loading text/non-text centroids
arrayf cens = <load _cens> ;
//Set pooling window size (w_pw and h_pw) and stride values (w_str and h_str)
vectorf psinfo ;
psinfo.push_back{w_pw} ;
psinfo.push_back{h_pw} ;
psinfo.push_back{w_str} ;
psinfo.push_back{h_str} ;
Boxes boxes ;
MPTGenerator MPTGen ;
MPTGen.ProposalBoxes(boxes, O, BW, cens, psinfo) ;
```

Bibliographie

[1] D. Karatzas, L. Gomez, A. Nicolaou, S. K. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida & E. Valveny ; «ICDAR 2015 Competition on Robust Reading» ; Proceedings of the 2015 13th International Conference on Document Analysis and Recognition p. – (2015).

[2] Karatzas, Dimosthenis, Shafait, Faisal, Uchida, Seiichi, Iwamura, Masakazu, Bigorda, L. Gomez, Mestre, S. Robles, Mas, Joan, Mota, D. Fernandez, Almazàn, J. Almazàn, de las Heras & L. Pere ; «ICDAR 2013 Robust Reading Competition» ; Proceedings of the 2013 12th International Conference on Document Analysis and Recognition p. 1484–1493 (2013).

[3] K. Wang, Belongie & Serge ; «Word Spotting in the Wild» ; Proceedings of the 11th European conference on Computer vision (ECCV) p. 591–604 (2010).

[4] B. Shi, C. Yao, M. Liao, M. Yang, P. Xu, L. Cui, S. Belongie, S. Lu & X. Bai ; «ICDAR2017 Competition on Reading Chinese Text in the Wild (RCTW-17)» ; arXiv preprint arXiv :1708.09585 (2017).

[5] Google ; «Google Translation Engine» ; https://play.google.com/store/apps/details?id=com.google.android.apps.translate&hl=en.

[6] Microsoft ; «Microsoft Translation Engine» ; https://www.microsoft.com/en-sg/store/p/translator/9wzdncrfj3pg.

[7] Orcam ; «Orcam my eye» ; https://www.orcam.com/en/.

[8] S. L. Joseph, X. Zhang & I. Dryanovski ; «Semantic Indoor Navigation with a Blind-User Oriented Augmented Reality» ; IEEE International Conference on Systems, Man, and Cybernetics (SMC) p. – (2013).

[9] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu & A. Y. Ng ; «Reading Digits in Natural Images with Unsupervised Feature Learning» ; NIPS workshop on deep learning and unsupervised feature learning p. – (2011).

[10] J. Greenhalgh & M. Mirmehdi ; «Recognizing Text-Based Traffic Signs» ; IEEE Transactions on Intelligent Transportation Systems **16**, p. 1–10 (2015).

[11] OpenALPR ; «License Plate Detection and Recognition» ; (Accessed on 05-Dec-2017). http://www.openalpr.com/on-premises.html.

[12] R. Smith ; «An Overview of the Tesseract OCR Engine» ; Ninth International Conference on Document Analysis and Recognition,ICDAR. (2007).

[13] Q. Ye & D. Doermann ; «Text detection and recogni- tion in imagery : A survey» ; Pattern Analysis and Machine Intelli- gence, IEEE Transactions on **37**, p. 1480–1500 (2015).

[14] K. Jung, K. I. Kim & A. K. Jain ; «Text information extraction in images and video : a survey» ; Pattern Recognition, Elsevier **37**, p. 977–997 (2004).

[15] J. Liang, D. Doermann & H. Li ; «Camera-based analysis of text and documents : a survey» ; International Journal on Document Analysis and Recognition (IJDAR) **2**, p. 84–104 (2005).

[16] S. Uchida ; «Text Localization and Recognition in Images and Video» ; Handbook of Document Image Processing and Recognition, springer p. 843–883 (2014).

[17] B. Computer Vision Center ; «Robust Reading Competition 2015» ; http://rrc.cvc.uab.es/.

[18] A. Veit, T. Matera, L. Neumann, J. Matas & S. Belongie ; «CoCo-Text : Dataset and Benchmark for Text Detection and Recognition in Natural Images» ; arXiv preprint arXiv :1601.07140 (2016).

[19] L. Gomez & D. Karatzas ; «TextProposals : a Text-Specific Selective Search Algorithm for Word Spotting in the Wild» ; Pattern Recognition **70**, p. 60–74 (2017).

[20] M. Jaderberg, K. Simonyan, A. Vedaldi & A. Zisserman ; «Reading Text in the Wild with Convolutional Neural Networks» ; International Journal of Computer Vision **16**, p. 1–20 (2016).

[21] Z. Zhong, L. Jin & S. Huang ; «DeepText : A new approach for text proposal generation and text detection in natural images» ; IEEE International Conference on Acoustics, Speech and Signal Processing p. 1208–1212 (2017).

[22] Z. Tian, W. Huang, T. He, P. He & Y. Qiao ; «Detecting Text in Natural Image with Connectionist Text Proposal Network» ; European Conference on Computer Vision (ECCV) p. – (2016).

[23] H. Jan, R. Benenson & B. Schiele ; «How good are detection proposals, really ?» ; Proceedings of the British Machine Vision Conference p. – (2014).

[24] Z. Zhang, W. Shen, C. Yao & X. Bai ; «Symmetry-Based Text Line Detection in Natural Scenes» ; IEEE Conference on Computer Vision and Pattern Recognition p. 2558–2567 (2015).

[25] D. Nguyen, S. Lu, N. Ouarti & M. Mokhtari ; «Text-Edge-Box : An Object Proposal Approach for Scene Text Localization» ; IEEE Winter Conference on Application of Computer Vision p. 1296–1305 (2017).

[26] L. Zitnick & P. Dollar ; «Edge Boxes : Locating Object Proposals from Edges» ; European Conference on Computer Vision p. 391–405 (2014).

[27] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers & A. W. M. Smeulders ; «Selective Search for Object Recognition» ; International Journal of Computer Vision **104**, p. 154–171 (2013).

[28] M. Jiri, C. Ondrej, U. Martin & P. Tomas ; «Robust Wide Baseline Stereo from Maximally Stable Extremal Regions» ; British Machine Vision Conference p. – (2002).

[29] S. Ren, K. He, R. Girshick & J. Sun ; «Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks» ; Advances in Neural Information Processing Systems 28 p. 91–99 (2015).

[30] Y. Jiang, X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu & Z. Luo ; «R2CNN : Rotational Region CNN for Orientation Robust Scene Text Detection» ; CoRR **abs/1706.09579**, p. – (2017).

[31] L. Rong, E. Mengyi, L. Jianqiang & Z. haibin ; «Weakly supervised text attention network for generating text proposals in natural scenes» ; International conference on document analysis and recognition (ICDAR) (2017).

[32] T. Shangxuan, Y. Pan, C. Huang, S. Lu, K. Yu & C. L. Tan ; «Text Flow : A Unified Text Detection System in Natural Scene Images» ; IEEE International Conference on Computer Vision p. 4651–4659 (2015).

[33] Z. Siyu & Z. Richard ; «A Text Detection System for Natural Scenes With Convolutional Feature Learning and Cascaded Classification» ; Conference on Computer Vision and Pattern Recognition (CVPR) p. 625–632 (2016).

[34] T. Wang, D. J. Wu, A. Coates & A. Y. Ng ; «End-to-end Text Recognition with Convolutional Neural Networks» ; Proceedings of the 2012 International Conference on Pattern Recognition (ICPR) p. 3304–3308 (2012).

[35] K. Wang, B. Babenko & S. Belongie ; «Word Spotting in the Wild» ; Proceedings of the 2011 International Conference on Computer Vision (ICCV) p. 1457–1464 (2011).

[36] X. Yin, W. Pei, J. Zhang & H. Hao ; «Multi-Orientation Scene Text Detection with Adaptive Clustering» ; IEEE Transactions on Pattern Analysis and Machine Intelligence **37**, p. 1930–1937 (2015).

[37] S. Qin & R. Manduchi ; «A Fast and Robust Text Spotter» ; IEEE International Conference on Applications of Computer Vision (WACV) p. 1–8 (2016).

[38] X. Yin, X. Yin, K. Huang & H. Hao ; «Robust Text Detection in Natural Scene Images» ; IEEE Transactions on Pattern Analysis and Machine Intelligence **36**, p. 970–983 (2014).

[39] L. Kang, Y. Li & D. Doermann ; «Orientation Robust Text Line Detection in Natural Images» ; IEEE Conference on Computer Vision and Pattern Recognition (CVPR) p. 4034–4041 (2014).

[40] H. Cho, M. Sung & B. Jun ; «Canny Text Detector : Fast and Robust Scene Text Localization Algorithm» ; IEEE Conference on Computer Vision and Pattern Recognition (CVPR) p. 3566–3573 (2016).

[41] M. C. Sung, B. Jun, H. Cho & D. Kim ; «Scene Text Detection with Robust Character Candidate Extraction Method» ; International Conference on Document Analysis and Recognition (ICDAR) p. 426–430 (2015).

[42] B. Epshtein, E. Ofek & Y. Wexler ; «Detecting Text in Natural Scenes with Stroke Width Transform» ; IEEE Conference on Computer Vision and Pattern Recognition (CVPR) p. 2963–2970 (2010).

[43] W. Huang, Z. Lin, J. Yang & Y. Wang ; «Text Localization in Natural Images using Stroke Feature Transform and Text Covariance Descriptors» ; IEEE International Conference on Computer Vision (ICCV) p. 1241–1248 (2013).

[44] S. Lu, T. Chen, S. Tian, J. Lim & C. L. Tan ; «Scene text extraction based on edges and support vector regression» ; International Journal on Document Analysis and Recognition **18**, p. 125–135 (2015).

[45] M. Buta, L. Neumann & J. Matas ; «FASText : Efficient Unconstrained Scene Text Detector» ; IEEE International Conference on Computer Vision (ICCV) p. 1206–1214 (2015).

[46] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou & Z. Cao ; «Scene Text Detection via Holistic, Multi-Channel Prediction» ; CoRR **abs/1606.09002**, p. – (2016).

[47] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu & X. Bai ; «Multi-Oriented Text Detection with Fully Convolutional Networks» ; IEEE Conference on Computer Vision and Pattern Recognition p. 4159–4167 (2016).

[48] T. He, W. Huang, Y. Qiao & J. Yao ; «Text-Attentional Convolutional Neural Network for Scene Text Detection» ; IEEE Transactions on Image Processing **25**, p. 2529–2541 (2016).

[49] K. Simonyan & A. Zisserman ; «Very Deep Convolutional Networks for Large-Scale Image Recognition» ; Proceedings of International Conference on Learning Representations (2015).

[50] M. Liao, B. Shi, X. Bai, X. Wang & W. Liu ; «TextBoxes : A Fast Text Detector with a Single Deep Neural Network» ; Association for the Advancement of Artificial Intelligence p. 4161–4167 (2017).

[51] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu & A. Berg ; «Single Shot MultiBox Detector» ; European Conference on Computer Vision p. 21–37 (2016).

[52] Y. Liu & L. Jin ; «Deep Matching Prior Network : Toward Tighter Multi-oriented Text Detection» ; IEEE Conference on Computer Vision and Pattern Recognition (CVPR) p. – (2017).

[53] W. He, X.-Y. Zhang, F. Yin & C.-L. Liu ; «Deep Direct Regression for Multi-Oriented Scene Text Detection» ; The IEEE International Conference on Computer Vision (ICCV) p. – (2017).

[54] B. Shi, X. Bai & S. J. Belongie ; «Detecting Oriented Text in Natural Images by Linking Segments» ; CoRR **abs/1703.06520**, p. – (2017).

[55] H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han & E. Ding ; «WordSup : Exploiting Word Annotations for Character Based Text Detection» ; The IEEE International Conference on Computer Vision (ICCV) p. – (2017).

[56] A. Polzounov, A. Ablavatski, S. Escalera, S. Lu & J. Cai ; «Word-Fence : Text Detection in Natural Images with Border Awareness» ; CoRR **abs/1705.05483**, p. – (2017).

[57] Y. Wu & P. Natarajan ; «Self-Organized Text Detection With Minimal Post-Processing via Border Learning» ; The IEEE International Conference on Computer Vision (ICCV) p. – (2017).

[58] G. Papandreou, L.-C. Chen, K. P. Murphy & A. L. Yuille ; «Weakly- and Semi-Supervised Learning of a Deep Convolutional Network for Semantic Image Segmentation» ; The IEEE International Conference on Computer Vision (ICCV) p. – (2015).

[59] S. Tian, S. Lu & C. Li ; «WeText : Scene Text Detection Under Weak Supervision» ; The IEEE International Conference on Computer Vision (ICCV) p. – (2017).

[60] B. Su & S. Lu ; «Accurate Scene Text Recognition based on Recurrent Neural Network» ; Asian Conference on Computer Vision p. 35–48 (2014).

[61] U. Boy, A. Mishra, K. Alahari & C. Jawahar ; «Scene Text Recognition and Retrieval for Large Lexicons» ; Asian Conference on Computer Vision p. 494–508 (2014).

[62] B. Shi, X. Bai & CongYao ; «An End-to-end Trainable neural network for image-based sequence recognition and its application to scene text recognition» ; IEEE Transactions on Pattern Analysis and Machine Intelligence, **PP,**, p. – (2016).

[63] B. Shi, X. Wang, P. Lyu, C. Yao & X. Bai ; «Robust Scene Text Recognition with Automatic Rectification» ; IEEE Conference on Computer Vision and Pattern Recognition (CVPR) p. 4168–4176 (2016).

[64] X. Yang, D. He, Z. Zhou, D. Kifer & C. L. Giles ; «Learning to Read Irregular Text with Attention Mechanisms» ; Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence p. 3280–3286 (2017).

[65] Y. Gao, Y. Chen, J. Wang & H. Lu ; «Reading Scene Text with Attention Convolutional Sequence Modeling» ; CoRR **abs/1709.04303** (2017)`http://arxiv.org/abs/1709.04303`.

[66] M. Jaderberg, A. Velaldi & A. Zisserman ; «Deep Features for Text Spotting» ; European Conference on Computer Vision p. 512–528 (2014).

[67] X. Bai, C. Yao & W. Liu ; «Strokelets : A Learned Multi-Scale Mid-Level Representation for Scene Text Recognition» ; IEEE Transactions on Image Processing **25**, p. 2798–2802 (2016).

[68] A. Mishra, K. Alahari & C. Jawahar ; «Enhancing Energy Minimization Framework for Scene Text Recognition with Top-Down Cues» ; Journal on Computer Vision and Image Understanding **145**, p. 30–42 (2016).

[69] C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao & Z. Zhang ; «Scene Text Recognition using Path-based Tree-structured Character Detection» ; IEEE Conference on Computer Vision and Pattern Recognition p. 2961–2968 (2013).

[70] P. He, W. Huang, Y. Qiao, C. C. Loy & X. Tang ; «Reading Scene Text in Deep Convolutional Sequences» ; Association for the Advancement of Artificial Intelligence p. 3501–3508 (2016).

[71] A. Bissaco, M. Cummins, Y. Netzer & H. Neven ; «PhotoOCR : Reading Text in Uncontrolled Conditions» ; IEEE International Conference on Computer Vision (ICCV) p. 785–792 (2013).

[72] K. Wang, B. Babenko & S. Belongie ; «End-to-End Scene Text Recognition» ; International Conference on Computer Vision p. 1457–1464 (2011).

[73] L. Neumann & J. Matas ; «Real-Time Scene Text Localization and Recognition» ; IEEE Conference on Computer Vision and Pattern Recognition p. 3538–3545 (2012).

[74] L. Neumann & J. Matas ; «Efficient Scene Text Localization and Recognition with Local Character Refinement» ; International Conference on Document Analysis and Recognition (ICDAR) p. 746–750 (2015).

[75] C. Yao, X. Bai & W. Liu ; «Un Unified Framework for Multioriented Text Detection and Recognition» ; IEEE Transactions on Image Processing **23**, p. 4737–4749 (2014).

[76] H. Li & C. Shen ; «Reading Car License Plates Using Deep Convolutional Neural Networks and LSTMs» ; CoRR **abs/1601.05610**, p. – (2016).

[77] M. Busta, L. Neumann & J. Matas ; «Deep TextSpotter : An End-To-End Trainable Scene Text Localization and Recognition Framework» ; The IEEE International Conference on Computer Vision (ICCV) p. – (2017).

[78] M. Jaderberg, K. Simonyan, A. Velaldi & A. Zisserman ; «Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition» ; Workshop on Deep Learning, NIPS (2014).

[79] R. Girshick, J. Donahue, T. Darrell & J. Malik ; «Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation» ; IEEE Conference on Computer Vision and Pattern Recognition p. 580–587 (2014).

[80] R. Girshick ; «Fast R-CNN» ; International Conference on Computer Vision (ICCV) (2015).

[81] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, K. Ashida, H. Nagai, M. Okamoto, H. Yamamoto, H. Miyao, J. Zhu, W. Ou, C. Wolf, J.-M. Jolion, L. Todoran, M. Worring & X. Lin ; «ICDAR 2003 Robust Reading Competitions : Entries, Results and Future Directions» ; (2005).

[82] A. Gupta, A. Vedaldi & A. Zisserman ; «Synthetic Data for Text Localisation in Natural Images» ; IEEE Conference on Computer Vision and Pattern Recognition (CVPR) p. 2315–2324 (2016).

[83] C. Yao, X. Bai, W. Liu, Y. Ma & Z. Tu ; «Detecting Texts of Arbitrary Orientations in Natural Images» ; IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2012).

[84] C. Wolf & J.-M. Jolion ; «Object count/Area Graphs for the Evaluation of Object Detection and Segmentation Algorithms» ; International Journal of Document Analysis **8**, p. 280–296 (2006).

[85] T. Ojala, M. Pietikäinen & T. Mäenpää ; «Multiresolution gray-scale and rotation invariant texture classification with local binary patterns» ; IEEE Transactions on Pattern Analysis and Machine Intelligence **24**, p. 971–987 (2002).

[86] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke & A. Rabinovich ; «Going deeper with convolutions» ; IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015).

[87] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva & A. Torralba ; «Learning Deep Features for Discriminative Localization» ; IEEE International conference on Computer Vision and Pattern Recognition (CVPR) (2016c).

[88] P. Dollár, R. Appel, S. Belongie & P. Perona ; «Fast Feature Pyramids for Object Detection» ; IEEE Transactions on Pattern Analysis and Machine Intelligence **36**, p. 1532–1545 (2014).

[89] J. Redmon & A. Farhadi ; «YOLO9000 : Better, Faster, Stronger» ; CoRR **abs/1612.08242** (2016).

[90] J. Canny ; «A Computational Approach to Edge Detection» ; IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-8**, p. 679–698 (1986).

[91] P. Krahenbuhl & V. Koltun ; «Geodesic Object Proposals» ; European Conference on Computer Vision p. 725–739 (2014).

[92] S. Manen, M. Guillaumin & L. V. Gool ; «Prime Object Proposals with Randomized Prim's Algorithm» ; IEEE International Conference on Computer Vision p. 2536–2543 (2013).

[93] J. PontTuset, P. Arbelaez, J. Barron, F. Marques & J. Malik ; «Multiscale Combinatorial Grouping for Image Segmentation and Object Proposal Generation» ; IEEE Transactions on Pattern Analysis and Machine Intelligence **39**, p. 128–140 (2017).

[94] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár & C. L. Zitnick ; «Microsoft COCO : Common Objects in Context» ; CoRR **abs/1405.0312** (2014).

[95] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg & L. Fei-Fei ; «ImageNet Large Scale Visual Recognition Challenge» ; International Journal of Computer Vision (IJCV) **115**, p. 211–252 (2015).

[96] J. Redmon, S. Divvala, R. Girshick & A. Farhadi ; «You Only Look Once : Unified, Real-Time Object Detection» ; CoRR **abs/1506.02640** (2015).

[97] C. Yi & Y. Tian ; «Text String Detection from Natural Scenes by Structure-Based Partition and Grouping» ; IEEE Transactions on Image Processing **20**, p. 2594–2605 (2011).

[98] X. Zhou, C. Yao, H. Wen & J. Liang ; «EAST : An Efficient and Accurate Scene Text Detector» ; IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017).

[99] T. He, W. Huang, Y. Qiao & J. Yao ; «Accurate Text Localization in Natural Image with Cascaded Convolutional Text Network» ; CoRR **abs/1603.09423** (2016).

[100] H. Freeman & R. Shapira ; «Minimum-Area Encasing Rectangle for an Arbitrary Closed Curve» ; Communications of the ACM **18**, p. 409–413 (1975).

[101] «Aging in Place : Growing Old at Home» ; https://www.nia.nih.gov/health/aging-place-growing-old-home.

[102] R. Igual, C. Medrano & I. Plaza ; «Challenges, issues and trends in fall detection systems» ; BioMedical Engineering OnLine (2013).

[103] «The Top Medication Reminder Apps for Patients» ; (2017). http://www.pharmacytimes.com/contributor/christina-tarantola/2017/12/the-top-medication-reminder-apps-for-patients ; latest access : 10-Jan-2018.

[104] «GPS Tracking Systems Help The Elderly» ; https://www.tracking-system.com/for-consumers/gps-elderly-tracking-system.html ; latest access : 10-Jan-2018.