Understanding Text in Scene Images

Thesis submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Engineering

by

Anand Mishra
200907004
anand.mishra@research.iiit.ac.in

Center for Visual Information Technology
International Institute of Information Technology
Hyderabad - 500 032, INDIA
December 2016

International Institute of Information Technology

Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis titled "Understanding Text in Scene Images" by Anand Mishra, has been carried out under our supervision and is not submitted elsewhere for a degree.

____________	______________________________
Date	Adviser: C. V. Jawahar
December 14, 2016	
Date	Adviser: Karteek Alahari (electronically signed)

To the *primal cause* of all creation

Acknowledgments

First of all, I thank my PhD adviser Prof. C. V. Jawahar for providing me excellent opportunities to excel in many aspects. His hard-work and dedication for the quality inspire me a lot. Next, my big thanks goes to my co-adviser Dr. Karteek Alahari. His detail comments on methods, presentations and paper drafts made this thesis possible. His thirst for the perfection is a must quality for a researcher. Thank you Dr. Karteek for all the training you provided during my PhD. I am also grateful to my PhD committee members: Prof. Santanu Chaudhury, Dr. Parag Chaudhuri, and Dr. Avinash Sharma for their valuable comments and suggestions on this thesis draft. I am grateful to Dr. Anoop Namboodiri for various useful suggestions on my PhD work. I also thank Prof. P. J. Narayanan and all other faculty members of IIIT for developing a wonderful research and academic environment in the institute. Especially, I thank Dr. K. Srinathan for his inspiring lectures and suggestion during struggling time in my life. He is a great motivator, and a great teacher.

My next thanks goes to my sponsors: Microsoft Research India, Google India, XRCI, IEEE, ACM-India for travel grants. I especially thank to Microsoft Research India for issuing me the PhD fellowship. This fellowship helped me a lot to keep myself motivated during PhD. I also thank IIIT - Sri city for giving me opportunity to teach during my PhD.

Logistic support in IIIT is excellent. I thank Satya Garu for all his support during reimbursements of international and national travels, and in the other logistics. I thank CVIT data annotation team: Phani Garu, Rajan and Nandini. Their excellent work made it possible to release the datasets we use in this thesis. I am also thankful to IIIT server room guys: Dharmendra and company for resolving PC related issues many times during my stay at IIIT. My next thank goes to the guy whom I used meet first in the morning during my PhD: Mutiyalu. I thank him for his excellent duty of making our workplace dust free.

I have been fortunate to work with several undergrad students (Dheeraj, Vibhor, Deepan, Vaidehi, Pranav, Ashutosh, Shyam) during my PhD. Working with these students was an excellent experience

for me. Thank you guys. Friends are essential for any accomplishments. I am fortunate to have great friends both inside and outside IIIT. I thank Praveen K. and Ajeet K. Singh (for the time we have shared), Nagendar, Nisarg, Madhu (for various discussions during early days of my PhD), Lalit (a wonderful friend whom I met during my Sri city visits, I still remember our relaxing dinners at Tada on weekends, and awesome visits to Srikalhasti temple and pulicat lake), Aniket (for all the sense of humor we shared), Yasaswi and Rajvi (for some useful review on paper drafts and many interesting discussions), Sneha (a family friend at Hyderabad), Amitesh, Manoj and Ashutosh (BIT Mesra friends for always keeping me motivated), Riyaj, Sridhar and Ravi (for all the PhD frustrations we shared), Bhaskar and Girjesh (friends from my home town for all their support).

I am fortunate to have great support from my family during PhD. I thank to my wife Swati who came in my life towards the end of my PhD. Nevertheless, she motivated me to complete the finishing things of this thesis. She has been the best friend and a great motivator of mine during last few months. I am also thankful to my little nieces and nephews for the wonderful childish moments we spent during vacations. Thank you little kids. My parents always supported me on my decisions. I do not have words for their support, their hided tears, and their sacrifice. Their blessing is my strength. Finally, I believe everything is created for a cause (so is this thesis), and their is the primal cause of all creation. Last but not the least, I thank Lord Vishnu as *the primal cause of all creation*.

Abstract

With the rapid growth of camera-based mobile devices, applications that answer questions such as, “What does this sign say?" are becoming increasingly popular. This is related to the problem of optical character recognition (OCR) where the task is to recognize text occurring in images. The OCR problem has a long history in the computer vision community. However, the success of OCR systems is largely restricted to text from scanned documents. Scene text, such as text occurring in images captured with a mobile device, exhibits a large variability in appearance. Recognizing scene text has been challenging, even for the state-of-the-art OCR methods. Many scene understanding methods recognize objects and regions like roads, trees, sky in the image successfully, but tend to ignore the text on the sign board. Towards filling this gap, we devise robust techniques for scene text recognition and retrieval in this thesis.

This thesis presents three approaches to address scene text recognition problems. First, we propose a robust text segmentation (binarization) technique, and use it to improve the recognition performance. We pose the binarization problem as a pixel labeling problem and define a corresponding novel energy function which is minimized to obtain a binary segmentation image. This method makes it possible to use standard OCR systems for recognizing scene text. Second, we present an energy minimization framework that exploits both bottom-up and top-down cues for recognizing words extracted from street images. The bottom-up cues are derived from detections of individual text characters in an image. We build a conditional random field model on these detections to jointly model the strength of the detections and the interactions between them. These interactions are top-down cues obtained from a lexicon-based prior, i.e., language statistics. The optimal word represented by the text image is obtained by minimizing the energy function corresponding to the random field model. The proposed method significantly improves the scene text recognition performance. Thirdly, we present a holistic word recognition framework, which leverages scene text image and synthetic images generated from lexicon words. We then recognize the text in an image by matching the scene and synthetic image features

with our novel weighted dynamic time warping approach. This approach does not require any language statistics or language specific character-level annotations.

Finally, we address the problem of image retrieval using textual cues, and demonstrate large-scale text-to-image retrieval. Given the recent developments in understanding text in images, an appealing approach to address this problem is to localize and recognize the text, and then query the database, as in a text retrieval problem. We show that this approach, despite being based on state-of-the art methods, is insufficient, and propose an approach without relaying on an exact localization and recognition pipeline. We take a query-driven search approach, where we find approximate locations of characters in the text query, and then impose spatial constraints to generate a ranked list of images in the database.

We evaluate our proposed methods extensively on a number of scene text benchmark datasets, namely, street view text, ICDAR 2003, 2011 and 2013, and a new dataset IIIT 5K-word, we introduced, and show better performance than all the comparable methods. The retrieval performance is evaluated on public scene text datasets as well as three large datasets, namely, IIIT scene text retrieval, Sports-10K and TV series-1M, we introduced.

Contents

Chapter Page

List of Figures

Figure Page

List of Tables

Chapter 1

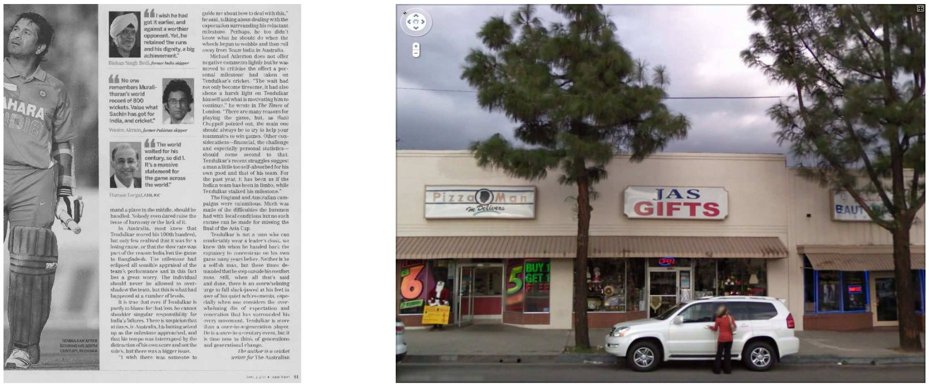
Introduction

According to a study, one trillion photos were captured in the year 2015[1], and this number is going to keep increasing! In the context of such large data collections that continue to grow, there are many challenging problems like recognizing and retrieving relevant content. Text in the scene images or videos can play a crucial role in understanding images. Given the rapid growth of camera-based applications readily available on mobile phones, understanding scene text is more important than ever, and application that are able to answer questions such as, “What does this sign say?" are becoming increasingly popular. This is related to the problem of optical character recognition (OCR), which has a long history in the computer vision community. However, the success of OCR systems is largely restricted to text from scanned documents. Scene text exhibits a large variability in appearance, and can prove to be challenging even for state of the art OCR methods. Many scene understanding methods recognize objects and regions like roads, trees, sky in the image successfully, but tend to ignore the text on sign boards. The goal of this thesis is to fill this gap in understanding the scene by devising robust techniques for scene text recognition. Figure 1.1 explains the larger goal of this thesis.

1.1 Scene text understanding: problems, challenges and applications

Substantial research effort has been dedicated to scene text understanding in the last decade [22, 30, 36, 40, 44, 67, 112, 114, 117, 133, 165, 168, 170, 172]. In this section, we discuss the challenges in scene text understanding and its applications, and finally provide a literature survey.

[1] `http://mylio.com/one-trillion-photos-in-2015/`



(a)

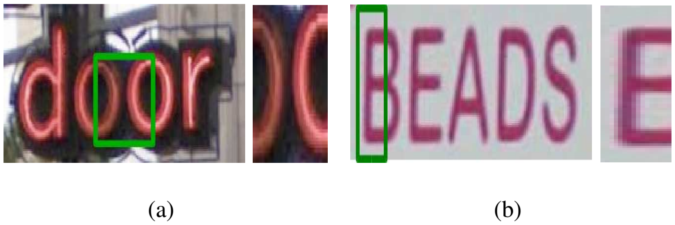
(b)

Figure 1.1 Images containing text can be categorized into two broad categories: (a) scanned documents, (b) scene image containing text. Contrary to problem of understanding scanned documents like (a), the problem of reading text in scene images is surprisingly challenging. At the same time, it has many potential applications. Consider a typical street scene image taken from Google street view shown in (b). It contains a few very prominent sign boards (with text) on the building. Given this text, one might infer that the photo is taken in an English-speaking country, and what this shop sells? This thesis attempts to address the broad area of scene text understanding, and addresses the problem of recognition and retrieval in this context

1.1.1 Challenges: scene text recognition $\neq$ OCR

Recognition of scene text is closely related to the classical problem of optical character recognition (OCR). For cleanly scanned English-language documents, OCR is no longer considered a problem. Contrary to, printed scanned documents, recognizing scene text images taken in the wild has many additional challenges, for example:

1. **Identifying texts in natural scenes .** In natural scenes, numerous objects, such as buildings, bikes, cars or parts of them have similar shape and appearances to text. For example corner of a building can easily be recognized as *I* or wheels of vehicle as *O*. This challenges the text detection techniques to discriminate text from non-text.



(a) (b)

Figure 1.2 Typical challenges in character detection. (a) A window containing parts of two o's is falsely detected as x, (b) a window containing a part of the character B is recognized as E.

2. **Binarization.** Traditional OCR engines first binarize word image and then segment characters using connected component information. Binarization of scene texts are seldom perfect and contain noise.

3. **Inter and intra-class confusion.** Scene text also pose challenge to a typical sliding window based object detector due to inter-class and intra-class confusion. Often, part of a character class has a similar appearance to some other character. Similarly, part of two character classes look similar to some other character. A couple of such examples are shown in Figure 1.2.

4. **Image specific challenges.** Often scene text have fancy/stylish fonts usually to attract the attention of viewers. Recognizing such unseen characters become challenging for a classifier, which is trained on normal fonts. Since it is unlikely to assume that images are captured in frontal view, there can be high variability in view points of the images. Unlike scanned documents where text is largely horizontal, scene text can have any orientations. Further, scene text images often come with the challenges like low contrast, low resolutions (especially, in digital-born images), blur. Few example images from two public datasets: the SVT and IIIT-5K are shown in Figure 1.3 to illustrate these challenges in the context of scene text recognition.

5. **No well defined layout.** Unlike scanned books or magazines which have a clean layout and contain text predominately, text regions and amount of text in scene images vary from image to image. It makes text region localization and word/line segmentation harder than typical OCR document images.

6. **Multilingual texts.** Most of the Latin languages have few character classes, non-European languages such as Chinese, Japanese and Korean and most of the Indian languages have hundreds or thousands of character classes with a lot of inter-class similarities. In multilingual environments, OCR on scanned documents remains a research problem [54], while text recognition in street

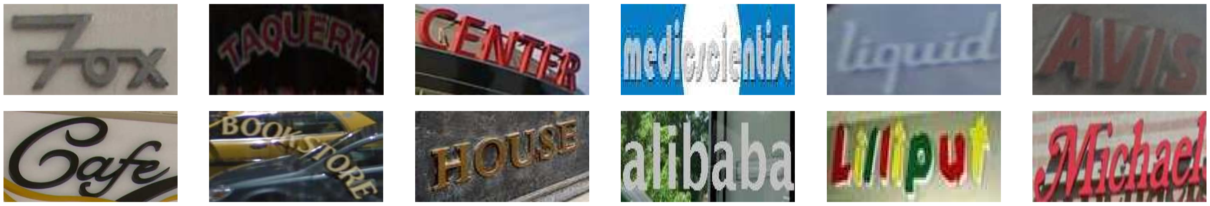


Figure 1.3 Challenges in scene text recognition. A few sample images from the SVT [165] and IIIT 5K-word [108] datasets are shown to highlight the variation in view point, orientation, non-uniform background, non-standard font styles and also issues such as occlusion, noise, and inconsistent lighting. Standard OCRs perform poorly on these datasets.

scene is much more difficult. The problem of recognition of multilingual scene text is beyond the scope of this thesis. Nevertheless, few of the techniques proposed in this thesis can be applied for multiple languages in the future.

7. **Lack of language context.** Scene texts often appear as a word or group of few words. Hence, applying larger contexts, such as, at the level of sentence or paragraph is many a times not possible.

1.1.2 Applications

There are many applications of scene text understanding in real life. Few of them are listed here.

1. **Multimedia indexing and retrieval.** Most of the available video and image search in the modern search engines are meta tag based. The search results look impressive only when there are appropriate meta-tags and the images are tagged correctly. However, often meta-tags are noisy, unavailable (consider images/video collected from personal photography), incomplete, or do not describe everything in the image or video. Recognizing the text within the image or video can play an important role in indexing and retrieval of multimedia data.

2. **Cross-lingual access.** Recognizing text in sign boards can be very useful for cross-lingual access. Consider following situation: a north Indian tourist who does not know Telugu (a south Indian language) is roaming in the Telangana state, where most of the sign boards are in Telugu. Recognizing these sign boards, and then translating it to the native language of tourists can prove very useful for them.

3. **Auto navigation.** For successful auto navigation systems, or driver-less cars, understanding text or sign boards can be very crucial.

4. **Apps for visually impaired.** Reading scene images in streets can be very useful for building apps which can assist visually impaired people while in markets andor other public places. Consider an app which can assist visually impaired people by reading grocery item names while shopping at a grocery store. Such apps can only be built when we have satisfactory progress in reading text in scene images.

5. **Industrial automation.** Recognizing text on packages, containers, houses, and maps has broad applications related to industrial automation. For example, recognition of addresses on envelopes is used in mail sorting systems. Automatic identification of container numbers improves logistics efficiency. Recognition of house numbers and text in maps can immensely benefit automatic geocoding systems.

1.1.3 Prior art

Scene text understanding can be approached by tackling smaller tasks such as, (i) scene character recognition, (i) text localization, and (iii) cropped word recognition. They have been tackled either individually [31,36,40], or jointly [67,114,165,168]. We review the prior art on scene text understanding area in this section.

- **Scene character recognition.** Isolated character recognition is one of the oldest problems in pattern recognition. In the context of scanned fonts and handwritten digits (e.g., MNIST [93]) this problem is considered as solved with error rates as low as 0.1% having been reported [100]. On the other hand, recognizing scene characters poses lots of challenges due to high variation in fonts, illumination, perspective distortion and background noise. To recognize characters of a single clean font, naive features such as principle component analysis (PCA) on raw feature with simple linear classifiers are sufficient, but these features are not effective in the case of scene characters.

  One of the earliest works on large-scale natural scene character recognition was presented in [36]. This work develops a multiple kernel learning (MKL) approach using a set of shape-based features. Recent works [109, 165] have improved over this with histogram of gradient (HOG) features [35]. Variations of HOG features have also been explored for scene character classification [110, 176]. In [177] author present a sparse representation of HOG feature for character classification. This method is inspired by Ren and Ramanan's recent work [131] on sparsifying histogram of gradient features. It achieves better performance than naive HOG features. Sheshadri *et al.* [142] applied an exemplar SVM to recognize distorted characters in scene images. Shi *et al.* [143] proposed a method using deformable part based models and sliding window classification to localize and recognize characters in scene images. In [172], a learned representation, namely, strokelets was proposed for character recognition. A histogram of these strokelets along with random forest classifier is used for character recognition. More recently, deep features [67] have been shown to outperform other methods on scene character classification.

- **Scene text localization.** The major techniques to solve text localization problem can be grouped into four categories: (i) texture based, (ii) component based, (iii) mid-level feature based, and (iv) unsupervised feature learning based. In texture-based methods, the scene image is scanned at different scales using sliding windows, and text and non-text regions are classified. Some of the recent works like [55, 165, 166] fall in this category. Often histogram-of-gradient (HOG) features along with SVM or random forest classifiers are used in such settings. The major drawback with such methods are high computational complexity and low precision character detection. In component based methods, low-level cues are used to discard majority of the background pixels, and then from the rest of the pixels, potential character candidates are formed using a set of heuristic properties. Further, stroke width, color consistency, aspect ratio and some other simple features are used to prune the false positives. The popular methods in this category are stroke width transform [40], class specific extremal regions [115] and segmentation based methods [61, 104]. In recent years mid-level feature based techniques have gained interest. In these methods, structural characteristics of characters at multiple scale are captured. The prominent methods in this category are: strokelets [172] and part based method [143]. The fourth category of text localization methods are unsupervised feature learning based techniques [33, 67, 167]. These techniques learn the text vs non text features and classifiers in an unsupervised manner. These methods achieve a noticeable success. In this thesis, our focus is more on enhancing recognition performance while relaying on existing text localization methods.

- **Cropped word recognition.** The core components of a typical cropped word recognition framework are: localize the characters, recognize them, and use statistical language models to compose the characters into words. In the following, we review the prior art. The reader is encouraged to refer to [174] for a more comprehensive survey of scene text recognition methods.

  A popular technique for localizing characters in an OCR system is to binarize the image and determine the potential character locations based on connected components [56]. Such techniques have also been adapted for scene text recognition [112], although with limited success. This is mainly because obtaining a clean binary output for scene text images is often challenging. An alternative approach is proposed in [144] using gradient information to find potential character locations. More recently, Yao *et al.* [172] proposed a mid-level feature based technique to localize characters in scene text.

  A study on human reading psychology shows that our reading improves significantly with prior knowledge of the language [130]. Motivated by such studies, OCR systems have used, often in post-processing steps [56, 159], statistical language models like n-grams to improve their performance. Bigrams or trigrams have also been used in the context of scene text recognition as a post-processing step, e.g., [18]. A few other works [38, 39, 156] integrate character recognition and linguistic knowledge to deal with recognition errors. For example, [156] computes n-gram probabilities from more than 100 million characters and uses a Viterbi algorithm to find the correct word. The method in [38], developed in the same year as our CVPR 2012 work [109], builds a graph on potential character locations and uses n-gram scores to constrain the inference algorithm to predict the word.

  The word recognition problem has been looked at in two contexts— with [51, 109, 133, 165, 167] and without [108, 169, 170] the use of an image-specific lexicon. In the case of image-specific lexicon-driven word recognition, also known as the closed vocabulary setting, a list of words is available for every scene text image. The task of recognizing the word now reduces to that of finding the best match from this list. This is relevant in many applications, e.g., recognizing text in a grocery store, where a list of grocery items can serve as a lexicon. Wang *et al.* [167] adapted a multi-layer neural network for this scenario. In [165], each word in the lexicon is matched to the detected set of character windows, and the one with the highest score is reported as the predicted word. In one of our work [51] (Chapter 5), we compared features computed on the entire scene text image and those generated from synthetic font renderings of lexicon words with a novel weighted dynamic time warping (wDTW) approach to recognize words. In [133], Rodriguez-Serrano and Perronnin proposed to embed word labels and word images into a common Euclidean space, wherein the text recognition task is posed as a retrieval problem to find the closest word label for a given word image. While all these approaches are interesting, their success is largely restricted to the closed vocabulary settings and they cannot be easily extended to the more general cases, for instance, when image-specific lexicon is unavailable. Weinman *et al.* [170] proposed a method to address this issue, although with a strong assumption of known character boundaries, which are not trivial to obtain with high precision on the datasets we use. The work in [169] generalizes their previous approach by relaxing the character-boundary requirement. It is, however, evaluated only on "roughly fronto-parallel" images of signs, which are less challenging than the scene text images used in our work. More recently, deep convolutional network based methods gained attention for scene text understanding problems (see [22, 60, 65, 67, 167] for example). These approaches are very effective in general. We compare these related contemporary methods with our approach in this thesis.

1.2 Goals of this thesis

This thesis addresses the problem of scene text understanding, and advances this area. In this space, we propose solutions for various associated sub-problems, and show ways of addressing scene text recognition problem. The major goals of this thesis are two fold, (i) designing robust text recognition framework. The state of the art OCR systems lack robustness and show poor performance on scene text recognition. Our thesis aims to improve scene text recognition performance, (ii) once text is recognized, the next immediate need is to retrieve the relevant images or video frames containing query text from a large database. There are large and fast-growing collections of videos and images in personal life and world wide web. Often these images and videos are textually rich, e.g., sports, news and educational videos, images captured in street scenes. Retrieving images or video content based on textual cues can prove very useful in obtaining relevant information. Despite advancements in text localization and recognition, text-to-image retrieval is not trivial due to weak performance of end to end systems for text localization and recognition. Additionally, the retrieval system should also able to deal with large number of distractors, e.g., images or frames which do not contain any text. One of the goals of this thesis is to demonstrate scalable and robust text-to-image retrieval.

To achieve these two primary goals we propose: robust and principled solution to text binarization (Chapter 3), improve scene character classification (Chapter 4), seamless integration of multiple cues in higher order CRF framework for scene text recognition (Chapter 4), holistic representation of word images and matching scheme for synthetic and scene word images (Chapter 5), query driven search technique for text-to-image retrieval (Chapter 6).

Moreover, many of the available datasets for the scene text understanding problem are either very small or not very challenging for real scenario. Hence, as a part of this thesis, we also introduce and benchmark many scene text recognition and retrieval datasets.

1.3 Contributions of this thesis

1. **Robust text segmentation.** We propose a principled framework for text binarization using color and stroke width cues. Use of color and stroke width cues in an optimization framework for text binarization is a major novelty here. We evaluate the performance using various measures, such as pixel-level accuracy, atom-level accuracy as well as recognition results, and compare it with the state of the art methods [58, 74, 77, 106, 116, 119, 138, 171]. To our knowledge, text binarization methods have not been evaluated in such a rigorous setting in the past. Our method generalizes well to multi-script scene texts, video texts as well as historical handwritten documents. In fact, our binarization improves the recognition results of an open source OCR [1] by more than 10% on various public scene text benchmarks. On a benchmark dataset of handwritten images, our method achieves comparable performance to the H-DIBCO 2012 competition winner and the state of the art method [58], which is specifically tuned for handwritten images.

2. **Integrating top-down and bottom-up cues for better recognition.** We propose a joint CRF framework with seamless integration of multiple cues—individual character detections and their spatial arrangements, pairwise lexicon priors, and higher-order priors which can be optimized effectively. The proposed method performs significantly better than other related energy minimization based methods for scene text recognition, and advances the scene text recognition area. Additionally, we analyzed the effectiveness of individual components of the framework, the influence of parameter settings, and the use of convolutional neural network (CNN) based features [67].

3. **Holistic approach to recognition.** We show that holistic word recognition for scene text images is possible with high accuracy, and achieve a significant improvement over prior art. We also present a novel word descriptor which is used efficiently to represent a wide variety of scene text images. The proposed method does not use any language-specific information, and can be easily adapted to any language. This can especially be useful for Indian language scene text recognition and retrieval where there hasn't been significant progress.

4. **Scalable text-to-image retrieval on image/video dataset.** We propose a query-driven approach for text-to-image retrieval from a large database without relying on an exact localization-recognition pipeline. We demonstrate our results on web-scale video and image datasets. The proposed method achieves significant performance gain in mAP over recent methods [114, 165].

5. **Datasets.** Most of the publicly available datasets in this area such as, ICDAR 2003/2011/2013 contain only few hundred images. We have introduced IIIT-5K word dataset containing 5000 images. The dataset is not only five time bigger than other related dataset but also more challenging [108]. Moreover, we also introduce datasets for problems like text-to-image retrieval and character localization and recognition. All our datasets are publicly available on our project page [2], and have been widely used by the community around the globe [67, 133, 172, 174].

In short, this thesis presents principled solutions to an important computer vision problem.

1.4 Publications

Part of the work described in this thesis has previously been presented as the following publications. The total number of citations for these publications is 383 [source: Google scholar, December 10, 2016].[2]

Journal:

1. **Anand Mishra**, Karteek Alahari, C. V. Jawahar, *Enhancing Energy Minimization Framework for Scene Text Recognition with Top-Down Cues*, Computer Vision and Image Understanding, volume 145, pages 30–42, 2016 (Received: 4 April 2015, Revised: 22 August 2015, Accepted: 4 January 2016, Available online: 21 January 2016)

[2] https://goo.gl/MO2gTq

Manuscript under review:

2. **Anand Mishra**, Karteek Alahari, C. V. Jawahar, *Unsupervised Refinement of Color and Stroke Features for Robust Text Binarization*, International Journal on Document Analysis and Recognition (Submitted: 4 December 2015, Revised: 11 September 2016)

Conference:

3. **Anand Mishra**, Karteek Alahari, C. V. Jawahar, *Image Retrieval using Textual Cues*, ICCV 2013, [Citations: 21].

4. Vibhor Goel, **Anand Mishra**, Karteek Alahari and C. V. Jawahar, *Whole is Greater than Sum of Parts: Recognizing Scene Text Words*, ICDAR 2013, [Citations: 41].

5. **Anand Mishra**, Karteek Alahari, C. V. Jawahar, *Scene Text Recognition using Higher Order Language Priors*, BMVC 2012 (Oral), [Citations: 71].

6. **Anand Mishra**, Karteek Alahari, C. V. Jawahar, *Top-down and Bottom-up cues for Scene Text Recognition*, CVPR 2012, [Citations: 160].

7. **Anand Mishra**, Karteek Alahari, C. V. Jawahar, *An MRF Model for Binarization of Natural Scene Texts*, ICDAR 2011 (Oral), [Citations: 76].

Other conference/workshop publications during PhD which are not part of this thesis are as follows:

8. Ashutosh Mishra, Shyam N. Rai, **Anand Mishra** and C. V. Jawahar, *IIIT-CFW: A Benchmark database of Cartoon Faces in the Wild,*, ECCV Workshop 2016.

9. Swetha Sirnam, **Anand Mishra**, G. M. Hegde and C. V. Jawahar, *Efficient Object Annotation for Surveillance and Automotive Applications*, WACV Workshop 2016.

10. Ajeet K. Singh, **Anand Mishra**, Pranav Dabaral and C. V. Jawahar, *A Simple and Effective Method for Script Identification in the Wild*, DAS 2016.

11. Udit Roy, **Anand Mishra**, Karteek Alahari and C. V. Jawahar, *Scene Text Recognition and Retrieval for Large Lexicons*, ACCV 2014.

12. Vijay Kumar, Amit Bansal, Gautom Hari Tulsiyan, **Anand Mishra**, Anoop Namboodari and C.V. Jawahar, *Sparse Document Image Coding for Restoration*, ICDAR 2013.

13. Deepan Gupta*, Vaidehi Chhajer*, **Anand Mishra** and C.V. Jawahar, *A Non-local MRF model for Heritage Architectural Image Completion*, ICVGIP 2012 (*: equal contribution).

14. **Anand Mishra**, Naveen T.S., Viresh Ranjan and C.V. Jawahar, *Automatic Localization and Correction of Line Segmentation Errors*, DAR 2012 (Oral).

15. Dheeraj Mundhra, **Anand Mishra**, C. V. Jawahar, *Automatic Localization of Page Segmentation Errors*, J-MOCR-AND 2011 (Oral).

1.5 Thesis outline

In Chapter 2, we provide the necessary background for the thesis and briefly summarize the aspects of energy minimization directly relevant to the work that follows.

In Chapter 3, we address the problem of word recognition by first binarizing the text and then using an open source OCR. For this, we propose a robust binarization technique for scene text using color and strokes cues. We show results on word images from the challenging ICDAR 2003, ICDAR 2011, street view text, and compare our performance with previously published methods.

Chapter 4 focuses on the energy minimization framework for scene text recognition. Here, we propose a CRF model that exploits both bottom-up and top-down cues for recognizing cropped words extracted from street images. We evaluate our proposed method on street view text, ICDAR 2003, 2011 and 2013 datasets, and IIIT 5K-word, and show better performance than comparable methods. In this chapter, we also perform a rigorous analysis of all the steps in our approach and analyze the results.

In Chapter 5, we describe a holistic approach to word recognition. In this chapter, we also validate our recognition methods for specific cases of scene text recognition such as text on curved surfaces. Finally, we compare the effectiveness of three recognition methods proposed by us towards the end of this chapter.

Chapter 6 presents a query-driven approach for text-to-image retrieval. The retrieval performance is evaluated on public scene text datasets as well as three large datasets, namely, IIIT scene text retrieval, Sports-10K and TV series-AM, we introduce.

Finally, Chapter 7 provides the summary of our work, comparisons with the related contemporary methods and the impact of this thesis. Here, we also discuss the future directions of our thesis.

Chapter 2

Background

In this chapter, we first discuss the evaluation of scene text understanding techniques, and then provide background material relevant to this thesis. Many of the methods proposed in this thesis, especially Chapter 3 and Chapter 4, are inspired by the success of energy minimization techniques in solving computer vision tasks. We discuss the motivation behind energy minimization techniques, introduce the labeling problem, define MRF, and devise MAP-MRF equivalence. Towards the end of this chapter, we provide a background on concepts related to word image matching (Chapter 5) such as dynamic time warping (DTW), and image retrieval (Chapter 6) such as indexing and re-ranking.

2.1 Evolution of scene text understanding

Research on scene text understanding has evolved significantly from the early days. We group the evolution of scene text understanding here.

- **Classical methods (till 2002).** Early scene text understanding methods were natural extension of techniques used for reading scanned documents [163]. The low level image features such as edges, connected components, pixel color, basic preprocessing techniques such as Gaussian filtering and image processing tools were used to address the problem [17]. There was a lack of standard benchmark datasets, and experiments were performed on a small set of images. Some efforts on recognizing and retrieving overlay text in videos as well [68,71] were also made during this time, however with limited success.

- **Advanced image processing based methods (2002 – 2010).** With the introduction of ICDAR 2003 robust reading competition [3], research on scene text localization and recognition has significantly increased. The major document image analysis conference (ICDAR) has started organizing dedicated workshop, namely, CBDAR and robust reading competition in its every edition. Advance image processing techniques such as stroke width transform [40] and symmetrical block patterns [31] were used for addressing text localization. Some basic machine learning techniques such as AdaBoost and energy minimization techniques such as simulated annealing also used in text understanding during this era [30, 31, 171].

- **Modern computer vision based methods (2009 – 2013).** Computer vision has made a significant progress in the last decade in solving many challenging problems such as segmentation, localization, and image classification. Inspired by these solutions, research on scene text started treating words and characters as objects. The seminal works by Wang *et al.* [165, 166] significantly increased the interest in the scene text understanding problems. Many works in this era have been formulated in energy minimization frameworks [108, 109, 165, 170]. The release of challenging public datasets such as SVT [4], IIIT-5K word [108] gathered a lot of attention among the computer vision community.

- **Deep learning based methods (2014 to present).** Deep learning has remarkably advanced the computer vision area by improving the state of the art in many applications by a large margin. The first noticeable work in this area is Coates *et al.* [33] where the authors propose an unsupervised learning approach for character recognition and text localization. Later, Wang *et al.* [167] adapted a multi-layer neural networks for end-to-end scene text localization and recognition. More recently, Jaderberg *et al.* [67] proposed a deep convolutional neural network (CNN) for scene text recognition. Their architecture is trained on synthetic data collection containing 8 million images. This method significantly improved the state of the art of scene text recognition. Contemporary to this method, Su and Lu [153] proposed a recurrent neural network (RNN) framework for recognizing scene text. Here, authors represent word images into a sequential histogram of gradient features, and a RNN is used to classify these sequential features into one of the English words.

Most of our works falls in the category of modern computer vision based methods. Nevertheless, as we show experimentally in this thesis, our methods can take advantage of deep learning based methods to further improve scene text understanding.

2.2 Energy minimization in computer vision

2.2.1 Motivation

Images can be imagined as a physical system molecules and atoms in a physical system corresponds to edges, pixel colors and edge orientation in images [50]. Any physical system in nature is not completely random, rather follows some order. Similarly a natural image can not be a complete random, it has some structure, pattern, and smoothness. Similar to the energy of a physical system, one can define the energy for an image such that its minimum follows some prior.

Over the past two decades, energy minimization has emerged as an essential tool in computer vision [134, 154]. Energy minimization refers to the problem of finding the values at which a function reaches its minimum. Many important and challenging vision tasks like foreground/background segmentation [134], pose estimation [79] and word recognition [109, 170] can be formulated in energy minimization framework. Although the problem of minimizing a general function is NP-hard, there exist certain class of functions (popularly known as submodular functions in discrete optimization literature) which can be minimized efficiently. In this section we provide required background and preliminaries for the energy minimization framework.

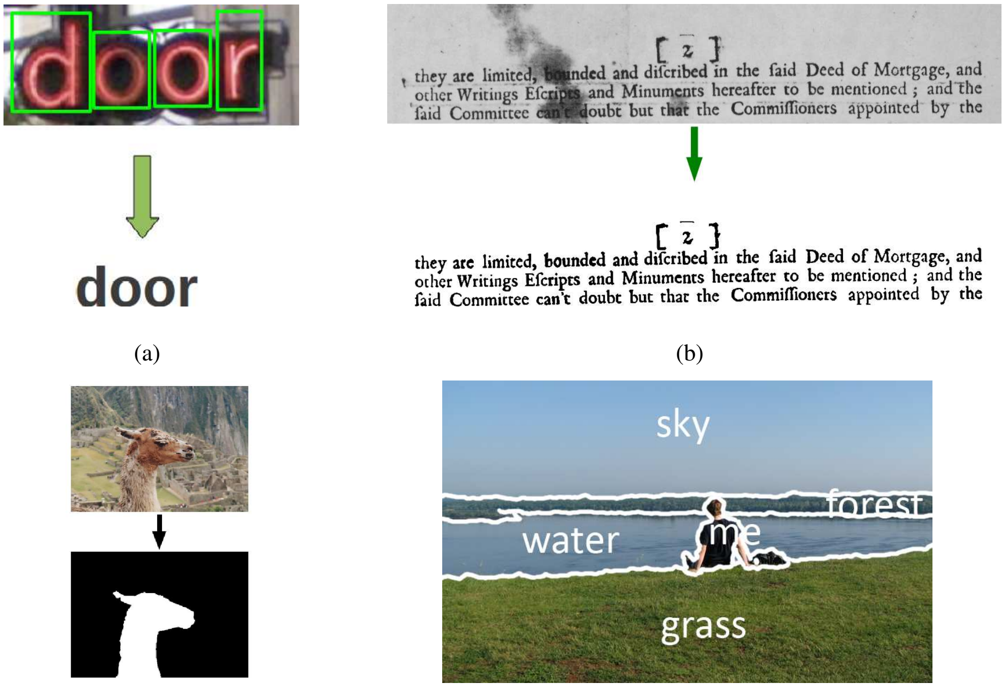
2.2.2 The labeling problem

Many computer vision problems can be formulated as a labeling problem. Few examples are shown in Figure 2.1. Labeling problem is specified in terms of sites and labels. A site often represents a point or a region in an Euclidean space. It could be image pixels, image patches, line segments or corners. We represent sites by a finite set $S = \{1, 2, \dots, m\}$ where each element represents an index of the site. A label is an event that may happen to the site. Labels depend on the problem. Let us the denote set of labels as $L = \{l_1, l_2, \dots, l_n\}$.

Each mapping from site to label is known as a configuration. Let F be the set of all possible configurations. If the cardinality of site and label sets are m and n respectively, then the cardinality of F can be computed as follows:

$$n \times n \times \cdots n \times n \, (m \, times) = n^m. \tag{2.1}$$

For example, consider the foreground-background segmentation as labeling problem. For an image of size 256×256, the cardinality of the configuration will be $2^{256 \times 256}$, which is a huge number. Therefore, any brute force algorithm to search optimal configuration will take exponential time, and is not feasible.



they are limited, bounded and deſcribed in the ſaid Deed of Mortgage, and other Writings Eſcripts and Minuments hereafter to be mentioned ; and the ſaid Committee can't doubt but that the Commiſſioners appointed by the

(a) (b)

(c) (d)

Figure 2.1 Many vision problems can be posed as a labeling problem. (a) Problem of word recognition can be formulated as a labeling problem where each window can take one of the label $L = \{A, \dots, Z\}$. (b) The binarization problem can also be viewed as binary labeling problem with label $L = \{0(text), 1(background)\}$. Similarly, (c) binary segmentation, and (d) semantic segmentation can also be formulated as a labeling problem. Image courtesy: (c) Rother *et al.* [134] (d) Blog by Roman Shapovalov [5].

In general, labeling problem is an NP-hard problem. Fortunately only a small number of the solutions are good and are of our interest, which leads to concepts like optimality criteria and the optimal solution.

Why optimization in vision? Every vision process is inherently uncertain. These uncertainties arise from various sources like noise, degradation, appearance, pose, and visual interpretation. The optimization techniques give powerful tools to deal with such uncertainty. Mostly, in an optimization framework our aim is to minimize an objective function. The objective function is a function from the solution space to the goodness of the solution. Often these objective functions are non-convex in nature. Thus they can not be trivially minimized. However, there are a class of functions which can be minimized efficiently. We will discuss those functions as we move forward.

2.2.3 MAP estimation

Given an observation z, posterior probability of a configuration x can be defined as:

$$P(x|z) = \frac{P(x)p(z|x)}{p(z)}. \tag{2.2}$$

Let x^* be the optimal solution, then the risk of estimating this solution can be defined as follows:

$$R(x^*) = \sum_{x \in F} C(x, x^*)P(x|z). \tag{2.3}$$

Here $C(\cdot, \cdot)$ is a cost function. A typical cost function can be of one of the following forms:

$$C(x, x^*) = ||x^* - x||^2. \tag{2.4}$$

$$C(x, x^*) = \begin{cases} 0 & |x^* - x| \leq \delta, \\ 1 & \text{otherwise.} \end{cases} \tag{2.5}$$

We can write equation 2.3 as:

$$R(x^*) = \sum_{|x^*-x| \leq \delta} C(x, x^*)P(x|z)dx + \sum_{|x^*-x| > \delta} C(x, x^*)P(x|z)dx, \tag{2.6}$$

where δ is arbitrarily small positive number. For simplicity, let us assume we use a cost function 2.5. Then the above equation can be written as.

$$R(x^*) = 1 - \sum_{|x^*-x| < \delta} P(x|z)dx. \tag{2.7}$$

If we take $\delta \to 0$, then risk can be rewritten as:

$$R(x^*) = 1 - kP(x|z), \tag{2.8}$$

where k is the volume of space containing all points for which $|x^* - x| < \delta$. We have to minimize this risk, i.e., we have to maximize $P(x|z)$, which leads to the conclusion that the optimum solution x^* is the one which maximizes posterior $P(x|z)$; or from (2.2), we can write the optimal configuration as:

$$x^* = \underset{x}{\operatorname{argmax}}\, p(z|x)P(x). \tag{2.9}$$

Images are not completely random; in other words, pixels are mutually dependent. The question then arise then is – how can such a dependency be modeled? We can treat the image as a Markov random field (MRF) and model contextual constraints.

2.2.4 Markov random field and conditional random field

Let $S = \{1, 2, \ldots, m\}$ be a set of sites and $L = \{l_1, l_2, \ldots, l_n\}$ be a set of levels. Further suppose $F = \{F_1, F_2, \ldots, F_m\}$ are a family of random variables defined on a set S. F can take all possible configuration defined over sites S and Labels L. We define term $P(x)$ as probability of a random vector F taking a particular configuration x.

Then F is said to be a **Markov random field** [95] on S with respect to the neighborhood $\mathcal{N}$ if and only if the following two conditions are satisfied:

1. $P(x) > 0, \forall x \in F$,

2. $P(x_i|f_{s-\{i\}}) = P(x_i|f_{\mathcal{N}_i})$.

A conditional random field (CRF) [89, 91] is an MRF globally conditioned on the data D. In other words, the conditional distribution $P(x|D)$ over the labellings of the CRF is a Gibbs distribution and can be written as:

$$P(x|D) = \frac{1}{Z}\, exp(-\sum_{c\in\mathcal{C}} V_c(x)), \tag{2.10}$$

where Z is a partition function, and $V_c(x)$ is the clique potential defined on clique size c.

2.2.5 MAP-MRF equivalence

Let us revisit the MAP estimation equation of section 2.2.3. The optimal labeling is the same as maximum-a-posterior probability, and is given by:

$$x^* = \underset{x}{\operatorname{argmax}}\, p(z|x)P(x). \tag{2.11}$$

Due to MRF-Gibbs equivalence, we can write:

$$P(x) = Z^{-1} exp(\frac{-1}{T}U(x)), \tag{2.12}$$

where $U(x)$ is known as prior energy (or clique potential or MRF prior). One example of clique potential is: $U(x) = \sum_i \sum_{j\in N_i} (x_i - x_j)^2$. Note that this is not the only type of clique potential. Clique potentials are modeled according to the problem. Sometimes, they can also be learned from the training data.

Let us assume there is identical independent Gaussian distribution $N(\mu, \sigma^2)$ in our observation and actual configuration, i.e.,

$$p(z|x) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-U(z|x)}, \tag{2.13}$$

where

$$U(z|x) = \sum_{i=1}^{m} \frac{(x_i - z_i)^2}{2\sigma_i^2}. \tag{2.14}$$

Thus we can write:

$$x^* = \underset{x}{\text{argmax}}\{\prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-U(z|x)} \times Z^{-1} e^{\frac{-1}{T}U(x)}\}. \tag{2.15}$$

Taking negative logarithm on both the side we can rewrite above equation as:

$$x^* = \underset{x}{\text{argmin}}\{U(z|x) + U(x)\}. \tag{2.16}$$

Or,

$$x^* = \underset{x}{\text{argmin}}\{\sum_{i=1}^{m} \frac{(x_i - z_i)^2}{2\sigma_i^2} + U(x)\}. \tag{2.17}$$

The right hand side of the above equation is popularly known as **posterior energy**. Therefore, the problem of finding the optimal configuration becomes equivalent to minimizing energy.

In this section, we have seen how MAP estimation in a MAP-MRF framework leads to energy minimization. The two terms of the posterior energy are: (i) $\sum_{i=1}^{m}(x_i - z_i)^2$, and (ii) $U(x)$. These terms are popularly known as data and smoothness term respectively.

2.3 Classes of energy functions

In this section, we will introduce classes of energy functions. Specifically, we will define a function known as submodular set function [24, 139]. In some respect, these functions in discrete optimization are similar to convex function in continuous optimization.

Definition 1 *Let $\mathbb{N} = \{1, 2, ..., n\}$ be a set. Then a set function $f : 2^n \to \mathbb{R}$ is said to be submodular if and only if for all subsets $A, B \subset \mathbb{N}$ it satisfies:*

$$f(A \cup B) + f(A \cap B) \leq f(A) + f(B). \tag{2.18}$$

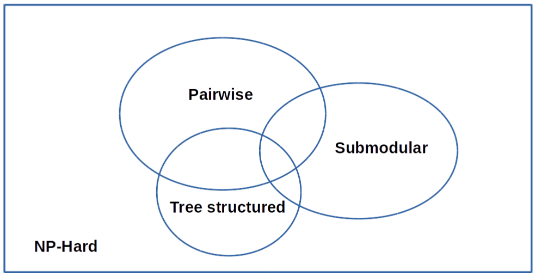


Figure 2.2 Class of energy functions: categorized based on complexity of minimization. In general, minimizing Energy function is an NP-hard problem. However, there exists certain class of functions which can be efficiently minimized.

Every subset of a set can be written in terms of a binary random vector. The entry of this random vector is 1 if corresponding element is present in the subset otherwise it is 0. For example, two sets $A = \{1\}$ and $B = \{2\}$ can be equivalently written as $X_1 = \{1, 0\}$ and $X_2 = \{0, 1\}$ respectively. In other words, we can re-write submodularity condition as follows: $f(X_1 \cup X_2) + f(X_1 \cap X_2) \leq f(X_1) + f(X_2)$ or equivalently, $f(0, 0) + f(1, 1) \leq f(0, 1) + f(1, 0)$. Many discrete optimization problems are equivalently posed as minimization of discrete functions. Minimizing a function, in general, is an NP-hard problem. However, there exist a class of energy functions for which can be minimized efficiently. One such class is submodular function.

Unfortunately, not all vision problems can be posed as problems of minimizing binary submodular functions. To get a useful approximate solution in such scenarios a number of approximate minimization algorithms have been proposed in literature. For example, move making algorithms such as α-expansion and $\alpha - \beta$-swap are two such which guarantees global solution with a constant approximation [27]. These two methods are widely used for solving multi-label optimization problems. Figure 2.2 shows the class of energy functions categorized based on the complexity of their minimization. In general minimizing an energy function is an NP-hard problem. However, if the interaction between random variables can be modeled as a tree, the corresponding energy function can be minimized exactly and in polynomial time. Similarly, a class of energy functions like submodular energy functions with the additional constraints that only pairwise interactions are allowed, can also be minimized very efficiently.

2.4 Popular energy minimization techniques

Many problems in computer vision can be elegantly expressed as Markov random fields, yet the resulting energy minimization problems have been widely viewed as intractable till the 90s. In last decade, algorithms such as graph cuts and loopy belief propagation [175] have proven to be efficient for many computer vision tasks. For example, the top performing segmentation and stereo methods are MRF based. There exists many energy minimization techniques in literature such as graph cuts, iterated conditional mode ICM algorithm [19], loopy belief propagation L-BP [175], and tree-reweighed message passing (TRW) [80]. In this section we briefly discuss graph cuts, L-BP, and TRW.

- **Graph cut.** In graph theory, a cut is a partition of the vertices of a flow graph into two disjoint subsets. There are many cuts possible in a flow graph. The cost of min-cut is same as the max-flow of the flow graph. There are many efficient algorithms for finding min-cut of a graph. The reader is encouraged to refer [34] for more details. A submodular energy function can be represented as a flow graph, and every cut of this graph corresponds to an assignment to the energy function [82]. For the pseudo binary submodular functions, global minima can be obtained using graph cut algorithm. On the other hand, energy functions which can take multiple labels, can not be exactly solved using graph cut. However, an approximate minima is obtained using move making algorithms in such cases [27].

  Minimizing submodular functions using graph cuts. Any pseudo Boolean submodular function can be represented using flow graphs. The min-cut of such graph divides the random variables into two sets and assigns 0 or 1 to each variable depending on whether the corresponding node belongs to S or T after the cut. Such assignment to the submodular function yields the global minimum of the function [82].

- **L-BP [175].** Loopy belief propagation (L-BP) is a message passing based inference algorithm. It has two variants: (i) sum-product, and (ii) max-product. The sum-product algorithm computes the marginal probability distribution of each node in the graph whereas the max-product is designed to find the solution corresponding to the lowest energy. The belief-propagation (BP) algorithm was originally designed for graphs without cycles. In such cases BP finds the global minima of the energy function. However, many vision problems leads to inference in a cyclic graph and in general, loopy BP does not guarantee convergence. However, it has been experimentally proven that L-BP produces a strong local minima.

- **TRW [80].** Tree-reweighed message passing is a variant of belief propagation. The only major difference is in the order of message passing. The TRW has following two variants:

1. TRW-T: It has three steps: (i) split the graph into trees, (ii) run BP on all the trees, and (iii) average all the nodes. It is not guaranteed to converge.
2. TRW-S: It has four steps: (i) split the graph into trees, (ii) pick a node p randomly. (iii) run BP on all the trees containing p. (iv) average node p and go to step – (ii) until convergence. TRW-S significantly outperforms TRW-T both computation and performance wise.

2.4.1 Recent advances

Most energy minimization problem in computer vision literature assume that the energy can be represented as unary and pairwise terms. This assumption severely restricts the expressiveness of these models making them poor for capturing the rich statistics of natural scenes. In the last decade, researchers have shown huge interest in modeling vision problems using higher order clique potentials [63, 78, 127, 136]. Higher order potentials have better expressiveness than pairwise potentials; however the higher order introduction to energy functions is not a trivial extension of pair-wise because, introducing higher order terms significantly reduces the efficiency of existing inference algorithms. Due to the lack of efficient algorithms for minimizing the energy functions with higher order terms, their use has been quite limited. In recent works [63,127,136], higher order clique potentials are reduced to unary and pair-wise term, and then efficiently solved. However, there exist many useful higher order potential functions for which the minimization problem does not have an efficient solution. This makes the area of the higher order potentials exciting to work on. In this thesis, we model our problem of scene text recognition in higher order framework, and decompose the higher order cost into unary and pairwise terms to efficiently solve the corresponding inference problem using TRW-S (Chapter 4).

2.5 Dynamic time warping (DTW)

In Chapter 5, we present a word matching scheme that compares synthetic and scene text images using dynamic time warping (DTW). DTW has been widely used for matching one-dimensional signals in many area such as, speech analysis, bio-informatics, and handwriting word spotting [128].

The DTW is used to compute the distance between two series [140]. A naive approach to compute the similarity between two time time series could be to uniformly sample equal number of points from them, and then compute the Euclidean distance between these points. This method does not produce the expected results, as it compares points that might not correspond well.

The DTW distance $DTW(m, n)$ between the sequences $X = \{x_1, x_2, \ldots, x_M\}$ and $Y = \{y_1, y_2, \ldots, y_N\}$ can be recursively computed using dynamic programming as:

$$DTW(i,j) = \min \begin{cases} DTW(i-1,j) + D(i,j) \\ DTW(i,j-1) + D(i,j) \\ DTW(i-1,j-1) + D(i,j), \end{cases} \tag{2.19}$$

where $D(i, j)$ is the distance between features x_i and y_j, and their choice varies from application to application. The DTW distance computation has a time complexity of $O(MN)$ to match two series of length M and N respectively.

Warping path. In context of DTW distance computation between two signal of length M and N respectively, a wrapping path is a sequence $\{p_0, p_1, \cdots p_K\}$ with $p_i = (x_i, y_i) \in [1 : M] \times [1 : N] \forall i \in 1, 2, \cdots, K$ satisfying the following three conditions.

1. Boundary condition: $p_0 = (1, 1)$ and $p_K = (M, N)$

2. Monotonicity condition: $x_1 \leq x_2 \leq \cdots x_K \; y_1 \leq y_2 \leq \cdots y_K$

3. Step size condition: $p_{i+1} - p_i \in \{(0, 1), (1, 1), (1, 0)\}$ for $i \in \{1, 2, \cdots, K-1\}$

2.6 Indexing and re-ranking

In the information retrieval community, indexing and re-ranking are two fundamental terms [102]. We provide a brief background of these two terms which are directly related to our Chapter 6 where we demonstrate text-to-image retrieval.

Indexing. The objective of storing an index is to optimize speed and performance in retrieving relevant images or video frames for a query text. An index is especially useful for demonstrating web-scale image retrieval. However, it comes with the cost of additional space and time required for an update, and often these are the traded off for the time saved during retrieval. In image search inverted index containing query text and relevant image is created. In our work, we create inverted index for fast text-to-image retrieval. This index contains a list of vocabulary words (potential queries) and their presence score in all images.

Re-ranking. For image retrieval, simple operations are performed on features of all the images of the dataset, and an initial retrieval results are obtained. Once initial results are obtained, more costly operations are performed on top-K images. In our work, we propose two types of re-ranking schemes based on spatial positioning and ordering of characters of the query word.

Chapter 3

Cropped Word Recognition: Robust Segmentation for Better Recognition

We propose a robust text segmentation (binarization) technique for addressing word recognition problem. Color and strokes are the salient features of text regions in an image. In this chapter, we use both these features as cues, and introduce a novel energy function to formulate the text binarization problem. The minimum of this energy function corresponds to the optimal binarization. We minimize the energy function using an iterative graph cut based algorithm. Our model is robust to variations in foreground and background as we learn Gaussian mixture models for color as well as strokes in each iteration of the graph cut. We show results on word images from the challenging ICDAR 2003, ICDAR 2011, and street view text datasets, and compare our performance with previously published methods. Our approach shows significant improvements in performance with respect to various performance measures commonly used to assess text binarization schemes. In addition, our algorithm is computationally efficient, and can adapt to a variety of document images such as video texts and handwritten images.

3.1 Introduction

Binarization is one of the key preprocessing steps in many document image analysis systems [152]. The performance of the subsequent steps like character segmentation and recognition is highly dependent on the success of binarization. Document image binarization has been an active area of research for many years [57, 58, 92, 106, 107, 152, 161]. *Is binarization a solved problem?* Certainly not, especially, in light of challenges posed by text in video sequences, born-digital (web and email) images, old historic manuscripts and natural scenes where the state of the art recognition performance is still poor. In this context of wide variety of imaging systems, designing a powerful text binarization algorithm can be considered a major step towards robust text understanding. The recent interest of the community

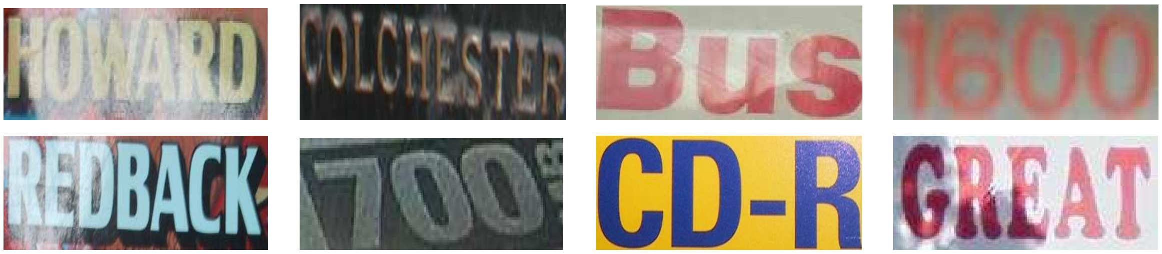


Figure 3.1 Few sample images we consider in work work. Due to large variations in foreground and background colors, most of the popular binarization techniques in the literature tend to fail on such images.

by organizing binarization contests like DIBCO [47], H-DIBCO [126] at major international document image analysis conferences further highlights its importance.

In this chapter, we focus on binarization of natural scene text images. These images contain numerous degradations which are not usually present in machine-printed ones such as, uneven lighting, blur, complex background, and perspective distortion. A few sample images from the popular datasets we use are shown in Fig. 3.1. Our proposed method also generalizes to historical handwritten document images. In fact, our method achieves significantly high text binarization performance on H-DIBCO 2012 historical handwritten image dataset [6].

Our method is inspired by the success of interactive graph cut [25] and GrabCut [134] algorithms for foreground-background segmentation of natural scenes. We formulate the binarization problem in an energy minimization framework, where text is foreground and anything else is background, and define a novel energy (cost) function such that the quality of the binarization is determined by the energy value. We minimize this energy function to find the optimal binarization using an iterative graph cut scheme. The graph cut method needs to be initialized with foreground and background seeds. To make the binarization fully automatic, we initialize the seeds by obtaining character-like strokes. At each iteration of graph cut, the seeds and the binarization are refined. This makes it more powerful than a one-shot graph cut algorithm. Moreover, we use two cues to distinguish text regions from background: (i) color, and (ii) stroke width.We model foreground and background colors as well as stroke widths in a Gaussian mixture Markov random field framework [23], to make the binarization robust to variations in foreground and background.

The contributions of this chapter are threefold: firstly, we proposed a principled framework for the text binarization problem, which is automatically initialized with character-like strokes. Use of color and stroke width cues in an optimization framework for the text binarization problem is a major novelty here.

Secondly, we present a comprehensive evaluation of the proposed binarization technique on multiple text datasets. We evaluate the performance using various measures, such as pixel-level accuracy, atom-level accuracy as well as recognition results, and compare it with the state of the art methods [58, 74, 77, 106, 116, 119, 138, 171]. To our knowledge, text binarization methods have not been evaluated in such a rigorous setting in the past, and are restricted to only few hundred images or only one category of document images (e.g., handwritten documents or scene text).

In contrast, we evaluate on more than 2000 images including scene text, video text, born-digital and handwritten images. Additionally, we also perform qualitative analysis on 6000 images containing video text of several non-European scripts. Interestingly, the performance of existing binarization methods varies widely across the datasets whereas our results are consistently compelling. In fact, our binarization improves the recognition results of an open source OCR [1] by more than 10% on various public benchmarks. Thirdly, we show the utility of our proposed method in binarizing degraded historical documents. On a benchmark dataset of handwritten images, our method achieves comparable performance to the H-DIBCO 2012 competition winner and a state of the art method [58], which is specifically tuned for handwritten images.

The remainder of this chapter is organized as follows. We discuss the related literature in Section 3.2. In Section 3.3, the binarization problem is formulated as a labeling problem, where we define an energy function such that its minimum corresponds to the target binary image. This section also briefly introduces the graph cut method. Section 3.4 explains the proposed iterative graph cut based binarization scheme. In Section 3.5, we discuss our automatic GMM initialization strategy. Section 3.6 gives details of the datasets, evaluation protocols, and performance measures used in this work. Experimental settings, results, discussions, and comparisons with various classical as well as modern binarization techniques are provided in Section 3.7, followed by the summary of this chapter.

3.2 Related work

Binarization is a highly researched area in the document image analysis community. Early methods for text binarization were mostly designed for clean scanned documents. In the context of images taken from street scenes, video sequences and historical handwritten documents, binarization poses many additional challenges. A few recent approaches aimed to address them for scene text binarization [76, 101, 106], handwritten text binarization [57, 58] and degraded printed text binarization [97]. In this section we review such literature as well as other works related to binarization (specifically text binarization), and argue for the need for better techniques.

We group text binarization approaches into three major categories: (1) classical binarization, (2) energy minimization based methods, and (3) others.

Classical binarization methods. They can be further categorized into: global approches (e.g., Otsu [119], Kittler [77]) and local (e.g., Sauvola [138], Niblack [116]). Global methods compute a binarization threshold based on global statistics of the image such as intra-class variance of text and background region, whereas local methods compute binarization threshold based on local statistics of the image such as mean and variance of pixel intensity in patches. The reader is encouraged to refer [152] for more details of these methods. Although most of these previous methods perform satisfactorily for many cases, they suffer from problems like: (i) manual tuning of parameters, (ii) high sensitivity to the choice of parameters, and (iii) failure in handling images with uneven lighting, noisy background, similar foreground-background colors.

Energy minimization based methods. Recently, energy minimization based methods have been proposed for text binarization problems [57, 58, 84, 106, 107, 121, 171]. In this framework, binarization problem is posed as an optimization problem, typically modeled using Markov random fields (MRFs). In [171], Wolf and Doermann posed binarization in an energy minimization framework, and applied simulated annealing (SA) to minimize the cost function. In [84], authors first classified a document into text region (TR), near text region (NTR) and background regions (BR), and then performed graph cut to produce the final binary image. An MRF based binarization for camera-captured document images was proposed in [121], where a thresholding based technique is used to produce an initial binary image which is refined with a graph cut scheme. The energy function used in [121] also uses stroke width as cues, and achieves good performance on printed document images. However, the method relies strongly on its first step, i.e., thresholding based binarization. Furthermore, it needs an accurate estimation of stroke width, which is not always trivial in the datasets we use (see Fig. 3.2). Unlike [121], our framework does not require an exact estimation of stroke width and proceeds with stroke as well as color initializations which are are refined over iterations.

Howe [57] used the Laplacian of the image intensity in the energy term for document binarization and improved it by devising methods for automatic tuning of parameters in [58]. These approaches were especially designed for handwritten images, and showed good performance. However, they fail to cope up with variations in scene text images, e.g., large changes in stroke width and foreground-background colors within a single image. Adopting a similar framework, Milyaev *et al.* [106] have proposed a scene text binarization technique, where they obtain an initial estimate of binarization with [116], and then use Lapalcian of image intensity to compute unary term of the energy function. Authors have shown applicability of this technique in end-to-end scene text understanding.

Other methods. Binarization has also been formulated as a text extraction problem [21, 40, 42, 48, 74]. Gatos *et al.* [48] presented a method with four steps: denoising with low-pass Wiener filter, rough estimation of text and background, using text and background estimate to compute local thresholds and post-processing to eliminate noise and preserve strokes. Epshtein *et al.* [40] presented a novel operator called the stroke width transform (SWT). The SWT operator computes the stroke width at every pixel of the input image. Then a set of heuristics are applied for text extraction. Kasar *et al.* [74] proposed a method which extracts text based on the candidate bounding boxes in a Canny edge image. Ezaki *et al.* [42] applied [119] on different image channels, and then used morphological operators as post processing. A few methods have also focused on color text binarization [76, 101]. However, they often use multiple heuristics, and can not be easily generalized. Feild and Learned-Miller [43] proposed a bilateral regression based binarization method. This method uses color clustering as a starting point to fit a regression model, and generate multiple hypotheses of text region. Histogram of gradient features [35] computed for English characters are then used to prune these hypotheses. More recently, Tian *et al.* [157] proposed a binarization technique which computes MSER on different color channels to obtain many connected components and then these connected components are pruned based on text vs non-text classifier to produce the binarization output.

In contrast to the binarization techniques in literature, we propose a method which models color as well as stroke width distributions of foreground (text) and background (non-text) using robust Gaussian mixture models, and perform an inference using an iterative graph cut algorithm to obtain clean binary images. We evaluate publicly available implementations of many of the existing methods on multiple benchmarks, and compare with them in Section 3.7.

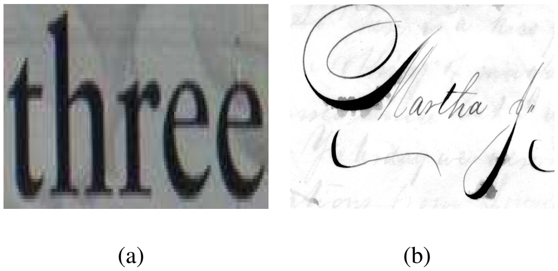

(a) (b)

Figure 3.2 (a) A scene text image from ICDAR 2003 dataset [7] (b) Part of a historical handwritten document image taken from HDIBCO 2012 dataset [6]. We observe that stroke width within text is not always constant, but varies smoothly. This motivates us to model stroke widths as GMMs.

3.3 The proposed formulation

We formulate the binarization problem in a labeling framework as follows. Binarization of an image can be expressed as a vector of binary random variables $\mathbf{X} = \{X_1, X_2, ..., X_n\}$, where each random variable X_i takes a label $x_i \in \{0, 1\}$ based on whether it is text (foreground) or non-text (background). Most of the heuristic based algorithms take the decision of assigning label 0 or 1 to x_i based on the pixel value at that location or local statistics computed in a neighborhood. Such algorithms are not effective in our case because of the variations in foreground and background color distributions.

In this work, we formulate the problem in a more principled framework where we represent image pixels as nodes in a conditional random field (CRF) and associate a unary and pairwise cost of labeling pixels. We then solve the problem by minimizing a linear combination of two energy functions E_c and E_s given by:

$$E_{all}(\mathbf{x}, \boldsymbol{\theta}, \mathbf{z}) = w_1 E_c(\mathbf{x}, \boldsymbol{\theta_c}, \boldsymbol{z_c}) + w_2 E_s(\mathbf{x}, \boldsymbol{\theta_s}, \boldsymbol{z_s}), \tag{3.1}$$

such that its minimum corresponds to the target binary image. Here $\mathbf{x} = \{x_1, x_2, ..., x_n\}$ is a set of labels at each pixel. The model parameters $\boldsymbol{\theta_c}$ and $\boldsymbol{\theta_s}$ are learned from the foreground/background color and stroke width distributions respectively. The vector $\boldsymbol{z_c}$ contains the color values, whereas the vector $\boldsymbol{z_s}$ contains pixel intensity and stroke width at every pixel. The weights w_1 and w_2 are automatically computed for the given image. To this end, we compute two image properties, namely edge density (ρ_1) and stroke width consistency (ρ_2). Edge density is defined as the fraction of edge pixels in the given image. We define stroke width consistency of the given image as standard deviation of stroke widths in that image. We observe that stroke cues are more reliable, firstly, when we have sufficient edge pixels, and secondly when standard deviation of stroke widths is low. Based on this intuition we compute the relative weights (w_1, w_2) between color and stroke terms as follows: $w_1 = |1 - \frac{1}{\rho_1 \times \rho_2}|$ $w_2 = |1 - w_1|$.

The idea here is to give more weight to the stroke width based term when the extracted strokes are more reliable, and vice-versa.

For simplicity, we will denote $\boldsymbol{\theta_c}$ and $\boldsymbol{\theta_s}$ as $\boldsymbol{\theta}$ and $\boldsymbol{z_c}$ and $\boldsymbol{z_s}$ as $\mathbf{z}$ from now. It should be noted that the formulation of stroke width based term E_s and color based term E_c are analogous. Hence, we will only show formulation of color based energy term in the subsequent texts. The color based energy term is expressed as follows:

$$E(\mathbf{x}, \boldsymbol{\theta}, \mathbf{z}) = \sum_i E_i(x_i, \boldsymbol{\theta}, z_i) + \sum_{(i,j) \in \mathbf{N}} E_{ij}(x_i, x_j, z_i, z_j), \tag{3.2}$$

where, $\mathbf{N}$ denotes the neighborhood system defined in the CRF, and E_i and E_{ij} correspond to data and smoothness terms respectively. The data term $E_i(\cdot)$ measures the degree of agreement of the inferred label x_i to the observed image data z_i. The smoothness term measures the cost of assigning labels x_i, x_j to adjacent pixels, essentially imposing spatial smoothness. A typical unary term can be expressed as:

$$E_i(x_i, \boldsymbol{\theta}, z_i) = -\log p(x_i | z_i), \tag{3.3}$$

where $p(x_i|z_i)$ is the likelihood of pixel i taking label x_i. The smoothness term is the standard Potts model [25]:

$$E_{ij}(x_i, x_j, z_i, z_j) = \lambda \frac{[x_i \neq x_j]}{dist(i,j)} \; \exp\left(\beta (z_i - z_j)^2\right), \tag{3.4}$$

where the scalar parameter λ controls the degree of smoothness, $dist(i,j)$ is the Euclidean distance between neighboring pixels i and j. Further, the smoothness term imposes the cost only for those adjacent pixels which have different labels, i.e., $[x_i \neq x_j]$. The constant β allows discontinuity-preserving smoothing, and is given by: $\beta = 1/2\mathbb{E}[(z_i - z_j)^2]$, where $\mathbb{E}[a]$ is expected value of a.

The problem of binarization is now to find the global minima of the energy function, i.e.,

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} E_{all}(\mathbf{x}, \boldsymbol{\theta}, \mathbf{z}). \tag{3.5}$$

The global minima of this energy function can be efficiently computed by graph cut [26] if it satisfies the criteria of submodularity [82]. To this end, a weighted graph $G = (V, E)$ is formed where each vertex corresponds to an image pixel, and edges link adjacent pixels. Two additional vertices source (s) and sink (t) are added to the graph. All the other vertices are connected to them with weighted edges. The weights of all the edges are defined in such a way that every cut of the graph is equivalent to some

Algorithm 1 Overall procedure of the proposed binarization scheme.

```
procedure
Input: Color or gray image
Output: Binary image
Initialize:
1. Number of GMM components ($2c_1$ and $2c_2$) for color and stroke GMMs.
2. $maxIT$: maximum number of iterations.
3. Seeds and GMMs (Section 3.5).
4. $iteration \leftarrow 1$.
CRF optimization:
    while $iteration \leq maxIT$ do
        5. Learn color and stroke GMMs from seeds
        6. Compute color ($E_c$) and stroke ($E_s$) based terms (Section 3.3)
        7. Construct $s$-$t$ graph representing the weighted energy
        8. Perform $s$-$t$ mincut
        9. Refine seeds (Section 3.5)
        10. $iteration \leftarrow iteration + 1$.
```

label assignment to the nodes. Note that the cut of the graph G is a partition of set of vertices V into two disjoint sets S and T and the cost of the cut is defined as the sum of the weights of edges going from vertices belonging to set S to T [24, 82]. The minimum cut of such a graph corresponds to the global minima of the energy function, which can be computed efficiently [26].

In [25], the set of model parameters $\boldsymbol{\theta}$ describe image foreground and background histograms. The histograms are constructed directly from the foreground and background seeds which are obtained with user interaction. However, the foreground/background distribution in our case (see images in Fig. 3.1) cannot be captured efficiently by a naive histogram distribution. Rather, we assume each pixel color (similarly stroke width) is generated from a Gaussian mixture model (GMM). In this regard, we are inspired by the success of the GrabCut [134] for object segmentation. The foreground and background GMMs in GrabCut [134] are initialized by user interaction. We aim to avoid any user interaction to make the binarization fully automatic. We achieve this by initializing GMMs with character-like strokes obtained using a method described in Section 3.5.

3.4 Iterative graph cut based binarization

Each pixel color is generated from one of the $2c_1$ Gaussian mixture models (GMMs) (c_1 GMMs each for foreground and background) with a mean μ and a covariance Σ.[1] In other words, each foreground color pixel is generated from the following distribution:

$$p(z_i|x_i, \boldsymbol{\theta}, k_i) = \mathcal{N}(\mathbf{z}, \boldsymbol{\theta}; \mu(x_i, k_i), \Sigma(x_i, k_i)), \tag{3.6}$$

where $\mathcal{N}$ denotes a Gaussian distribution, $x_i \in \{0, 1\}$ and $k_i \in \{1, ..., c_1\}$. To model the foreground color using this distribution, an additional vector $\mathbf{k} = \{k_1, k_2, ..., k_n\}$ is introduced where each k_i takes one of the c_1 GMM components. Similarly, background color is modeled from one of the c_1 GMM components. Further, the likelihood probabilities of observation can be assumed to be independent of the pixel position, thus can be expressed as:

$$p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}, \mathbf{k}) = \prod_i p(z_i|x_i, \boldsymbol{\theta}, k_i), \tag{3.7}$$

$$= \prod_i \frac{1}{\sqrt{|\Sigma_i|}} \pi_i exp\left(\frac{-\tilde{z_i}^T \Sigma_i^{-1} \tilde{z_i}}{2}\right), \tag{3.8}$$

where $\pi_i = \pi(x_i, k_i)$, $\Sigma_i = \Sigma(x_i, k_i)$ and $\tilde{z_i} = (z_i - \mu(x_i, k_i))$. Further, π_i is Gaussian mixture weighting coefficient. Due to the introduction of GMMs the data term in (3.2) becomes dependent on its assignment to GMM component, and it is given by:

$$E_i(x_i, k_i, \boldsymbol{\theta}, z_i) = -\log p(z_i|x_i, \boldsymbol{\theta}, k_i). \tag{3.9}$$

In order to make the energy function robust to low contrast color images we introduce introduce a novel term into the smoothness function which measures the “edginess" of pixels as:

$$E_{ij}(x_i, x_j, z_i, z_j) = \lambda_1 \sum_{(i,j)\in \mathbf{N}} Z_{ij} + \lambda_2 \sum_{(i,j)\in \mathbf{N}} G_{ij}, \tag{3.10}$$

where, $Z_{ij} = [x_i \neq x_j] \exp(-\beta_c||z_i - z_j||^2)$ and $G_{ij} = [x_i \neq x_j] \exp(-\beta_g||g_i - g_j||^2)$. Here g_i denotes the magnitude of gradient (edginess) at pixel i. Two neighboring pixels with similar edginess values are more likely to belong to the same class according this constraint. The constants λ_1 and λ_2 determine the

[1] Similarly, for stroke-based term it assumed that stroke width and intensity of each pixel are generated from one of the $2c_2$ GMMs.

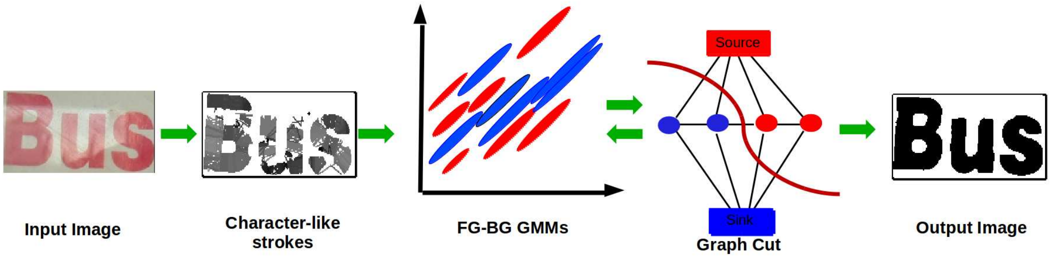


Figure 3.3 Overview of the proposed method. Given an input image containing text, we first obtain some character-like strokes using the method described in Section 3.5. GMMs for foreground (text) and background (non-text) are learnt from these initial seeds. We learn two types of GMMs: one using RGB color values and another using stroke width and intensity values. Unary costs and pairwise costs are computed for every pixel, and are appropriately weighted (see Section 3.3). An s-t graph is constructed with these. The min cut of this graph produces an initial binary image, which is used to refine the seeds, and the GMMs. The GMM refinement and graph cut steps are repeated a few times to obtain a final clean binary image. (**Best viewed in color**.)

relative strength of the color and edginess differences term with unary term respectively and are fixed to 25 empirically, and the parameters β_c and β_g are automatically computed from the image as follows.

$$\beta_c = \frac{1}{\xi} \sum_{(i,j) \in \mathbf{N}} (z_i - z_j)^2, \tag{3.11}$$

$$\beta_g = \frac{1}{\xi} \sum_{(i,j) \in \mathbf{N}} (g_i - g_j)^2, \tag{3.12}$$

where term $\xi = 2(4wh - 3w - 3h + 2)$ denotes total number of edges in the 8-neighborhood system **N** with w and h denoting the width and the height of the image respectively.

To sum up the energy formulation, both color as well as stroke width of foreground and background regions are modeled as GMMs. The GMMs for color and stroke width need to be initialized. To initialize these GMMs we obtain character-like strokes for the given image as described in Section 3.5. Once GMMs are initialized, we compute unary and pairwise terms from (3.3) and (3.10) for both color and stroke based terms. Then, at each iteration, the initializations are refined, new GMMs are learned from them, and the relative weights between color and stroke terms are recomputed. It makes the algorithm robust as it adapts to variations in foreground and background. The overview of our proposed method is illustrated in Fig. 3.3 and Algorithm 1.

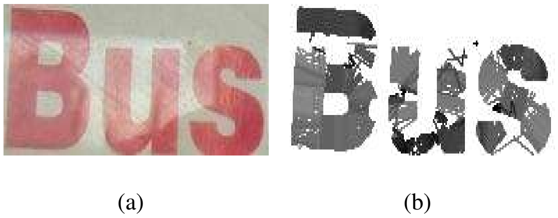


Figure 3.4 (a) Input image (b) character-like strokes obtained using the method presented in Section 3.5. The intensity values show stroke width in this image.

3.5 GMM initialization

To perform automatic binarization we need to obtain foreground and background seeds for initialization of GMMs. This can play a crucial role as it may be hard to recover from the random initialization of foreground and background seeds. In this work we propose to obtain initial seeds from character-like strokes. The idea of obtaining character-like strokes is similar in spirit to the work of Epshtein *et al.* [40]. However, unlike [40], our method is robust to incorrect strokes as we iteratively refine the initializations by learning new color and stroke GMMs in each iteration. Alternative techniques can also be used for initialization, for example other any automatic binarization technique.

Obtaining character-like strokes. We begin by extracting an edge image using Canny edge operator, and then find the character-like strokes with two-step approach:

Step 1: We first automatically detect the polarity of the image from a simple method. If the average gray pixel value of the middle strip of an image is greater than average gray pixel value of boundary, then we assign polarity of 1 (i.e., light text on dark background), otherwise we assign polarity of 0 (i.e., dark text on light background). If the polarity of the image is 1, i.e., it has light text on dark background, then we subtract π from the original gradient orientation. It should be noted that the handwritten images are always assumed as dark text on light background.

Step 2: Let u be a non-traversed edge pixel with gradient orientation θ. For every such edge pixel u, we traverse the edge image in direction of θ until we hit an edge pixel v whose gradient orientation is $(\pi - \theta) \pm \frac{\pi}{36}$, i.e., approximately the opposite gradient direction. We mark this line segment $\overline{uv}$ as a character-like stroke. We repeat this process for all the non-traversed edge pixels, and mark all the line segments as character-like strokes.

Dataset	No. of images	Type	Available annotations
ICDAR 2003 word [7]	1110	Scene text	Pixel, text
ICDAR 2011 word [8]	1189	Scene text	Pixel, text
Street view text [166]	647	Scene text	Pixel, text
CVSI 2015 [9]	6000	Video text	-
H-DIBCO 2012 [6]	14	Handwritten	Pixel

Table 3.1 Datasets. We use three scene text, a video text and a handwritten image dataset in our experiments.

We use these character-like strokes as initial foreground seeds, and pixel with no strokes are used as background seeds. Fig. 3.4 shows an example image and the corresponding character-like strokes obtained as described above. We initialize two types of GMMs: one with color values, and other with stroke width and pixel intensity values, for both foreground and background, from these initial seeds. Note that unlike our previous work [107], (i) we do not use any heuristics to discard few strokes, and keep all of them, and refine over iterations, (ii) background seeds do not need to be explicitly computed, rather, all the pixels with no strokes are initialized as probable background.

3.6 Datasets and performance measures

To conduct a comprehensive evaluation of the proposed binarization method, we use three scene text, a video text and a handwritten image datasets. These are summarized in Table 3.1. In this section, we briefly describe the datasets and provided annotations.

ICDAR cropped word datasets [7, 8]. ICDAR 2003 and 2011 robust reading dataset was originally introduced for tasks like text localization, cropped word recognition, scene character recognition. We use the cropped words these datasets for evaluating binarization performance. These datasets contain 1,110 and 1189 word images respectively. Pixel-level annotations for both these datasets are provided by Kumar *et al.* [85]. It should be noted that for ICDAR 2011 pixel-level annotations are available only for 716 images. We show pixel-level and atom-level results for only those images for this dataset, and refer this subset dataset as ICDAR 2011 (S). However, we show recognition results on all 1189 images of ICDAR 2011.

Street view text [166]. The street view text (SVT) dataset contains images harvested from Google Street View. As noted in [166], most of the images come from business signage and exhibit a high degree of variability in appearance and resolution. We show binarization results on the cropped words of SVT-word, which contains 647 word images. Pixel-level annotations for SVT is publicly available [85].

Video script identification dataset [9]. The CVSI dataset is composed of images from news videos of various Indian languages. It contains 6000 text images from 10 different scripts, namely English, Hindi, Bengali, Oriya, Gujarati, Punjabi, Kannada, Tamil, Telugu and Arabic, commonly used in India. This dataset was originally introduced for script identification [9], and does not include pixel level annotations. We use it solely for qualitative evaluation of binarization methods.

H-DIBCO 2012 [126]. Although our binarization scheme is designed for scene text images, it can also be applied for handwritten images. To demonstrate this we also test our method on H-DIBCO 2012 dataset. It contains 14 degraded handwritten images and their corresponding ground truth images with pixel-level annotations.

3.6.1 Performance measures

Although binarization is a highly researched problem, performance evaluation of binarization methods remains an ill-defined area [32]. Due to the lack of well-defined performance measures or lack of ground truth, many works in the past perform only a qualitative evaluation of binarization [72,96]. Some other works measure binarization accuracy in term of OCR performance [87]. Although improving the recognition (or OCR) performance is an end goal of binarization, unfortunately OCR systems often rely on many factors (i.e., character classification, statistical language models), and not just the quality of text binarization. Hence, OCR-level evaluation can only be considered as an indirect performance measure for evaluating binarization methods [32].

A well-established practice in document image binarization competitions at ICDAR is to evaluate binarization at pixel-level [47]. This evaluation does however have few drawbacks: (i) pixel-level ground truth for large scale datasets is difficult to acquire, (ii) due to anti-aliasing and blur, defining pixel accurate ground truth becomes subjective, (iii) a small error in ground truth can alter the ranking of binarization performance significantly as studied in [150]. Considering these drawbacks, Clavelli *et al.* [32] proposed a measure for text binarization by performing atom-level evaluation. An atom is defined as

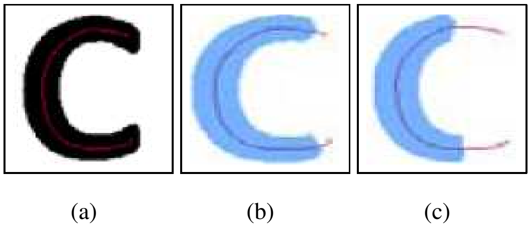


Figure 3.5 Illustration of criteria-(i) (minimal coverage in atom-level evaluation): (a) a ground truth image, (b) binary image where minimal coverage criteria is satisfied, and (c) binary image where minimal coverage criteria is not satisfied. Here, a thin line across the character ‘c’ is the skeleton of the ground truth.

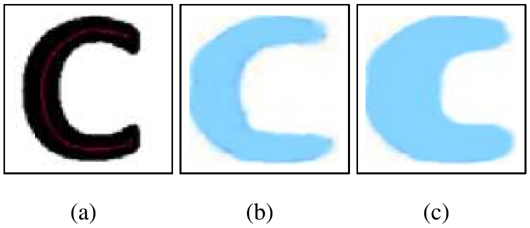
the minimum unit of text segmentation which can be recognized on its own. This performance measure does not require pixel accurate ground truths, and measures various characteristics of binarization algorithms such as producing broken texts, merging characters.

In order to provide a complete analysis we evaluate binarization methods on all three measures, namely pixel-level, atom-level and recognition (OCR) accuracy.

Pixel-level evaluation. For this evaluation given a ground truth image annotated at pixel-level and the output image of a binarization method, each pixel in the output image is classified as one of the following: (i) true positive if it is a text pixel in both the output and the ground truth image, (ii) false positive if it is a background pixel in the output image but a text pixel in the ground truth, (3) false negative if it is a text pixel in the output image but background pixel in the ground truth, or (4) true negative if it is background pixel in both the output and the ground truth image. With these in hand we compute *precision*, *recall* and *f-score* for every image, and we then report mean values of these measures over all the images in the dataset to compare binarization methods.

Atom-level evaluation. In this evaluation each connected component in the binary output image is classified as one of the six categories [32]. To determine this following two criteria are used: (i) the connected component intersects with the skeleton of ground truth with at least θ_{min}, (ii) if a connected component comprises pixels that do not overlap with text-area in the ground truth, then the distance of such component pixels are from the area edge should not exceed θ_{max}. The thresholds θ_{min} and θ_{max} are chosen as suggested by Clavelli *et al.* [32]. These two criteria are pictorially depicted in Figure 3.5

Figure 3.6 Illustration of criteria-(ii) (maximal coverage in atom-level evaluation): (a) a ground truth image, (b) binary image where maximal coverage criteria is satisfied (c) binary image where maximal coverage criteria is not satisfied.

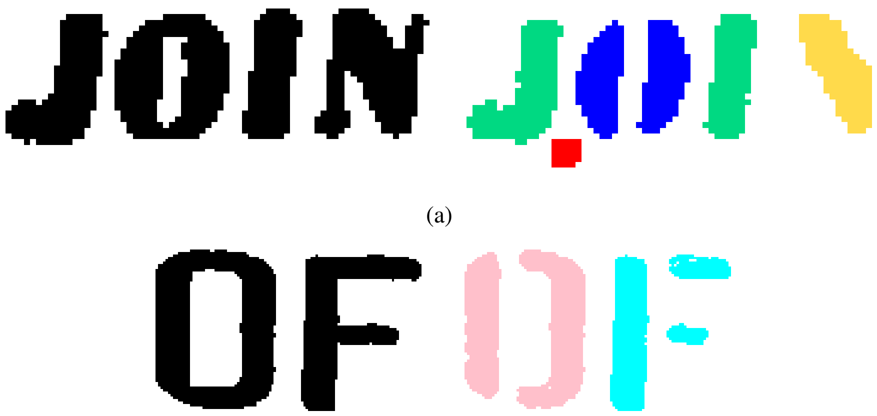


and Figure 3.6 respectively. Based on these two criteria each connected component in the output image is classified as one of the following categories.

- *whole* (w). If the connected component overlaps with skeleton of the ground truth, and both criteria are satisfied.
- *background* (b). If the connected component does not overlap with any of the skeleton of the ground truth.
- *fraction* (f). If the connected component overlaps with one skeleton of the ground truth, and only criteria-(ii) is satisfied.
- *multiple* (m). If the connected component overlaps with many skeletons of the ground truth, and only criteria-(i) is satisfied.
- *fraction and multiple* (fm). If the connected component overlaps with many skeletons of the ground truth, and only criteria-(ii) is satisfied.
- *mixed* (mi). If the connected component overlaps with many skeletons of the ground truth, and neither criteria-(i) nor criteria-(ii) is satisfied.

The number of connected components in the above categories is normalized by the number of ground truth connected components for every image to obtain scores (denoted by w, b, f, m, fm, mi). Then the mean values of these scores over the entire dataset can be used to compare binarization methods. Higher values (maximum = 1) for w whereas lower values (minimum = 0) for all the other categories are desired. We have shown examples of each of these categories in Figure 3.7. Further, to represent atom-level performance with a single measure, we compute:

$$\textit{atom-score} = \frac{1}{\frac{1}{w} + b + f + fm + mi}. \tag{3.13}$$



(a)

(b)

Figure 3.7 Two ground truth and binary image examples with six different categories of connected components with respect to atom-level evaluation. Categories are shown in following different color codes– green: *whole*, red: *background*, blue: *multiple*, yellow: *fraction*, pink: *fraction and multiple*, and cyan: *mixed*.

The *atom-score* is computed for every image, and the mean over all the images in the dataset is reported. The desired mean *atom-score* for a binarization method is 1, denoting an ideal binarization output.

OCR-level evaluation. We use two well-known off-the-shelf OCRs: Tesseract [1] and ABBYY fine Reader 8.0 [10] to evaluate binarization methods with OCR performance. Tesseract is an open source OCR whereas ABBYY fine Reader 8.0 is a commercial OCR product. We report word recognition accuracy which is defined as the number of correctly recognized words divided by the total number of words in the dataset. Following the ICDAR competition protocols [141], we do not perform any edit distance based correction with lexicons, and report case-sensitive word recognition accuracy.

3.7 Experimental analysis

Given a color or gray image containing text, our goal is to binarize it such that the pixels corresponding to the text and non-text get label 0 and 1 respectively. In this section, we perform a comprehensive evaluation of the proposed binarization scheme on the datasets introduced in Section 3.6. We compare

Method	ICDAR 2003			ICDAR 2011 (S)			Street View Text		
	Prec.	*Rec.*	*f-score*	*Prec.*	*Rec.*	*f*-score	*Prec.*	*Rec.*	*f-score*
Otsu [119]	0.86	0.90	**0.87**	0.87	0.91	**0.88**	0.64	0.83	0.70
Kittler [77]	0.75	0.89	0.78	0.79	0.89	0.80	0.55	0.81	0.62
Niblack [116]	0.68	0.87	0.74	0.75	0.86	0.79	0.52	0.78	0.60
Sauvola [138]	0.65	0.83	0.67	0.73	0.81	0.71	0.52	0.76	0.57
Wolf [171]	0.81	0.91	0.84	0.83	0.90	0.85	0.58	0.81	0.66
Kasar [74]	0.72	0.64	0.65	0.65	0.47	0.52	0.70	0.71	0.69
Milyaev [106]	0.71	0.69	0.63	0.72	0.73	0.65	0.52	0.66	0.51
Howe [58]	0.76	0.84	0.76	0.76	0.87	0.78	0.62	0.77	0.64
Bilateral [43]	0.84	0.85	0.83	0.89	0.87	0.87	0.64	0.79	0.68
Ours (color)	0.82	0.90	0.85	0.86	0.90	0.87	0.62	0.84	0.70
Ours (color+stroke)	0.82	0.91	0.86	0.86	0.91	**0.88**	0.64	0.82	**0.71**
Ours (MI)	0.92	0.95	0.93	0.96	0.98	0.97	0.87	0.95	0.90

Table 3.2 Pixel-level binarization performance. We compare binarization techniques with respect to mean *precision*, *recall* and *f-score*. Here “Ours (color)" and “Ours (color+stroke)" refer to the proposed iterative graph cut, where only the color and the color+stroke term is used respectively. “Ours (MI)" refers to proposed method with manual initialization, and denotes an upper bound

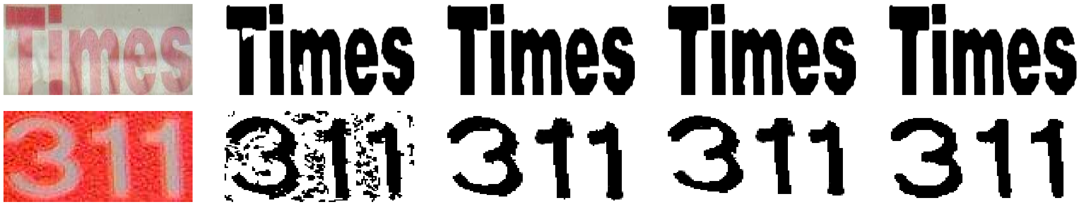


Figure 3.8 Qualitative illustration of binarization with different number of iterations. Here, we have shown original image and results with four different iterations: 1, 3, 5 and 8 (from left to right). We observe that iteration indeed helps in refining binarization output.

our method with classical as well as modern top performing text binarization techniques based on the performance measures presented in Section 3.6.1.

3.7.1 Implementation details

The proposed method is implemented in C++ and it takes about 0.8s on a cropped word image of size 60 × 180 to produce the final result on a system with 2 GB RAM and Intel$^{®}$ core$^{\text{TM}}$-2 Duo CPU

Method	ICDAR 2003						ICDAR 2011 (S)						Street View Text					
	whole	*background*	*mixed*	*fraction*	*multiple*	*atom-score*	*whole*	*background*	*mixed*	*fraction*	*multiple*	*atom-score*	*whole*	*background*	*mixed*	*fraction*	*multiple*	*atom-score*
Otsu [119]	0.69	2.97	0.06	0.24	0.02	0.59	0.73	1.94	0.04	0.21	0.03	0.63	0.42	0.75	0.08	0.10	0.06	0.34
Niblack [116]	0.50	14.70	0.17	0.74	0.02	0.23	0.57	14.77	0.12	0.85	0.02	0.31	0.35	6.19	0.15	0.20	0.03	0.16
Sauvola [138]	0.37	4.72	0.16	0.44	0.01	0.25	0.44	5.07	0.11	0.63	0.02	0.31	0.26	2.93	0.11	0.33	0.02	0.17
Kittler [77]	0.59	1.34	0.07	0.19	0.04	0.45	0.65	1.05	0.04	0.16	0.04	0.52	0.30	0.59	0.09	0.12	0.05	0.23
Wolf [171]	0.67	3.77	0.08	0.32	0.02	0.56	0.68	1.97	0.06	0.22	0.03	0.58	0.37	1.05	0.12	0.12	0.06	0.28
Kasar [74]	0.51	1.65	0.06	0.34	0.01	0.43	0.38	1.59	0.07	0.33	0.00	0.31	0.49	3.19	0.08	0.26	0.03	**0.41**
Milyaev [106]	0.36	2.44	0.11	0.37	0.02	0.30	0.37	1.04	0.11	0.30	0.03	0.30	0.27	4.87	0.09	0.18	0.03	0.24
Howe [58]	0.52	0.34	0.11	0.18	0.02	0.46	0.55	0.26	0.10	0.11	0.03	0.50	0.38	13.38	0.09	0.12	0.04	0.32
Bilateral [43]	0.62	2.21	0.08	0.38	0.02	0.52	0.69	2.40	0.04	0.34	0.02	0.60	0.40	5.35	0.09	0.21	0.04	0.31
Ours (color)	0.67	0.58	0.06	0.17	0.03	0.60	0.71	0.38	0.03	0.17	0.03	0.65	0.41	0.75	0.08	0.08	0.07	0.34
Ours (col+str)	0.68	0.49	0.06	0.15	0.03	**0.62**	0.74	0.50	0.04	0.13	0.03	**0.67**	0.40	0.33	0.08	0.07	0.07	0.34
Ours (MI)	0.77	0.20	0.02	0.13	0.03	0.72	0.86	0.26	0.01	0.09	0.02	0.80	0.64	0.17	0.03	0.09	0.07	0.60

Table 3.3 Atom-level evaluation. A connected component in the output image is classified into one of six categories. We show the fractions of connected components classified as *whole*, *background*, *mixed*, *fraction*, and *multiple*. Moreover, we also show the *atom-score*. Here “Ours (color)" and “Ours (col+str)" refer to the proposed iterative graph cut, where only color based term and color+stroke based term is used in the energy function. “Ours (MI)" refers to proposed method with manual initialization of GMMs and indicates an upper bound.

Iteration	1	2	3	4	5	6	7	8	9	10
f-score	0.85	0.86	0.88	0.89	0.89	0.89	0.90	0.90	0.90	0.90

Table 3.4 *f-score* on ICDAR 2003-validation set with different iterations

with 2.93 GHz processor system. We will make our implementation publicly available on the project website [2].

For our method we empirically chose number of GMMs as 5 (for both color and stroke based GMMs), number of graph cut iterations as 8 and $\lambda = 2$ in (3.4) for all our experiments. We used the standard public implementations of well-known binarization techniques, namely Otsu [119], Kittler [77], Niblack [116] and Sauvola [138] for comparison. Global binarization techniques Otsu [119] and Kittler [77] are parameter-independent. For local thresholding methods Niblack [116] and Sauvola [138], we choose the parameters by cross-validating on a validation set of ICDAR 2003 dataset. For contemporary methods like Kasar *et al.* [74], Bi-lateral regression [43], Howe *et al.* [58], Milyaev *et al.* [106] we use the implementations provided by the authors. Among these methods, Kasar *et al.* [74] is parameter-independent whereas for the others we use the parameter settings suggested by the corresponding authors. Further, [74] is originally designed for full scene image, and uses some heuristics on candidate character bounding box. Considering this we modify these heuristics (e.g., the maximum allowed height for a character candidate bounding box is changed from 80% of image height to 99% of image size). This modification makes the implementation suitable for cropped word images.

Polarity check. Most of the binarization methods in the literature produce white text on black background for images with light text on dark background. Since ground truth typically contains black text on white ground, hence we perform following simple automatic polarity check before evaluating the method. If the average gray pixel value of the middle strip of a given word image is greater than average gray pixel value of boundary, then we assign reverse polarity, i.e., light text on dark background, to it, and invert the corresponding output image before comparing it with the ground truth. Note that our method produces black text on white background irrespective of the polarity of the word image, and hence does not require this inversion.

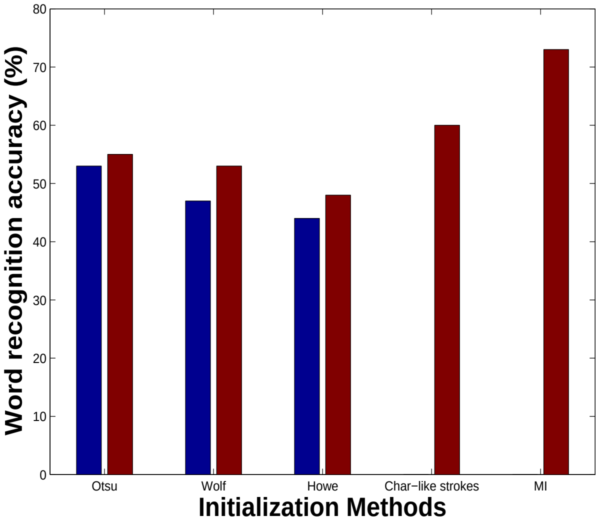
We now provide empirical evidence for choice of various parameters, such as, number of iterations, the GMM initialization method, number of GMMs and weights λ_1 and λ_2 in our method. For these studies we use the validation set of ICDAR 2003 dataset for which the pixel level annotations are provided by [105].

Number of iterations. We refine the initial strokes and color obtained by our unsupervised automatic initialization scheme (char-like strokes). This refinement is performed using iterative graph cuts. The number of iterations is a key parameter here. To illustrate the refinement of color and stroke with iteration, we conducted a study on the ICDAR 2003-validation set. We varied the iteration count from 1 to 10, and noted the pixel-level *f-score* on this dataset. This result is shown in Table 3.4. We observe that the pixel-level *f-score* improves with iterations and saturates at seven iterations. We also show qualitative examples of binarization with different number of iteration in Fig. 3.8. We observe that the iterative refinement using graph cut improves the pixel-level *f-score*, and supports our claim that color and strokes get refined with iteration. Based on this study we fix number of iterations to 8 in all our experiments.

GMM initialization. We initialize GMMs by character-like strokes (see Section 3.5). However, in our method GMMs can be initialized using any binarization method. To study the impact of initialization, we conduct following experiment. We initialize foreground and background GMMs from following best performing binarization methods in literature: Otsu [119], Wolf [171] and Howe [58], and study the word recognition performance on ICDAR 2003-validation set. We also studied the effect of user-assisted initialization of foreground and background GMMs. We call this initialization technique as manual initialization (MI). In Fig. 3.9 we show the word recognition performance of an open source OCR (Tesseract) on ICDAR 2003-validation set on following two settings: (i) when the above binarization techniques are as such used, and binarized images are fed to the OCR (blue bars), (ii) when the above mentioned binarization techniques are used for GMM initialization followed by the proposed iterative graph cut based scheme is used for binarization, and the output images are fed to the OCR (red bars). We observe that although initialization is critical to our method, our proposed binarization method improves the word recognition performance irrespective of the initialization method used. This is primarily due to the fact that our method iteratively refines the initial seeds over iterations by using color and stroke cues, and improves the binarization, and subsequently the recognition performance. Further, our proposed scheme in interactive framework (i.e., using manual initialization) achieves a very high recognition performance on this dataset. This shows that the proposed technique can also prove handy for user-assisted binarization as in [98, 99].

Other parameters. We fix parameters, such as, number of color and stroke GMMs (c), and relative weights between color and edginess terms (λ_1 and λ_2), using grid search strategy on ICDAR 2003-validation set. We vary number of color and stroke GMMs from 5 to 20 in step of 5, and compute the validation accuracy (pixel-level *f-score*). We observe only small change (± 0.02) in *f-score* for different numbers of color and stroke GMM. Based on this study we fix number of color and stroke GMMs as 5 for all our experiments. Further, we use similar strategy for choosing λ_1 and λ_2, and vary these two parameters from 5 to 50 in step of 5. We compute the pixel-level *f-score* on validation set for all these pairs, and fix a value of λ_1 and λ_2 as 25 for all our experiments based on this empirical study.

Figure 3.9 Effect of GMM initialization techniques. We show the word recognition accuracy of Tesseract for validation set of ICDAR 2003 dataset. Here, blue bars show recognition results after applying binarization techniques [58, 119, 171], and red bar shows recognition results of proposed iterative graph cut based method with these techniques used as initialization. We also show recognition results when initialization is performed from character-like strokes and manually (MI).



3.7.2 Quantitative evaluation

Pixel-level evaluation. We show these results in Table 3.2 as pixel-level mean *precision*, *recall* and *f-score* for all the images on three datasets. Values of these performance measures vary from 0 to 1, and a high value is desired for a good binarization method. We observe that the proposed scheme with only color and color+stroke based terms achieves reasonably high *f-score* on all the datasets. The classical method [119] performs better at pixel-level than many of the recent works, and is comparable to ours on ICDAR 2003 dataset and poorer on the other two datasets.

Atom-level evaluation. Recall that in this evaluation each connected component in the output image is classified as one of the following categories: *whole*, *background*, *fraction*, *multiple*, *mixed* or *fraction-multiple* (see Section 3.6.1). The fractions of these categories by various binarization methods are shown in Table 3.3. We do not show *fraction-multiple* scores as they are insignificant for all the binarization techniques. Further, we also evaluate binarization methods based on the *atom-score*. An ideal binarization algorithm should achieve 1 for the *atom-score* and whole category whereas 0 for all the categories. Note that these measures are considered more reliable than pixel-level measures [32, 106].

We observe that our method with color only and color+stroke based terms achieves reasonable *atom-score*. On ICDAR 2003 and ICDAR 2011 our method achieves rank-1 based on *atom-score* and improves by 3% and 4% respectively with respect to the next best method [119]. On SVT our method achieves rank-2. Other recent methods [43, 58, 106] although perform well on a few selected images, but fall short in comparison, when tested on multiple datasets.

OCR-level evaluation. OCR results on ICDAR 2003 and ICDAR 2011 datasets are summarized in Table 3.5. We observe that our method improves the performance of open source OCR by more than 10% on both these dataset. For example, on ICDAR 2003 dataset the open source OCR [1] (without any binarization) achieves word recognition accuracy of 47.93% whereas when our binarization is applied on these image prior to recognition we achieve 56.14%. Our binarization method with off-the-shelf OCR improves the performance over Otsu by nearly 5%. Note that all these results are based on case-sensitive evaluation, and we do not perform any edit distance based corrections. It should also be noted the aim of this work is obtain clean binary images, and evaluate binarization methods on this performance measure. Hence, we dropped recent word recognition methods which bypass binarization [67, 109, 117, 143], in this comparison.

3.7.3 Qualitative evaluation

We compare our proposed approach with other binarization methods in Fig. 3.10. Sample of images with uneven lighting, hardly distinguishable foreground/background colors, noisy foreground colors, are shown in this figure. We observe that our approach produces clearly readable binary images with lesser noise compared to [43, 106]. The global thresholding method [119] performs reasonably well for some examples but unpredictably fails in cases of high variations in text intensities (e.g., rows 2-3, 7-10). Our method is successful even in such cases and produces clean binary images.

3.7.4 Video text and handwritten images

For video text images we qualitatively evaluate binarization methods on all the images in the CVSI dataset [9]. A few selected examples of our results on this dataset are shown in Fig. 3.11. Despite very low-resolution images, performance of our method is encouraging on this dataset. Since our method uses general text cues like color and strokes, which are independent of languages, it easily generalizes to multiple languages. It should be noted that this is an encouraging step for processing Indian language scene or video text. We have also performed OCR evaluation on an Indian language Telugu. Specifically, we run Tesseract OCR trained on Telugu language on word images prior to binarization as well as after our binarization. These results are shown in Figure 3.12. The Telugu language OCR is more sensitive to

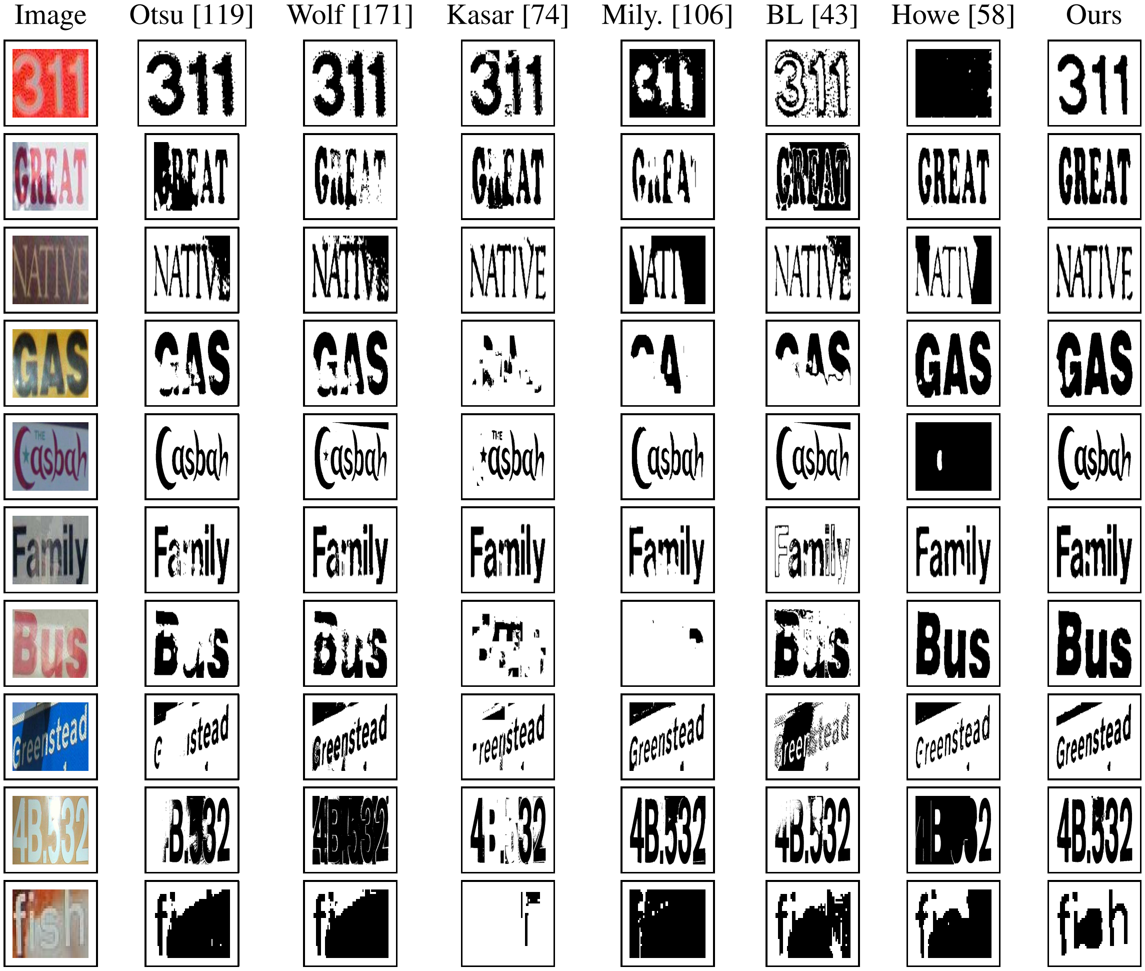


Figure 3.10 Comparison of binarization results. From left to right: input image, Otsu [119], Wolf and Doerman [171], Kasar *et al.* [74], Milyaev *et al.* [106], bilateral regression [43], Howe [58] and our method which uses color and stroke cues. Other classical techniques [77, 116, 138] are not very successful on these images.

Method	ICDAR 2003		ICDAR 2011	
	Tesseract	*ABBYY*	*Tesseract*	*ABBYY*
No Binarization*	47.93	46.51	47.94	46.00
Otsu [119]	51.71	49.10	55.92	53.99
Kittler [77]	44.55	43.25	48.84	48.61
Sauvola [138]	19.73	17.60	26.24	26.32
Niblack [116]	15.59	14.45	22.20	21.27
Kasar [74]	33.78	32.75	12.95	12.11
Wolf [171]	46.52	44.90	50.04	48.78
Milyaev [106]	22.70	21.87	22.07	22.54
Howe [58]	42.88	41.50	43.99	41.04
Bilateral [43]	50.99	47.35	45.16	43.06
Ours (color)	52.25	49.81	59.97	55.00
Ours (col+str)	**56.14**	**52.97**	**62.57**	**58.11**

Table 3.5 Word recognition accuracy (in %): open vocabulary setting. Results shown here are case sensitive, and without minimum edit distance based correction. * No binarization implies that color images are used directly to respective OCR systems.

small errors in binarization due to many similar characters and *matras*. We observe that our binarization generally has positive impact on the OCR result. However, this is a preliminary experiment towards Indian language OCR. A more comprehensive study on multiple Indian languages is an exciting future direction of our work.

We also evaluated on handwritten images of HDIBCO 2012 [6], and compare it with other methods for this task. Quantitative results on HDIBCO 2012 dataset are summarized in Table 3.6. We observe that our proposed method outperforms modern and classical binarization methods, and is comparable to the H-DIBCO 2012 competition winner [58]. Moreover, we achieve noticeable improvement by using stroke based term on this dataset, which shows the importance of stroke based terms on handwritten images. We show qualitative results for couple of examples in Fig. 3.13. We observe that despite ink bleed and high variations in pixel intensities and strokes our method produces clean binary result. The significance of stroke based terms is also clear on these examples.

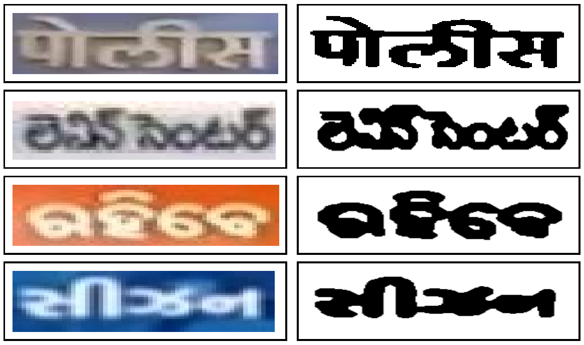


Figure 3.11 A few results on the CVSI dataset. We show results on images (left to right) with Devanagari, Telugu, Oriya and Gujarati scripts. Since our method does not use any language specific information, it is applicable to this dataset containing 10 different scripts used in India.

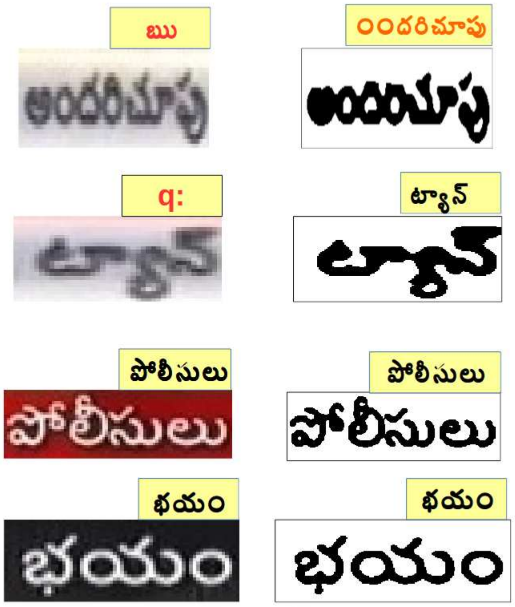


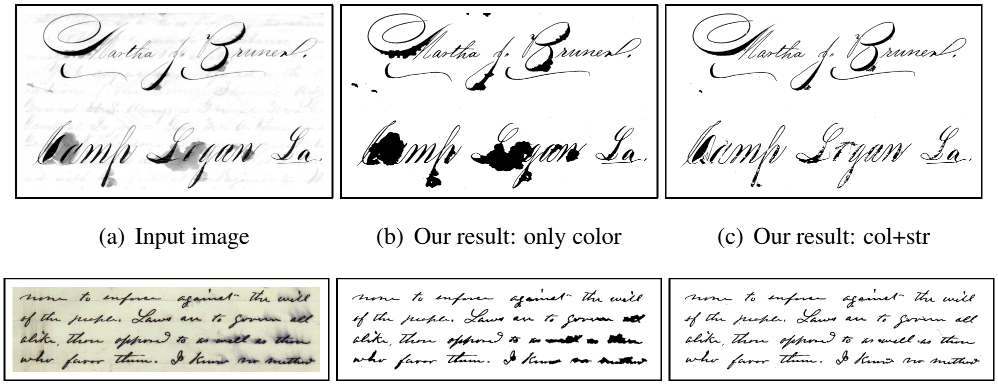
Figure 3.12 We run Tesseract OCR trained on Telugu language on word images prior to binarization as well as after our binarization. The OCR output is shown on the top-right corner of each image. The red color text indicates incorrect output. Indian languages are more senstive to small error in binarization due to similar looking characters and *matras*.

3.8 Summary

In this work we proposed a novel binarization technique, and compared it with the state of the art. Many existing methods have restricted their focus to small datasets containing only few images [43, 58, 106, 107]. They show impressive performance on these selective datasets, but this does not necessarily generalize to the larger and wide variety of datasets we consider in this work. Our method consistently performs well on all the datasets on various performance measures as we do not make any assumptions specific to images. We also compare recognition results on two public benchmarks ICDAR 2003 and ICDAR 2011, where the utility of our work is more evident. The proposed method integrated with an open source OCR [1] clearly outperforms other binarization techniques (see Table 3.5). On a dataset of video text images of multiple scripts, our results are promising, and on a benchmark dataset

Method	*f-score*
Otsu [119]	0.75
Kittler [77]	0.71
Sauvola [138]	0.14
Niblack [116]	0.19
Kasar [74]	0.74
Wolf [171]	0.78
Milyaev [106]	0.84
Howe [58]	0.89
Ours (Color)	0.84
Ours (Color+Stroke)	0.90

Table 3.6 Results on handwritten images from H-DIBCO 2012.



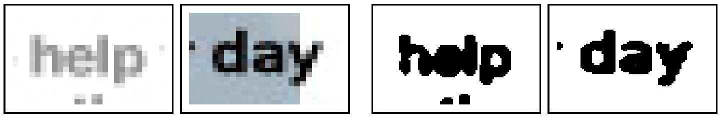
(a) Input image (b) Our result: only color (c) Our result: col+str

(d) Input image (e) Our result: only color (f) Our result: col+str

Figure 3.13 Sample images from HDIBCO 2012 dataset. (a) Input image, and results of our binarization technique: (b) with only color based term, (c) with color and stroke based terms. We observe that the color+stroke based term shows significant improvement over color only term.

of handwritten images we achieve pixel-level *precision/recall/f-score* of 0.85/0.95/0.90 which is comparable to the state of the art [58].

Comparison with other energy minimization based methods. A few binarization techniques in the literature are based on an energy minimization framework [57, 58, 106, 121, 171]. Our method also falls



(a) Input images (b) Output images

Figure 3.14 A few example where the binarization result is over-smoothed..

in this category, but differs significantly in the energy formulation and minimization technique used. We compare our method experimentally with [58, 106, 171] in Tables 3.2, 3.3 and 3.5. Two other energy minimization based methods [84, 121] were dropped for experimental comparison due to unavailability of their implementation. Our method consistently outperforms these approaches on all the datasets. The robustness of our method can be attributed to the proposed iterative graph cut based algorithm which minimizes an energy function composed of color and stroke based terms.

Further improvements. Oversmoothing, one of the limitations of our method is pronounced in the case of low resolution images where inter-character gaps and holes within characters like ‘e’, ‘a’ are only a few pixels (say 3-4 pixels). In such cases our method smooths these regions. A few such example images, where our method suffers from oversmoothing, are shown in Figure 3.14. Such limitations are not new to the optimization community, and advance techniques like cooperative graph cuts [69] can be explored in this context in the future. Moreover, a noisy automatic initialization is sometime hard to recover from. A better initialization or image enhancement technique can further improve our performance.

To sum up, we have proposed a general and principled framework for the text binarization problem. Our method can be applied to a wide variety of text images (including handwritten documents), and is computationally efficient. However, the success of this method is restricted to high contrast, roughly frontal, nearly uniform background scene text images. In the upcoming chapters, we propose more effective recognition methods which are applicable to challenging scene text images captured in the wild.

Chapter 4

Cropped Word Recognition: Integrating Top-Down and Bottom-Up Cues

Recognizing scene text is a challenging problem, even more so than the recognition of scanned documents. This problem has gained significant attention from the computer vision community in recent years, and several methods based on energy minimization frameworks and deep learning approaches have been proposed. In this chapter, we present the energy minimization framework for scene text recognition and propose a model that exploits both bottom-up and top-down cues for recognizing cropped words extracted from street images. The bottom-up cues are derived from individual character detections from an image. We build a conditional random field model on these detections to jointly model the strength of the detections and the interactions between them. These interactions are top-down cues obtained from a lexicon-based prior, i.e., language statistics. The optimal word represented by the text image is obtained by minimizing the energy function corresponding to the random field model. We evaluate our proposed algorithm extensively on a number of cropped scene text benchmark datasets, namely street view text, ICDAR 2003, 2011 and 2013 datasets, and IIIT 5K-word, and show better performance than comparable methods. We perform a rigorous analysis of all the steps in our approach and analyze the results. We also show that state of the art convolutional neural network features can be integrated in our framework to further improve the recognition performance.

4.1 Introduction

The problem of understanding scenes semantically has been one of the challenging goals in computer vision for many decades. It has gained considerable attention over the past few years, in particular, in the context of street scenes [28, 49, 90]. This problem has manifested itself in various forms, namely object detection [37,45], object recognition and segmentation [94,146]. There have also been significant

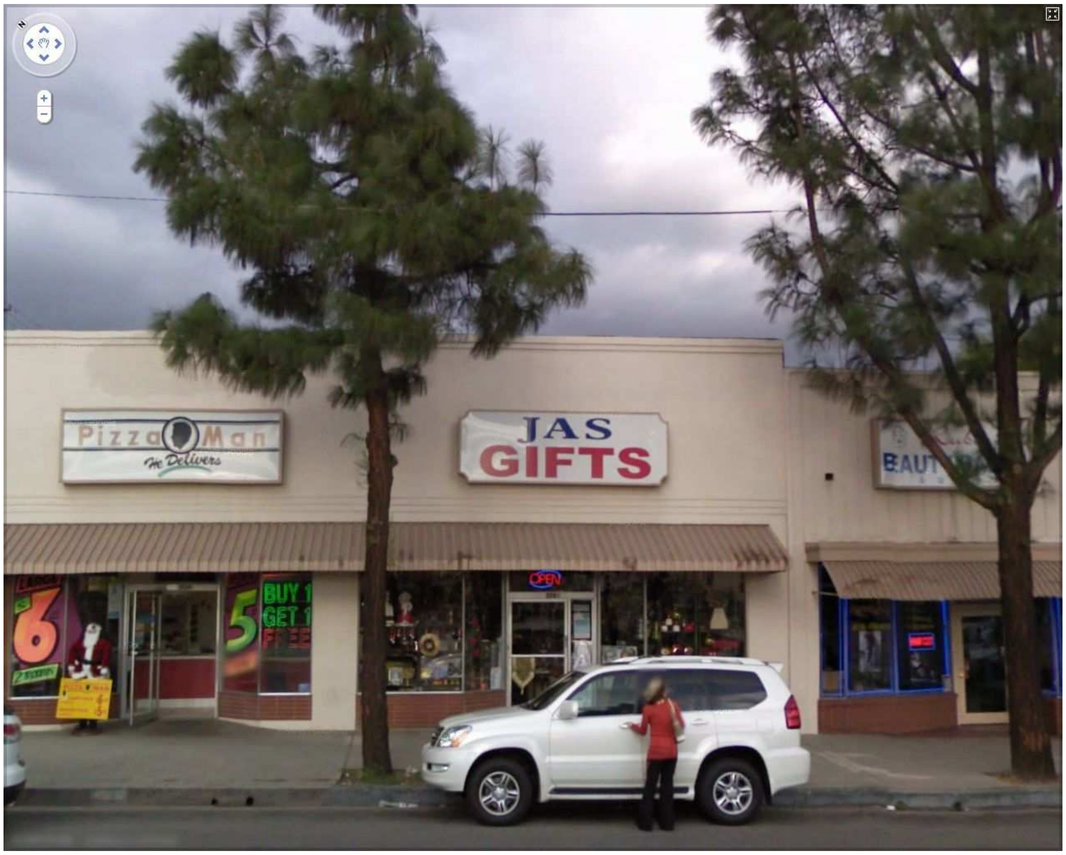


Figure 4.1 A typical street scene image taken from Google Street View. It contains very prominent sign boards with text on the building and its windows. It also contains objects such as car, person, tree, and regions such as road, sky. Many scene understanding methods recognize these objects and regions in the image successfully, but overlook the text on the sign board, which contains rich, useful information. The goal of this work is to address this gap in understanding scenes.

attempts at addressing all these tasks jointly [53, 90, 173]. Although these approaches interpret most of the scene successfully, regions containing text are overlooked. As an example, consider an image of a typical street scene taken from Google Street View in Fig. 4.1. One of the first things we notice in this scene is the sign board and the text it contains. However, popular recognition methods ignore the text, and identify other objects such as car, person, tree, and regions such as road, sky. The importance of text in images is also highlighted in the experimental study conducted by Judd *et al.* [70]. They found that viewers fixate on text when shown images containing text and other objects. This is further evidence that text recognition forms a useful component in understanding scenes.

In addition to being an important component of scene understanding, scene text recognition has many potential applications, such as image retrieval, auto navigation, scene text to speech systems, developing apps for visually impaired people [110, 113]. Our method for solving this task is inspired by the many advancements made in the object detection and recognition problems [35, 37, 45, 146]. We present a framework for recognizing text that exploits bottom-up and top-down cues. The bottom-up cues are derived from individual character detections from an image. Naturally, these windows contain true as well as false positive detections of characters. We build a conditional random field (CRF) model [91] on these detections to determine not only the true positive detections, but also the word they represent jointly. We impose top-down cues obtained from a lexicon-based prior, i.e., language statistics, on the model. In addition to disambiguating between characters, this prior also helps us in recognizing words.

The first contribution of this work is a joint framework with seamless integration of multiple cues—individual character detections and their spatial arrangements, pairwise lexicon priors, and higher-order priors—into a CRF framework which can be optimized effectively. The proposed method performs significantly better than other related energy minimization based methods for scene text recognition. Our second contribution is devising a recognition framework which is applicable not only to closed vocabulary text recognition (where a small lexicon containing the ground truth word is provided with each image), but also to a more general setting of the problem, i.e., open vocabulary scene text recognition (where the ground truth word may or may not belong to a generic large lexicon or the English dictionary). The third contribution is comprehensive experimental evaluation, in contrast to many recent works, which either consider a subset of benchmark datasets or are limited to the closed vocabulary setting. We evaluate on a number of datasets (ICDAR 2003, 2011 and 2013 [3], SVT [4], and IIIT 5K-word [108]) and show results in closed and open vocabulary settings. Additionally, we analyzed the effectiveness of individual components of the framework, the influence of parameter settings, and the use of convolutional neural network (CNN) based features [67].

The remainder of this chapter is organized as follows. Section 4.2 describes our scene text recognition model and its components. We then present the evaluation protocols and the datasets used in experimental analysis in Section 4.3. Comparison with related approaches is shown in Section 5.3, along with implementation details. We then make concluding remarks in Section 4.6.

4.2 The recognition model

We propose a conditional random field (CRF) model for recognizing words. The CRF is defined over a set of N random variables $x = \{x_i | i \in \mathcal{V}\}$, where $\mathcal{V} = \{1, 2, \ldots, N\}$. Each random variable x_i denotes a potential character in the word, and can take a label from the label set $\mathcal{L} = \{l_1, l_2, \ldots, l_k\} \cup \epsilon$, which is the set of English characters, digits and a null label ϵ to discard false character detections. The most likely word represented by the set of characters x is found by minimizing the energy function, $E : \mathcal{L}^n \rightarrow \mathbb{R}$, corresponding to the random field. The energy function E can be written as sum of potential functions:

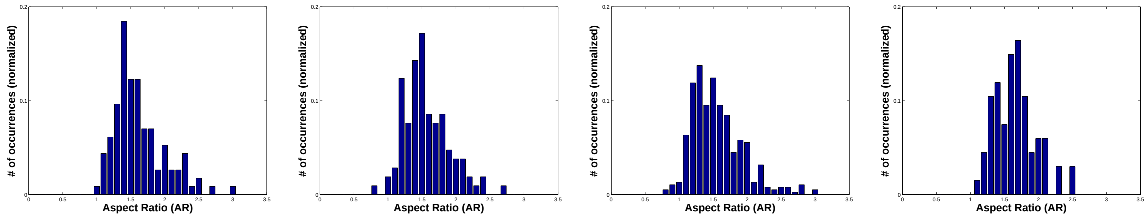
$$E(x) = \sum_{c \in \mathcal{C}} \psi_c(x_c), \tag{4.1}$$

where $\mathcal{C} \subset \mathcal{P}(\mathcal{V})$, with $\mathcal{P}(\mathcal{V})$ denoting the powerset of $\mathcal{V}$. Each x_c defines a set of random variables included in subset c, referred to as a clique. The function ψ_c defines a constraint (potential) on the corresponding clique c. We use unary, pairwise and higher order potentials in this work, and define them in Section 4.2.2. The set of potential characters is obtained by the character detection step discussed in Section 4.2.1. The neighbourhood relations among characters, modelled as pairwise and higher order potentials, are based on the spatial arrangement of characters in the word image.

4.2.1 Character detection

The first step in our approach is to detect potential locations of characters in a word image. In this work we use a sliding window based approach for detecting characters, but other methods, e.g., [172], can also be used instead.

Sliding window detection. This technique has been very successful for tasks such as, face [164] and pedestrian [35] detection, and also for recognizing handwritten words using HMM based methods [20]. Although character detection in scene images is similar to such problems, it has its unique challenges. Firstly, there is the issue of dealing with many categories (63 in all) jointly. Secondly, there is a large amount of inter-character and intra-character confusion, as illustrated in Fig. 1.2. When a window contains parts of two characters next to each other, it may have a very similar appearance to another character. In Fig. 1.2(a), the window containing parts of the characters 'o' can be confused with 'x'. Furthermore, a part of one character can have the same appearance as that of another. In Fig. 1.2(b), a part of the character 'B' can be confused with 'E'. We build a robust character classifier and adopt an additional pruning stage to overcome these issues.



(a) (b) (c) (d)

Figure 4.2 Distribution of aspect ratios of few digits and characters: (a) 0 (b) 2 (c) B (d) Y. The aspect ratios are computed on character from the IIIT-5K word training set.

The problem of classifying natural scene characters typically suffers from the lack of training data, e.g., [36] uses only 15 samples per class. It is not trivial to model the large variations in characters using only a few examples. To address this, we add more examples to the training set by applying small affine transformations [111, 147] to the original character images. We further enrich the training set by adding many non-character negative examples, i.e., from the background. With this strategy, we achieve a significant boost in character classification accuracy (see Table 4.2).

We consider windows at multiple scales and spatial locations. The location of the ith window, d_i, is given by its center and size. The set $\mathcal{K} = \{c_1, c_2, \ldots, c_k\}$, denotes label set. Note that $k = 63$ for the set of English characters, digits and a background class (null label) in our work. Let ϕ_i denote the features extracted from a window location d_i. Given the window d_i, we compute the likelihood, $p(c_j|\phi_i)$, of it taking a label c_j for all the classes in $\mathcal{K}$. In our implementation, we used explicit feature representation [162] of histogram of gradient (HOG) features [35] for ϕ_i, and the likelihoods p are (normalized) scores from a one vs rest multi-class support vector machine (SVM). Implementation details of the training procedure are provided in Section 4.4.1.

This basic sliding window detection approach produces many potential character windows, but not all of them are useful for recognizing words. We discard some of the weak detection windows using the following pruning method.

Pruning windows. For every potential character window, we compute a score based on: (i) SVM classifier confidence, and (ii) a measure of the aspect ratio of the character detected and the aspect ratio learnt for that character from training data. The intuition behind this score is that, a strong character window candidate should have a high classifier confidence score, and must fall within some range of the sizes observed in the training data. In order to define the aspect ratio measure, we observed the

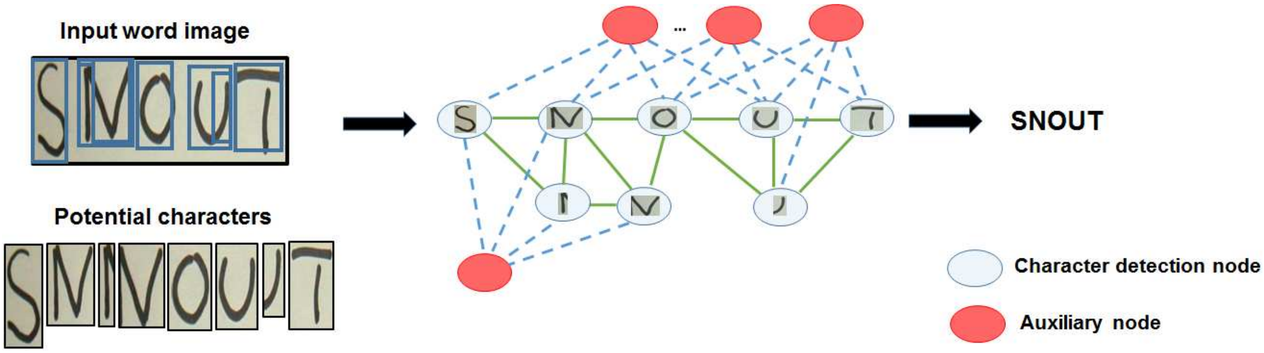


Figure 4.3 The proposed model illustrated as a graph. Given a word image (shown on the left), we evaluate character detectors and obtain potential character windows, which are then represented in a graph. These nodes are connected with edges based on their spatial positioning. Each node can take a label from the label set containing English characters, digits, and a null label (to suppress false detections). To integrate language models, i.e., n-grams, into the graph, we add auxiliary nodes (shown in red), which constrain several character windows together (sets of 4 characters in this example). Auxiliary nodes take labels from a label set containing all valid English n-grams and an additional label to enforce high cost for an invalid n-gram.

distribution of aspect ratios of characters from the IIIT-5K word training set. A few examples of these distributions are shown in Fig. 4.2. Since they follow a Gaussian distribution, we chose this score accordingly. For a window d_i with an aspect ratio a_i, let c_j denote the character with the best classifier confidence value given by S_{ij}. The mean aspect ratio for the character c_j computed from training data is denoted by μ_{a_j}. We define a goodness score (GS) for the window d_i as:

$$\text{GS}(d_i) = S_{ij} \exp\left(-\frac{(\mu_{a_j} - a_i)^2}{2\sigma_{a_j}^2}\right), \tag{4.2}$$

where σ_{a_j} is the variance of the aspect ratio for character c_j in the training data. A low goodness score indicates a weak detection, which is then removed from the set of candidate character windows.

We then apply character-specific non-maximum suppression (NMS), similar to other sliding window detection methods [45], to address the issue of multiple overlapping detections for each instance of a character. In other words, for every character class, we select detections which have a high confidence score, and do not overlap significantly with any of the other stronger detections of the same character class. We perform NMS after aspect ratio pruning to avoid wide windows with many characters suppressing weaker single character windows they overlap with. The pruning and NMS steps are performed conservatively, to discard only the obvious false detections. The remaining false positives are modelled in an energy minimization framework with language priors and other cues, as discussed below.

4.2.2 Graph construction and energy formulation

We solve the problem of minimizing the energy function (4.1) on a corresponding graph, where each random variable is represented as a node in the graph. We begin by ordering the character windows based on their horizontal location in the image, and add one node each for every window sequentially from left to right. The nodes are then connected by edges. Since it is not natural for a window on the extreme left to be strongly related to another window on the extreme right, we only connect windows which are close to each other. The intuition behind close-proximity windows is that they could represent detections of two separate characters. As we will see later, the edges are used to encode the language model as top-down cues. Such pairwise language priors alone may not be sufficient in some cases, for example, when an image-specific lexicon is unavailable. Thus, we also integrate higher order language priors in the form of n-grams computed from the English dictionary by adding an auxiliary node connecting a set of n character detection nodes.

Each (non-auxiliary) node in the graph takes one label from the label set $\mathcal{L} = \{l_1, l_2, \ldots, l_k\} \cup \epsilon$. Recall that each l_u is an English character or digit, and the null label ϵ is used to discard false windows that represent background or parts of characters. The cost associated with this label assignment is known as the unary cost. The cost for two neighbouring nodes taking labels l_u and l_v is known as the pairwise cost. This cost is computed from bigram scores of character pairs in the English dictionary or an image-specific lexicon. The auxiliary nodes in the graph take labels from the extended label set $\mathcal{L}_e$. Each element of $\mathcal{L}_e$ represents one of the n-grams present in the dictionary and an additional label to assign a constant (high) cost to all n-grams that are not in the dictionary. The proposed model is illustrated in Fig. 4.3, where we show a CRF of order four as an example. Once the graph is constructed, we compute its corresponding cost functions as follows.

4.2.2.1 Unary cost

The unary cost of a node taking a character label is determined by the SVM confidence scores. The unary term ψ_1, which denotes the cost of a node x_i taking label l_u, is defined as:

$$\psi_1(x_i = l_u) = 1 - p(l_u|x_i), \tag{4.3}$$

where $p(l_u|x_i)$ is the SVM score of character class l_u for node x_i, normalized with Platt's method [124]. The cost of x_i taking the null label ϵ is given by:

$$\psi_1(x_i = \epsilon) = \max_u p(l_u|x_i) \exp\left(-\frac{(\mu_{a_u} - a_i)^2}{\sigma^2_{a_u}}\right), \tag{4.4}$$

where a_i is the aspect ratio of the window corresponding to node x_i, μ_{a_u} and σ_{a_u} are the mean and variance of the aspect ratio respectively of the character l_u, computed from the training data. The intuition behind this cost function is that, for taking a character label, the detected window should have a high classifier confidence and its aspect ratio should agree with that of the corresponding character in the training data.

4.2.2.2 Pairwise cost

The pairwise cost of two neighbouring nodes x_i and x_j taking a pair of labels l_u and l_v respectively is determined by the cost of their joint occurrence in the dictionary. This cost ψ_2 is given by:

$$\psi_2(x_i = l_u, x_j = l_v) = \lambda_\text{l} \exp(-\beta p(l_u, l_v)), \tag{4.5}$$

where $p(l_u, l_v)$ is the score determining the likelihood of the pair l_u and l_v occurring together in the dictionary. The parameters λ_l and β are set empirically as $\lambda_\text{l} = 2$ and $\beta = 50$ in all our experiments. The score $p(l_u, l_v)$ is commonly computed from joint occurrences of characters in the lexicon [38, 39, 151, 156]. This prior is effective when the lexicon size is small, but it is less so as the lexicon increases in size. Furthermore, it fails to capture the location-specific information of pairs of characters. As a toy example, consider a lexicon with only two words CVPR and ICPR. Here, the character pair (P,R) is more likely to occur at the end of the word, but a standard bigram prior model does not incorporate this location-specific information.

To overcome the lack of location-specific information, we devise a node-specific pairwise cost by adapting [132] to the scene text recognition problem. We divide each lexicon word into T parts, where T is computed as word image width divided by average character window width. We then use only the first $1/T$th of the word for computing the pairwise cost between the initial nodes, similarly the next $1/T$th for computing the cost between the next few nodes, and so on. In other words, we do a region of interest (ROI) based search in the lexicon. The ROI is determined based on the spatial position of a detected window in the word, e.g., if two windows are on the left most side then only the first couple of characters of lexicons are considered for calculating the pairwise term between windows. This pairwise cost using the node-specific prior is given by:

$$\psi_2(x_i = l_u, x_j = l_v) = \begin{cases} 0 & \text{if } (l_u, l_v) \in \text{ROI}, \\ \lambda_\text{l} & \text{otherwise}. \end{cases} \tag{4.6}$$

We evaluated our approach with both the pairwise terms (4.5) and (4.6), and found that the node-specific prior (4.6) achieves better performance. The cost of nodes x_i and x_j taking label l_u and ϵ respectively is defined as:

$$\psi_2(x_i = l_u, x_j = \epsilon) = \lambda_\text{o} \exp(-\beta(1 - O(x_i, x_j))^2), \tag{4.7}$$

where $O(x_i, x_j)$ is the overlap fraction between windows corresponding to the nodes x_i and x_j. The pairwise cost $\psi_2(x_i = \epsilon, x_j = l_u)$ is defined similarly. The parameters are set empirically as $\lambda_\text{o} = 2$ and $\beta = 50$ in our experiments. This cost ensures that when two character windows overlap significantly, only of one them is assigned a character/digit label, in order to avoid parts of characters being labelled.

4.2.2.3 Higher order cost

Let us consider a CRF of order $n = 3$ as an example to understand this cost. An auxiliary node corresponding to every clique of size 3 is added to represent this third order cost in the graph. The higher order cost is then decomposed into unary and pairwise terms with respect to this node, similar to [136]. Each auxiliary node in the graph takes one of the labels from the extended label set $\{L_1, L_2, \ldots, L_M\} \cup L_{M+1}$, where labels $L_1 \ldots L_M$ represent all the trigrams in the dictionary. The additional label L_{M+1} denotes all those trigrams which are absent in the dictionary. The unary cost ψ_1^a for an auxiliary variable y_i taking label L_m is:

$$\psi_1^a(y_i = L_m) = \lambda_\text{a} \exp(-\beta P(L_m)), \tag{4.8}$$

where λ_a is a constant. We set $\lambda_\text{a} = 5$ empirically, in all our experiments, unless stated otherwise. The parameter β controls penalty between dictionary and non-dictionary n-grams, and is empirically set to 50. The score $P(L_m)$ denotes the likelihood of trigram L_m in the English, and is further described in Section 4.2.2.4. The pairwise cost between the auxiliary node y_i taking a label $L_m = l_u l_v l_w$ and the left-most non-auxiliary node in the clique, x_i, taking a label l_r is given by:

$$\psi_2^a(y_i = L_m, x_i = l_r) = \begin{cases} 0 & \text{if } r = u \\ 0 & \text{if } l_r = \epsilon \\ \lambda_\text{b} & \text{otherwise}, \end{cases} \tag{4.9}$$

where λ_b penalizes a disagreement between the auxiliary and non-auxiliary nodes, and is empirically set to 1. The other two pairwise terms for the second and third nodes are defined similarly. Note that when one or more x_i's take null label, the corresponding pairwise term(s) between x_i(s) and the auxiliary node are set to 0.

Example. For a word to be recognized as "OPEN" the following energy function should be the minimum.

$$\psi(O,P,E,N) = \psi_1(O) + \psi_1(P) + \psi_1(E) + \psi_1(N) + \psi_2(O,P) + \psi_2(P,E) + \psi_2(E,N) + \\ \psi_3(O,P,E) + \psi_3(P,E,N). \tag{4.10}$$

The third order terms $\psi_3(O,P,E)$ and $\psi_3(P,E,N)$ are decomposed as follows.

$$\psi_3(O,P,E) = \psi_1^a(OPE) + \psi_2^a(OPE,O) + \psi_2^a(OPE,P) + \psi_2^a(OPE,E). \tag{4.11}$$

$$\psi_3(P,E,N) = \psi_1^a(PEN) + \psi_2^a(PEN,P) + \psi_2^a(PEN,E) + \psi_2^a(PEN,N). \tag{4.12}$$

4.2.2.4 Computing language priors

We compute n-gram based priors from the lexicon (or dictionary) and then adapt standard techniques for smoothing these scores [52, 75, 156] to the open and closed vocabulary cases.

Our method uses the score denoting the likelihood of joint occurrence of pair of labels l_u and l_v represented as $P(l_u, l_v)$, triplets of labels l_u, l_v and l_w denoted by $P(l_u, l_v, l_w)$ and even higher order (e.g., fourth order). Let $C(l_u)$ denote the number of occurrences of l_u, $C(l_u, l_v)$ be the number of joint occurrences of l_u and l_v next to each other, and similarly $C(l_u, l_v, l_w)$ is the number of joint occurrences of all three labels l_u, l_v, l_w next to each other. The smoothed scores [75] $P(l_u, l_v)$ and $P(l_u, l_v, l_w)$ are now:

$$P(l_u, l_v) = \begin{cases} 0.4 & \text{if } l_u, l_v \text{ are digits}, \\ \frac{C(l_u,l_v)}{C(l_v)} & \text{if } C(l_u, l_v) > 0, \\ \alpha_{l_u} P(l_v) & \text{otherwise}, \end{cases} \tag{4.13}$$

$$P(l_u, l_v, l_w) = \begin{cases} 0.4 & \text{if } l_u, l_v, l_w \text{ are digits}, \\ \frac{C(l_u,l_v,l_w)}{C(l_v,l_w)} & \text{if } C(l_u, l_v, l_w) > 0, \\ \alpha_{l_u} P(l_v, l_w) & \text{else if } C(l_u, l_v) > 0, \\ \alpha_{l_u,l_v} P(l_w) & \text{otherwise}, \end{cases} \tag{4.14}$$

	Training Set			Test Set		
	#words	#characters	ABBYY9.0(%)	#words	#characters	ABBYY9.0(%)
Easy	658	-	44.98	734	-	44.96
Hard	1342	-	16.57	2266	-	5.00
Total	2000	9658	20.25	3000	15269	14.60

Table 4.1 Our IIIT 5K-word dataset contains a few less challenging (Easy) and many very challenging (Hard) images. To present analysis of the dataset, we manually divided the words in the training and test sets into *easy* and *hard* categories based on their visual appearance. The recognition accuracy of a state of the art commercial OCR – ABBYY9.0 – for this dataset is shown in the last column. Here we also show the total number of characters, whose annotations are also provided, in the dataset.

Image-specific lexicons (small or medium) are used in the closed vocabulary setting, while in the open vocabulary case we use a lexicon containing half a million words (henceforth referred to as large lexicon) provided by [170] to compute these scores. The parameters α_{l_u} and α_{l_u,l_v} are learnt on the large lexicon using SRILM toolbox.[1] They determine the low score values for n-grams not present in the lexicon. We assign a constant value (0.4) when the labels are digits, which do not occur in the large lexicon.

4.2.2.5 Inference

Having computed the unary, pairwise and higher order terms, we use the sequential tree-reweighted message passing (TRW-S) algorithm [80] to minimize the energy function. The TRW-S algorithm maximizes a concave lower bound of the energy. It begins by considering a set of trees from the random field, and computes probability distributions over each tree. These distributions are then used to reweight the messages being passed during loopy belief propagation [120] on each tree. The algorithm terminates when the lower bound cannot be increased further, or the maximum number of iterations has been reached.

In summary, given an image containing a word, we: (i) locate the potential characters in it with a character detection scheme, (ii) define a random field over all these potential characters, (iii) compute the language priors and integrate them into the random field model, and then (iv) infer the most likely word by minimizing the energy function corresponding to the random field.

[1] Available at: `http://www.speech.sri.com/projects/srilm/`

4.3 Datasets and evaluation protocols

Several public benchmark datasets for scene text understanding have been released in recent years. ICDAR [3] and street view text (SVT) [4] datasets are two of the initial datasets for this problem. They both contain data for text localization, cropped word recognition and isolated character recognition tasks. In this chapter we use the cropped word recognition part from these datasets. Although these datasets have served well in building interest in the scene text understanding problem, they are limited by their size of a few hundred images. To address this issue, we introduced the IIIT 5K-word dataset [108], containing a diverse set of 5000 words. Here, we provide details of all these datasets and the evaluation protocol.

SVT. The street view text (SVT) dataset contains images taken from Google Street View. As noted in [166], most of the images come from business signage and exhibit a high degree of variability in appearance and resolution. The dataset is divided into SVT-spot and SVT-word, meant for the tasks of locating and recognizing words respectively. We use the SVT-word dataset, which contains 647 word images.

Our basic unit of recognition is a character, which needs to be localized before classification. Failing to detect characters will result in poorer word recognition, making it a critical component of our framework. To quantitatively measure the accuracy of the character detection module, we created ground truth data for characters in the SVT-word dataset. This ground truth dataset contains around 4000 characters of 52 classes, and is referred to as as SVT-char, which is available for download [2].

ICDAR 2003 dataset. The ICDAR 2003 dataset was originally created for text detection, cropped character classification, cropped and full image word recognition, and other tasks in document analysis [3]. We used the part corresponding to the cropped word recognition called robust word recognition. Following the protocol of [165], we ignore words with less than two characters or with non-alphanumeric characters, which results in 859 words overall. For subsequent discussion we refer to this dataset as ICDAR(50) for the image-specific lexicon-driven case (closed vocabulary), and ICDAR 2003 when this lexicon is unavailable (open vocabulary case).

ICDAR 2011/2013 datasets. These datasets were introduced as part of the ICDAR robust reading competitions [73, 141]. They contain 1189 and 1095 word images respectively. We show case-sensitive open vocabulary results on both these datasets. Also, following the ICDAR competition evaluation protocol, we do not exclude words containing special characters (such as &, :), and report results on the entire dataset.

IIIT 5K-word dataset. The IIIT 5K-word dataset [2, 108] contains both scene text and born-digital images. Born-digital images—category of images which has gained interest in ICDAR 2011 competitions [141]—are inherently low-resolution, made for online transmission, and have a variety of font sizes and styles. This dataset is not only much larger than SVT and the ICDAR datasets, but also more challenging. All the images were harvested through Google image search. Query words like billboard, signboard, house number, house name plate, movie poster were used to collect images. The text in the images was manually annotated with bounding boxes and their corresponding ground truth words. The IIIT 5K-word dataset contains in all 1120 scene images and 5000 word images. We split it into a training set of 380 scene images and 2000 word images, and a test set of 740 scene images and 3000 word images. To analyze the difficulty of the IIIT 5K-word dataset, we manually divided the words in the training and test sets into *easy* and *hard* categories based on their visual appearance. Table 4.1 shows these splits in detail. We observe that a commercial OCR performs poorly on both the train and test splits. Furthermore, to evaluate components like character detection and recognition, we also provide annotated character bounding boxes. It should be noted that around 22% of the words in this dataset are not in the English dictionary, e.g., proper nouns, house numbers, alphanumeric words. This makes this dataset suitable for open vocabulary cropped word recognition. We show an analysis of dictionary and non-dictionary words in Table 4.6.

Evaluation protocol. We evaluate the word recognition accuracy in two settings: closed and open vocabulary. Following previous work [108, 143, 165], we evaluate case-insensitive word recognition on SVT, ICDAR 2003, IIIT 5K-word, and case-sensitive word recognition on ICDAR 2011 and ICDAR 2013. For the closed vocabulary recognition case, we perform a minimum edit distance correction, since the ground truth word belongs to the image-specific lexicon. On the other hand, in the case of open vocabulary recognition, where the ground truth word may or may not belong to the large lexicon, we do not perform edit distance based correction. We perform many of our analyses on the IIIT 5K-word dataset, unless otherwise stated, since it is the largest dataset for this task, and also comes with character bounding box annotations.

Method	SVT	ICDAR	c74K	IIIT 5K	Time
Exempler SVM [142]	-	71	-	-	-
Elagouni *et al.* [38]	-	70	-	-	-
Coates *et al.* [33]	-	**82**	-	-	-
FERNS [165]	-	52	47	-	-
RBF [109]	62	62	64	61	3ms
MKL+RBF [36]	-	-	57	-	11ms
H-36+AT+Linear	**69**	73	**68**	**66**	2ms
H-31+AT+Linear	64	73	67	63	1.8ms
H-13+AT+Linear	65	72	66	64	**0.8ms**
H-36+AT+Linear (CI)	75	77	79	75	0.8ms
CNN feat+classifier [67] (CI)	83	86	$*$	85	1ms

Table 4.2 Character classification accuracy (in %). A smart choice of features, training examples and classifier is key to improving character classification. We enrich the training set by including many affine transformed (AT) versions of the original training data from ICDAR and Chars74K (c74k). The three variants of our approach (H-13, H-31 and H-36) show noticeable improvement over several methods. The character classification results shown here are case sensitive (all rows except the last two). It is to be noted that [36] only uses 15 training samples per class. The last two rows show a case insensitive (CI) evaluation. $*$We do not evaluate the convolutional neural network classifier in [67] (CNN feat+classifier) on the c74K dataset, since the entire dataset was used to train the network.

4.4 Experiments

Given an image region containing text, cropped from a street scene, our task is to recognize the word it contains. In the process, we develop several components (such as a character recognizer) and also evaluate them to justify our choices. The proposed method is evaluated in two settings, namely, closed vocabulary (with an image-specific lexicon) and open vocabulary (using an English dictionary for the language model). We compare our results with the best-performing recent methods for these two cases.

4.4.1 Character classifier

We use the training sets of ICDAR 2003 character [3] and Chars74K [36] datasets to train the character classifiers. This training set is augmented with 48×48 patches harvested from scene images, with buildings, sky, road and cars, which do not contain text, as additional negative training examples. We then apply affine transformations to all the character images, resize them to 48×48, and compute HOG features. Three variations (13, 31 and 36-dimensional) of HOG were analyzed (see Table 4.2). We then use an explicit feature map [162] and the χ^2 kernel to learn the SVM classifier. The SVM parameters are estimated by cross-validating on a validation set. The explicit feature map not only allows a significant reduction in classification time, compared to non-linear kernels like RBF, but also achieves a good performance.

The two main differences from our previous work [109] in the design of the character classifier are: (i) enriching the training set, and (ii) using an explicit feature map and a linear kernel (instead of RBF). Table 4.2 compares our character classification performance with [33, 36, 38, 109, 142, 165] on several test sets. Note that we achieve at least 4% improvement over our previous work (RBF [109]) on all the datasets, and also perform better than [36, 165]. We are also comparable to a few other recent methods [38,142], which show a limited evaluation on the ICDAR 2003 dataset. Following an evaluation insensitive to case (as done in a few benchmarks, e.g., [67, 143], we obtain 77% on ICDAR 2003, 75% on SVT-char, 79% on Chars74K, and 75% on IIIT 5K-word. It should be noted that feature learning methods based on convolutional neural networks, e.g., [33, 67], show an excellent performance. This inspired us to integrate them into our framework. We used publicly available features [67]. This will be further discussed in Section 4.4.3. We could not compare with other related recent methods [22, 168] since they did not report isolated character classification accuracy.

In terms of computation time, linear SVMs trained with HOG-13 features outperform others, but since our main focus is on word recognition performance, we use the most accurate combination, i.e., linear SVMs with HOG-36. We observed that this smart selection of training data and features not only improves character recognition accuracy but also improves the second and third best predictions for characters.

4.4.2 Character detection

Sliding window based character detection is an important component of our framework, since our random field model is defined on these detections. At every possible location of the sliding window, we evaluate a character classifier. This provides the likelihood of the window containing the respective character. We pruned some of the windows based on their aspect ratio, and then used the goodness measure (4.2) to discard the windows with a score less than 0.1 (refer Section 4.2.1). Character-specific NMS is done on the remaining windows with an overlap threshold of 40%, i.e., if two detections have more than 40% overlap and represent the same character class, we suppress the weaker detection. We evaluated the character detection results with the intersection over union measure and a threshold of 50%, following ICDAR 2003 [3] and PASCAL-VOC [41] evaluation protocol. Our sliding window approach achieves recall of 80% on the IIIT 5K-word dataset, significantly better than using a binarization scheme for detecting characters (see Table 4.7 and Section 4.4.4).

4.4.3 Word Recognition

Closed vocabulary recognition The results of the proposed CRF model in closed vocabulary setting are presented in Table 4.3. We compare our method with many recent works for this task. To compute the language priors we use lexicons provided by authors of [165] for SVT and ICDAR(50). The image-specific lexicon for every word in the IIIT 5K-word dataset was developed following the method described in [165]. These lexicons contain the ground truth word and a set of distractors obtained from randomly chosen words (from all the ground truth words in the dataset). We used a CRF with higher order term (n=4), and similar to other approaches, applied edit distance based correction after inference. The constant λ_a in (4.8) to 1, given the small size of the lexicon.

The gain in accuracy over our previous work [109], seen in Table 4.3, can be attributed to the higher order CRF and an improved character classifier. The character classifier uses: (i) enriched training data, and (ii) an explicit feature map, to achieve about 5% gain (see Section 4.4.1 for details). Other methods, in particular, our previous work on holistic word recognition [51], label embedding [133] achieve a reasonably good performance, but are restricted to the closed vocabulary setting, and their extension to more general settings, such as the open vocabulary case, is unclear. Methods published since our original work [109], such as [143, 168], also perform well. Very recently, methods based on convolutional neural networks [22, 67] have shown very impressive results for this problem. It should

Method	**ICDAR 2003 (50)**	**SVT**	**IIIT-5K (small)**
Baselines			
ABBYY	56.04	35.00	24.50
(CSER+tesseract) [61]	57.27	37.71	33.07
Novikova *et al.* [117]	82.80	72.90	-
Our Holistic recognition [51]	89.69	77.28	75.00
Rodriguez & Perronnin [133]	-	-	76.10
Deep learning approaches			
Wang *et al.* [167]	90.00	70.00	-
Deep features [67]	**96.20**	86.10	-
PhotoOCR [22]	-	**90.39**	-
Other energy min. approaches			
PLEX [165]	72.00	57.00	-
Shi *et al.* [143]	87.04	73.51	-
Weinman *et al.* [168]	-	78.05	-
Our variants:			
Pairwise CRF [109]	81.74	73.26	66.13
Higher order [This work, HOG]	84.07	75.27	71.80
Higher order [This work, CNN]	88.02	78.21	**78.07**

Table 4.3 Word recognition accuracy (in %): closed vocabulary setting. We present results of our proposed higher order model ("This work") with HOG as well as CNN features. See text for details.

be noted that such methods are typically trained on much larger datasets, for example, 10M compared to 0.1M typically used in state of the art methods, which are not publicly available [22]. Inspired by these successes, we use a CNN classifier [67] to recognize characters, instead of our SVM classifier based on HOG features (see Sec. 4.2.1). We show results with this CNN classifier on SVT, ICDAR 2003 and IIIT-5K word datasets in Table 4.3 and observe significant improvement in accuracy, showing its complementary nature to our energy based method.

Open vocabulary recognition. In this setting we use a lexicon of 0.5 million words from [170] instead of image-specific lexicons to compute the language priors. Many character pairs are equally likely in such a large lexicon, thereby rendering pairwise priors is less effective than in the case of a small lexicon. We use priors of order four to address this (see also analysis on the CRF order in Section 4.4.4). Results on various datasets in this setting are shown in Table 4.4. We compare our method with recent work by Feild and Miller [44] on the ICDAR 2003 dataset, where our method with HOG features shows a comparable performance. Note that [44] additionally uses web-based corrections, unlike our method, where the results are obtained directly by performing inference on the higher order CRF model. On the ICDAR 2011 and 2013 datasets we compare our method with the top performers from the respective competitions. Our method outperforms the ICDAR 2011 robust reading competition winner (TH-OCR method) method by 17%. This performance is also better than a recently published work from 2014 by Weinman *et al.* [168]. On the ICDAR 2013 dataset, the proposed higher order model is significantly better than the baseline and is in the top-5 performers among the competition entries. The winner of this competition (PhotoOCR) uses a large proprietary training dataset, which is unavailable publicly, making it infeasible to do a fair comparison. Other methods (NESP [87], MAPS [86], PLT [88]) use many preprocessing techniques, followed by off-the-self OCR. Such preprocessing techniques are highly dataset dependent and may not generalize easily to all the challenging datasets we use. Despite the lack of these preprocessing steps, our method shows a comparable performance. On the IIIT 5K-word dataset, which is large (three times the size of ICDAR 2013 dataset) and challenging, the only published result to our knowledge is Strokelets [172] from CVPR 2014. Our method performs 7% better than Strokelets. Using CNN features instead of HOG further improves our word recognition accuracy, as shown in Table 4.4.

To sum up, our proposed method performs well consistently on several popular scene text datasets. Fig. 4.5 shows the qualitative performance of the proposed method on a few sample images. The higher order CRF outperforms the unary and pairwise CRFs. This is intuitive due to the better expressiveness of the higher order potentials. One of the failure cases is shown in the last row in Fig. 4.5, where the higher order potential is computed from a lexicon which does not have sufficient examples to handle alphanumeric words.

4.4.4 Further analysis

Lexicon size. The size of the lexicon plays an important role in the word recognition performance. With a small-size lexicon, we obtain strong language priors which help overcome inaccurate character detection and recognition in the closed vocabulary setting. A small lexicon provides much stronger priors than the large lexicon in this case, as the performance degrades with increase in the lexicon size.

Method	ICDAR 2003	ICDAR 2011	IIIT-5K (large)
Baselines			
ABBYY	46.51	46.00	14.60
(CSER+tesseract) [61]	50.99	51.98	25.00
Feild and Miller [44]	62.76	48.86	-
Weinman *et al.* [168]	-	57.70	-
ICDAR'11 competition [141]			
TH-OCR System	-	41.20	-
KAIST AIPR System	-	35.60	-
Neumann's Method	-	33.11	-
Stroklets [172]	-	-	38.30
Our variants			
Pairwise [109]	50.99	48.11	32.00
Higher order [This work, HOG]	63.02	**58.03**	44.50
Higher order [This work, CNN]	**67.67**	-	**46.73**

Table 4.4 Word recognition accuracy (in %): open vocabulary setting. The results of our proposed higher order model ("This work") with HOG as well as CNN features are presented here. Since the network used here to compute CNN features, i.e. [67], is learnt on data from several sources (e.g., ICDAR 2011), we evaluated with CNN features only on ICDAR 2003 and IIIT-5K word datasets, as recommended by the authors. Note that we also compare with top performers (as given in [73, 141]) in the ICDAR 2011 and 2013 robust reading competitions. We follow standard protocols for evaluation – case sensitive on ICDAR 2011 and 2013 and case insensitive on ICDAR 2003 and IIIT 5K-Word.

We show this behaviour on the IIIT 5K-word dataset in Table 4.5 with small (50), medium (1000) and large (0.5 million) lexicons. We also compare our results with a state of the art methods [133, 172]. We observe that [133, 172] shows better recognition performance with the small lexicon, when we use HOG features, but as the size of the lexicon increases, our method outperforms [133].

Binarization based methods. We investigated alternatives to sliding window character detection. To this end, we replaced our detection module with a binarization based character extraction scheme, in particular, a traditional binarization technique [119] and a more recent random field based approach presented in Chapter 3. A connected component analysis was performed on the binarized images to

Method	S	M	L
Rodriguez & Perronnin [133]	76.10	57.50	-
Strokelets [172]	**80.20**	69.30	38.30
Higher order [This work, HOG]	71.80	62.17	44.50
Higher order [This work, CNN]	78.07	**70.13**	**46.73**

Table 4.5 Studying the influence of the lexicon size – small (S), medium (M), large (L) – on the IIIT 5K-word dataset in the closed vocabulary setting.

obtain a set of potential character locations. We then defined the CRF on these characters and performed inference to get the text contained in the image. These results are summarized in Table 4.7. We observe that binarization based methods perform poorly compared to our model using a sliding window detector, both in terms of character-level recall and word recognition. They fail in extracting characters in the presence of noise, blur or large foreground-background variations. These results further justify our choice of sliding window based character detection.

CRF order. We varied the order of the CRF from two to six and obtained accuracy of 32%, 43%, 45%, 43%, 42% respectively on the IIIT 5K-word dataset in the open vocabulary setting. Increasing the CRF order beyond four forces a recognized word to be one from the dictionary, which leads to poor recognition performance for non-dictionary words, and thus deteriorates the overall accuracy. Empirically, the fourth order prior shows the best performance.

Effect of pruning. We propose a pruning step to discard candidates based on a combination of character-specific aspect ratio and classification scores (4.2), instead of simply using extreme aspect ratio to discard character candidates. This pruning helps in removing many false positive windows, and thus improves recognition performance. We conducted an experiment to study the effect of pruning on the IIIT-5K dataset in the open vocabulary setting, and observed a gain of 4.23% (46.73% vs 42.50%) due to pruning.

Limits of statistical language models. Statistical language models have been very useful in improving traditional OCR performance, but they are indeed limited [83, 151]. For instance, using a large weight for language prior potentials may bias the recognition towards the closest dictionary word. This is especially true when the character recognition part of the pipeline is weak. We study such impact

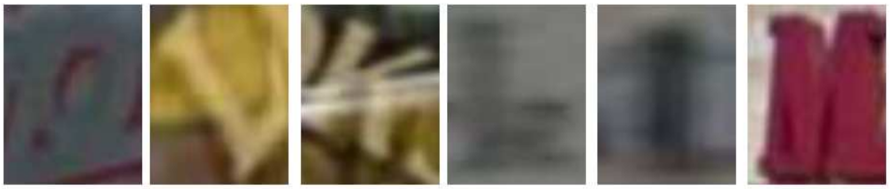


Figure 4.4 A few challenging character examples we missed in the sliding window stage. These examples are very difficult even for a human. We observed that all these potential character windows were missed due to poor SVM scores.

of language models in this experiment. Our analysis on the IIIT 5K-word dataset suggests that many of the non-dictionary words are composed of valid English n-grams (see Table 4.6). However, there are few exceptions, e.g., words like 35KM, 21P, which are composed of digits and characters; see last row of Fig. 4.5. Using language models has an adverse effect on the recognition performance in such cases. This results in inferior recognition performance on non-dictionary words as compared to dictionary words, e.g. on IIIT-5K dataset our method achieves 51% and 24% word recognition accuracy on dictionary and non-dictionary words respectively.

Limits of sliding window technique. The sliding window based character detection is successful in dealing with many challenging cases. However, it has certain limitations, e.g., localizing characters in extremely low contrast images or characters with arbitrary orientations, where sliding window based detection gives very small detection scores. Few such examples where our method fails to localize the characters are shown in Figure 4.4.

4.5 Scalability, advantages and limitations of CRF

In our higher order CRF framework, we compute the pairwise and higher order joint probabilities with external data (i.e., the English dictionary containing 0.5 million words) and perform Katz smoothing [75] on these joint probabilities to obtain pair wise and higher order potentials. This smoothing helps us to estimate the probability distribution of n-grams in the English dictionary. For inference, we use TRW-S [81] which is an efficient variant of BP, and more importantly it has convergence guarantees. The computation complexity of this inference technique depends on number of nodes in the CRF graph

	IIIT 5K train	IIIT 5K test
Non-dict. words	23.65	22.03
Digits	11.05	7.97
Dict. 3-grams	90.27	88.05
Dict. 4-grams	81.40	79.27
Dict. 5-grams	68.92	62.48

Table 4.6 Analysis of the IIIT 5K-word dataset. We show the percentage of non-dictionary words (Non-dict.), including digits, and the percentage of words containing only digits (Digits) in the first two rows. We also show the percentage of words that are composed from valid English trigrams (Dict. 3-grams), four-grams (Dict. 4-grams) and five-grams (Dict. 5-grams) in the last three rows. These statistics are computed using the large lexicon.

Char. method	C. recall	Unary	Pairwise	H. order
Otsu [119]	56	17.07	20.20	24.87
MRF model (Chapter 3)	62	20.10	22.97	28.03
Sliding window	**80**	**25.83**	**32.00**	**44.50**

Table 4.7 Character recall (C. recall) and recognition accuracy, with unary only (Unary), unary and pairwise (Pairwise) and the full higher order (H. order) models, (all in %), on the IIIT 5K-word dataset with various character extraction schemes (Char. method). See text for details.

(potential characters in our case) and number of labels (valid English n-grams in our case), and it does not change with the vocabulary size. Reader is encouraged to refer [81] for theoretical details regarding computational complexity and convergence proof of TRW-S method. The average time required for our word recognition method is 6 second on a system with 8 GB RAM and Core-2 duo processor.

Advantages and deep connection. Conditional random fields (CRFs) [91] offer statistical advantages over generative models and have already proven superior to HMMs in sequence labeling tasks [125]. Moreover, it also has several advantages for structure prediction tasks [155], such as learning from external data and learning the structure of the problem. The CRFs also mingle well with the deep learning frameworks. Deep learning based methods have improved state of the art in many computer vision tasks in last few years. The CRFs are a way of combining the advantages of discriminative

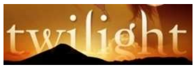
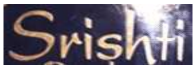
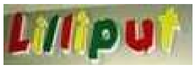
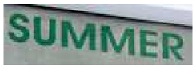
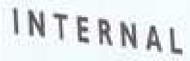
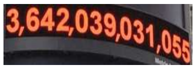
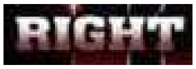
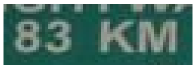
Test Image	Unary	Pairwise	Higher order(=4)
	TWI1IOHT	TWILIOHT	TWILIGHT
	SRISNTI	SRISNTI	SRISHTI
	LIIIIPUT	LIIIIPUT	LILLIPUT
	EUMMER	EUMMER	SUMMER
	IDTERNAL	IDTERNAL	INTERNAL
	364203903105S	3642039031055	3642039031055
	REGHT	REGHT	RIGHT
	83KM	BOKM	BOOM

Figure 4.5 Results of our higher order model on a few sample images. Characters in red represent incorrect recognition. The unary term alone, based on the SVM classifier, yields poor accuracy, and adding pairwise terms to it improves this. Due to their limited expressiveness, they do not correct all the errors. Higher order potentials capture larger context from the English language, and help address this issue. Note that our method also deals with non-dictionary words (e.g., second row) and non-horizontal text (sixth row). A typical failure case containing alphanumeric words is shown in the last row. (**Best viewed in colour**).

classification and graphical models. In our CRF framework, we have used CNN for obtaining character classification, and obtain a significant gain in word recognition accuracy. There are few recent works, such as [178], where authors proposed an approach which combines the strengths of both CNN and CRF based graphical model in a unified framework. We believe such unified framework can be explored for further enhancement of our proposed method in the future.

Limitations. Despite many advantages, CRFs also have some limitations, e.g., (i) high computation complexity of inference techniques, (ii) priors used in CRF framework often show strong influence in the final outcome. In our case, these influence causes a bias towards English dictionary words which results in lower success rates in recognizing non-dictionary words using our higher order CRF method.

4.6 Summary

This chapter proposes an effective method to recognize scene text. Our model combines bottom-up cues from character detections and top-down cues from lexicon. We jointly infer the location of true characters and the word they represent as a whole. We evaluated our method extensively on several challenging street scene text datasets, namely SVT, ICDAR 2003/2011/2013, and IIIT 5K-word and showed that our approach significantly advances the energy minimization based approach for scene text recognition. In addition to presenting the word recognition results, we analyzed the different components of our pipeline, presenting their pros and cons. Finally, we showed that the energy minimization framework is complementary to the resurgence of convolutional neural network based techniques, which can help build better scene understanding systems.

Chapter 5

Cropped Word Recognition: Holistic View

This chapter presents a holistic view of recognizing scene text. Previous methods addressed this problem by first detecting individual characters, and then forming them into words. Such approaches often suffer from weak character detections, due to large intra-class variations, even more so than characters from scanned documents. We take a different view of the problem and present a holistic word recognition framework in this chapter. In this, we first represent the scene text image and synthetic images generated from lexicon words using gradient-based features. We then recognize the text in the image by matching the scene and synthetic image features with our novel weighted dynamic time warping (wDTW) approach. The proposed holistic word recognition approach can be applied for recognition of non-European scene texts where the literature has not progressed much so far. However, we restrict to English languages for our experimental analysis in this chapter.

5.1 Introduction

The document image analysis community has shown a huge interest in the problem of scene text understanding in recent years [33, 122, 167]. This problem involves various sub-tasks, such as text detection, isolated character recognition, word recognition. Due to recent works [31, 40, 114], text detection accuracies have significantly improved. However, the success of methods for recognizing words still leaves a lot to be desired. We aim to address this issue in this chapter.

The problem of recognizing words has been looked at in two broads contexts – with and without the use of a lexicon [108, 109, 165, 170]. In the case of lexicon-driven word recognition, a list of words is available for every scene text image. The problem of recognizing the word now reduces to that of finding the best match from the list. This is relevant in many applications, such as: (1) recognizing certain text in a grocery store, where a list of grocery items can serve as a lexicon, (2) robotic vision in an indoor/outdoor environment.

Lexicon-driven scene text recognition may appear to be an easy task, but the best methods up until now have only achieved accuracies in the low 70s on this problem. Some of these recent methods can be summarized as follows. In [165], each word in the lexicon is matched to the detected set of character windows, and the one with the highest score is reported as the predicted word. This strongly top-down approach is prone to errors when characters are missed or detected with low confidence. In our earlier work [109], we improved upon on this model by introducing a framework, which uses top-down as well as bottom-up cues. Rather than pre-selecting a set of character detections, we defined a global model that incorporates language priors (top-down) and all potential characters (bottom-up). In [167], Wang *et al.* combined unsupervised feature learning and multi-layer neural networks for scene text detection and recognition. While both these recent methods improved the previous art significantly, they suffer from the following drawbacks: (i) The need for language-specific character training data. (ii) Do not use the entire visual appearance of the word. (iii) Prone to errors due to false or weak character detections.

In this chapter, we present an alternative path and propose a holistic word recognition method for scene text images. We address the problem in a *recognition by retrieval* framework. This is achieved by transforming the lexicon into a collection of synthetic word images, and then posing the recognition task as the problem of retrieving the best match from the lexicon image set. The retrieval framework introduced in our approach is similar in spirit to the influential work of [128] in the area of handwritten and printed word spotting. We, however, differ from their approach as follows. (1) Our matching score is based on a novel feature set, which shows better performance than the profile features in [128]. (2) We formulate the problem of finding the best word match in a maximum likelihood framework and maximize the probability of two features sequences originating from same word class. (3) We propose a robust way to find the match for a word, where k in k-NN is not hand picked, rather dynamically decided based on the randomness of the top retrievals.

Motivation and overview. The problem of recognizing text (including printed and handwritten text) has been addressed in many ways. Detecting characters and combining them to form a word is a popular approach as mentioned above [109, 165]. Often these methods suffer from weak character detections as shown in Fig. 5.1(a). An alternative scheme is to learn a model for words [15]. There are also approaches that recognize a word by first binarizing the image, and then finding each connected component [86]. These methods inherently rely on finding a model to represent each character or word. In the context of

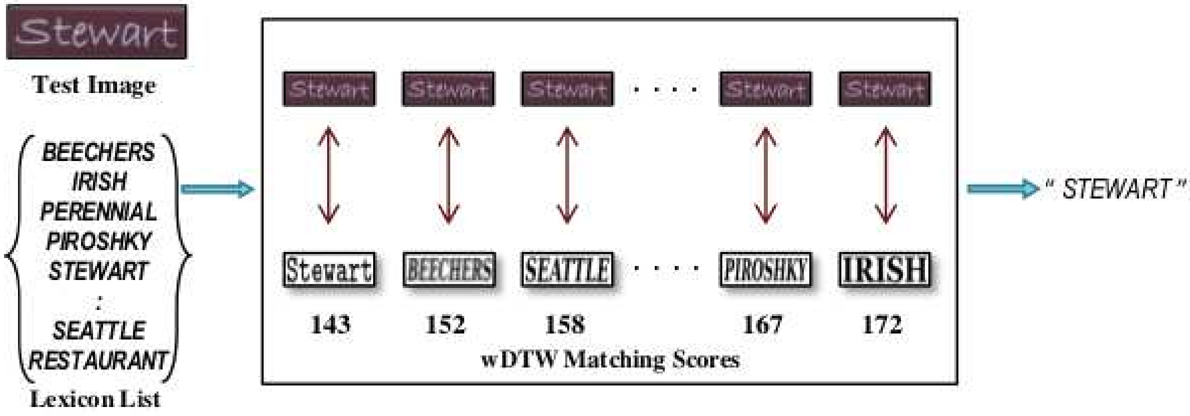
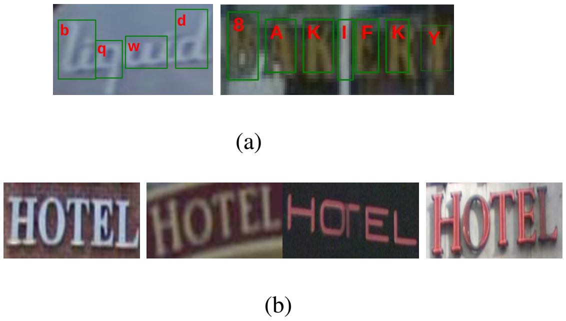


Figure 5.1 Overview of the proposed system. We recognize the word in the test image by matching it with synthetic images corresponding to the lexicon words. A novel gradient based feature set is used to represent words. Matching is done with a weighted DTW scores computed with these features. We use the top k matches to determine the most likely word in the the scene text image.

scene text recognition, this creates the need for a large amount of training data to cover the variations in scene text. Examples of such variations are shown in Fig. 5.1(b). Our method is designed to overcome these issues.

We begin by generating synthetic images for the words from the lexicon with various fonts and styles. Then, we compute gradient-based features for all these images as well as the scene text (test) image. We then recognize the text in the image by matching the scene and synthetic image features with our novel weighted dynamic time warping (DTW). The weights in the DTW matching scores are learned from the synthetic images, and determine the discriminativeness of the features. We use the top k retrieved synthetic images to determine the word most likely to represent the scene text image (see Section 5.2). An overview of our method is shown in Fig. 5.1.

We present results on two challenging public datasets, namely street view text (SVT) and ICDAR 2003 (see Section 5.3). We experimentally show that popular features like profile features are not robust enough to deal with challenging scene text images. Our experiments also suggest that the proposed *gradient at edges* based features outperform profile features for the word matching task. In addition to being simple, the proposed method improves the accuracy by more than 5% over recent works [109, 165, 167].



(a)

(b)

Figure 5.2 (a) Character detection is a challenging problem in the context of scene text images. (b) Large intra-class variations in scene text images makes it challenging to learn models to represent words.

The main contributions of this chapter are two fold: (i) We show that holistic word recognition for scene text images is possible with high accuracy, and achieve a significant improvement over prior art. (ii) The proposed method does not use any language-specific information, and thus can be easily adapted to any language. Additionally, the robust synthetic word retrieval for scene text queries also shows that our framework can be easily extended for text-to-image retrieval. However, this is beyond the scope of the chapter.

5.2 Word representation and matching

We propose a novel method to recognize the word contained in an image as a whole. We extract features from the image, and match them with those computed for each word in the lexicon. To this end, we present a gradient based feature set, and then a weighted dynamic time warping scheme in the remainder of this section.

Gradient based features. Some of the previous approaches binarize a word image into character vs non-character regions before computing features [86]. While such pre-processing steps can be effective to reduce the dimensionality of the feature space, it comes with its disadvantages. The results of binarization are seldom perfect, contain noise, and this continues to be an unsolved problem in the context of scene text images. Thus, we look for other effective features, which do not rely on binarized images. Inspired by the success of Histogram of Oriented Gradient (HOG) features [35] in many vision tasks, we adapted them to the word recognition problem.

To compute the adapted HOG features, we begin by applying the Canny edge operator on the image. Note that we do not expect a clean edge map from this result. We then compute the orientation of gradient at each edge pixel. The gradient orientations are accumulated into histograms over vertical (overlapping) strips extracted from the image. The histograms are weighted by the magnitude of the

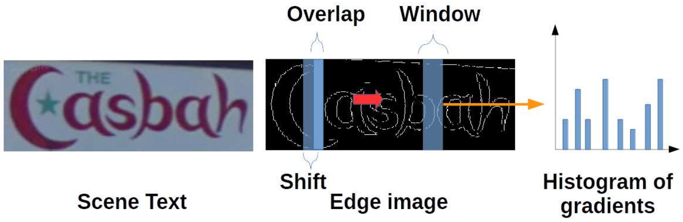


Figure 5.3 An illustration of feature computation. We divide the word image into vertical strips. In each strip we compute histogram of gradient orientation at edges. These features are computed for overlapping vertical strips.

gradient. An illustration of the feature computation process in shown in Fig. 5.3. At the end of this step, we have a representation of the image in terms of a set of histograms. In the experimental section we will show that these easy to compute features are robust for the word matching problem.

Matching words. Once words are represented using a set of features, we need a mechanism to match them. The problem is how to match the scene text and synthetic lexicon based images[1]. We formulate the problem of matching scene text and synthetic words in a maximum likelihood framework.

Let $X = \{x_1, x_2, \dots, x_m\}$ and $Y = \{y_1, y_2, \dots, y_m\}$ be the feature sequences from a given word and its candidate match respectively. Each vector x_i and y_i is a histogram of gradient features extracted from a vertical strip. Let $\omega = \{\omega_1, \omega_2, \dots, \omega_K\}$ represent a set of word images where K is the total number of lexicon words. Since we assume features at each vertical strips are independent, the joint probability that the feature sequences X and Y originate from the same word ω_k, i.e. $P(X, Y|\omega_k)$ can be written as the multiplication of joint probabilities of features originating from the same strip, i.e.,

$$P(X, Y|\omega_k) = \prod_i P(x_i, y_i|\omega_k). \tag{5.1}$$

In a maximum likelihood framework, the problem of finding an optimal feature sequence Y for a given feature sequence X is equivalent to maximize $\prod_i P(x_i, y_i|\omega_k)$ over all possible Ys. This can be written as minimization of an objective function f, i.e., $\min_Y \sum_i f(x_i, y_i|\omega_k)$. Where f is the weighted squared l^2-distance between feature sequences X and Y i.e.,$f(x_i, y_i) = (x_i - y_j)w_i(x_i - y_j)$. Here w_i is the weight to feature x_i. These weights are learned from the synthetic images, and are proportional to the discriminitiveness of features. In other words, given a feature sequence X and a set of candidate sequences Ys, the problem of finding the optimal matching sequence becomes as minimizing f over all candidate sequences Y. This leads to the problem of alignment of sequences. We propose a weighted dynamic programming based solution to solve this problem. Dynamic time warping [137] is used to

[1] Details for generating the synthetic lexicon-based images are given in Section 5.3.

compute a distance between two time series. The weighted DTW distance $DTW(m, n)$ between the sequences X and Y can be recursively computed using dynamic programming as:

$$DTW(i,j) = \min \begin{cases} DTW(i-1,j) + D(i,j) \\ DTW(i,j-1) + D(i,j) \\ DTW(i-1,j-1) + D(i,j), \end{cases} \tag{5.2}$$

where $D(i, j)$ is the distance between features x_i and y_j, and the local distance matrix D is written as: $D = (X - Y)^T W(X - Y)$. The diagonal matrix W is learnt from synthetic images. For this we cluster all the feature vectors computed over vertical strips of synthetic images and entropy of each cluster as follows.

$$H(\text{cluster}_p) = -\sum_{k=1}^{K} Pr(y_j \in \omega_k, y_j \in \text{cluster}_p) \times \log_K(Pr(y_j \in \omega_k, y_j \in \text{cluster}_p)), \tag{5.3}$$

where Pr is the joint probability of feature y_j originating from class ω_k and falling in cluster$_p$. High entropy of a cluster indicates that the features corresponding to that cluster are almost equally distributed in all the word classes. In other words, such features are less informative, and thus are assigned a low weight during matching. The weight w_j associated with a feature vector y_j is computed as: $w_j = 1 - H(cluster_p), \text{if } y_j \in cluster_p$.

Warping path deviation based penalty. To give high penalty to those warping paths (ref. Chapter 2) which deviate from the near diagonal paths we multiply them with a penalty function $\log_{10}(wp - wp_o)$, where wp and wp_o are warping path of DTW matching and diagonal warping path respectively. This penalizes warping paths where a small portion in one word is matched with a large portion in another word.

Dynamic k-NN. Given a scene text and a ranked list of matched synthetic words (each corresponding to one of the lexicon words), our goal is to find the text label. To do so, we apply k-nearest neighbor. One of the issues with a nearest neighbor approach is finding a good k. This parameter is often set manually. To avoid this, we use dynamic k-NN. We start with an initial value of k and measure the randomness of the top k retrievals. Randomness is maximum when all the top k retrievals are different words, and is minimum (i.e. zero) when all the top k retrieval are same. We increment k by 1 until this randomness decreases. At this point we assign the label of the most frequently occurring synthetic word to a given scene text.

In summary, given a scene text word and a set of lexicon words, we transform each lexicon into a collection of synthetic images, and then represent each image as a sequence of features. We then

Method	SVT-WORD	ICDAR(50)
Profile features + DTW [128]	38.02	55.39
Gradient based features + wDTW	75.43	87.25
NL + Gradient based features + wDTW	77.28	89.69

Table 5.1 Feature comparison: We observe that gradient based features outperform profile features for the holistic word recognition task. This is primarily due to the robustness of gradient features in dealing with blur, noise, large intra-class variations. Non-local (NL) means filtering of scene text images further improves recognition performance.

pose the problem of finding candidate optimal matches for a scene text image in a maximum likelihood framework and solve it using weighted DTW. The weighted DTW scheme provides a set of candidate optimal matches. We then use dynamic k-NN to find the optimal word in a given scene text image.

5.3 Experiments and results

In this section we present implementation details of our approach, and its detailed evaluation, and compare it with the best performing methods for this task, namely [109, 118, 165, 167].

5.3.1 Datasets

For the experimental analysis we used two datasets, namely street view text (SVT) [4] and ICDAR 2003 robust word recognition [11]. The SVT dataset contains images taken from Google Street View. We used the SVT-word dataset, which contains 647 images, relevant for the recognition task. A lexicon of 50 words is also provided with each image. The lexicon for the ICDAR dataset was obtained from [165]. Following the protocol of [165], we ignore words with less than two characters or with non-alphanumeric characters, which results in 863 words overall. Note that we could not use the ICDAR 2011 dataset since it has no associated lexicon.

5.3.2 Implementation details

Synthetic Word Generation. For every lexicon word we generated synthetic words with 20 different styles and fonts using ImageMagic.[2] We chose some of the most commonly occurring fonts, such as

[2] www.imagemagick.org/

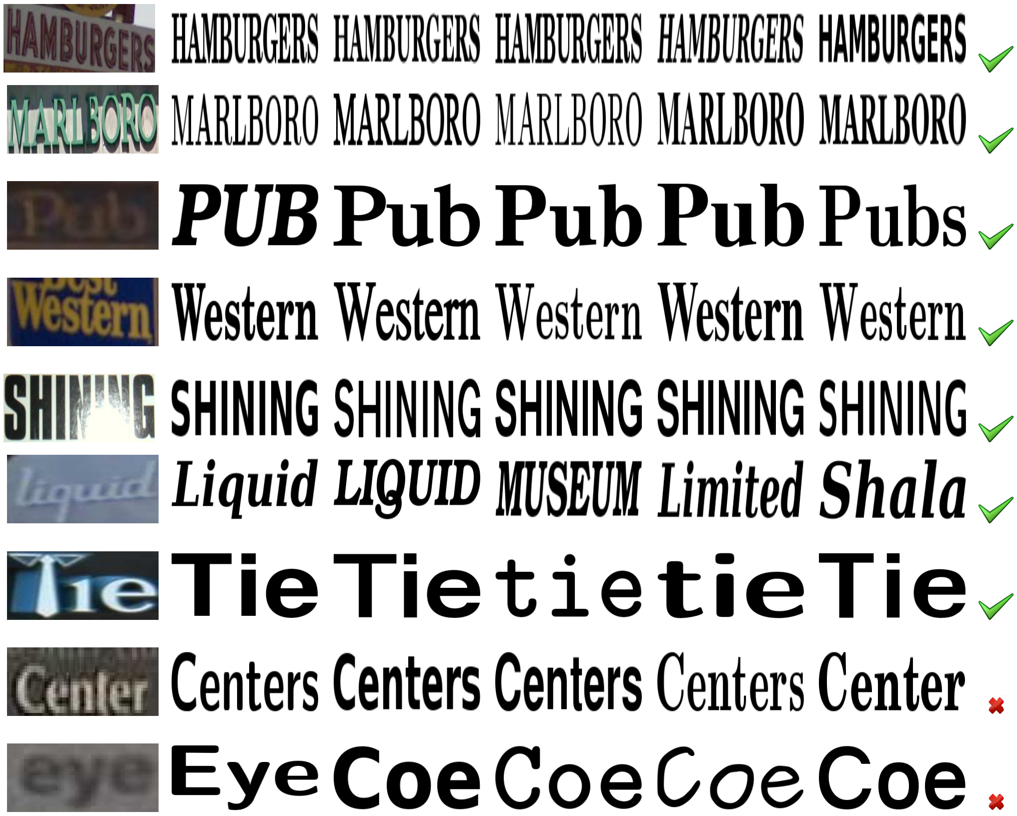

HAMBURGERS HAMBURGERS HAMBURGERS HAMBURGERS HAMBURGERS ✓
MARLBORO MARLBORO MARLBORO MARLBORO MARLBORO ✓
PUB Pub Pub Pub Pubs ✓
Western Western Western Western Western ✓
SHINING SHINING SHINING SHINING SHINING ✓
Liquid LIQUID MUSEUM Limited Shala ✓
Tie Tie tie tie Tie ✓
Centers Centers Centers Centers Center ✗
Eye Coe Coe Coe Coe ✗

Figure 5.4 Few sample results. Top-5 synthetic word retrieval results for scene text query. First column shows the test image. Top-5 retrieval for the test image are shown from left to right in each row. The icon in the right most column shows whether a word is correctly recognized or not. We observe that the proposed word matching method is robust to variations in fonts and character size. In the fourth row, despite the unseen style of word image “liquid" the top two retrievals are correct. (Note that following the experimental protocol of [165], we do case-insensitive recognition). The last two rows are failure cases of our method, mainly due to near edit distance words (like center and centers) or high degradations in the word image.

Method	SVT-WORD	ICDAR(50)
ABBYY [10]	35	56
Wang *et al.* [165]	56	72
Wang *et al.* [167]	70	**90**
Novikova *et al.* [118]	72	82
Mishra *et al.* [109]	73	82
This work	**77.28**	89.69

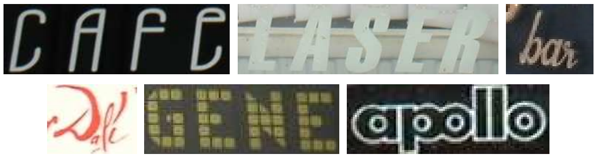
Table 5.2 Cropped word recognition accuracy (in %): We show a comparison of the proposed method to the popular commercial OCR system ABBYY and many recent methods. We achieve a significant improvement over previous works on SVT and ICDAR.

Arial, Times, Georgia. Our observations suggest that font selection is not a very crucial step for overall performance of our method. A five pixel-width padding was done for all the images. We noted that all the lexicon words were in uppercase, and that the scene text may contain lowercase letters. To account for these variations, we also generated word images where, (i) only the first character is in upper case; and (ii) all characters are in lower case. This results in $3 \times$ lexicon size $\times$ 20 images in the synthetic database. For the SVT dataset, the synthetic dataset contains around 3000 images.

Preprocessing. Prior to feature computation, we resized all the word images to a width of 300 pixels, with the respective aspect ratio. We then applied the popular non-local means filter smoothing on scene text images. We also remove the stray edges pixels less than 20 in number. Empirically, we did not find this filtering step to be very critical in our approach.

Features. We used vertical strips of width 4 pixels and a 2-pixel horizontal shift to extract the histogram of gradient orientation features. We computed signed gradient orientation in this step. Each vertical strip was represented with a histogram of 9 bins. We evaluated the performance of these features in Table 5.1, in comparison with that of profile features used in [128]. Profile features consist of: (1) projection profile, which counts the number of black pixels in each column. (2) upper and lower profile, which measures the number of background pixels between the word and the word-boundary (3) transition profile, is calculated as the number of text-background transitions per column. We used the binarization method in [107] prior to computing the profile features. Profile features have shown noteworthy performance on tasks such as handwritten and printed word spotting, but fail to cope with the additional complexities in scene text (e.g., low contrast, noise, blur, large intra-class variations).

Figure 5.5 Few images from ICDAR 2003 dataset where our method fails. This may be addressed with inclusion of more variations in our synthetic image database.



Infact, our results show that gradient features substantially outperform profile based features for scene text recognition.

Weighted dynamic time warping. In our experiments we used 30 clusters to compute the weights. Our analysis comparing various methods are shown in Table 5.1. We observe that with wDTW, we achieve a high recognition accuracy on both the datasets.

Dynamic k-nearest neighbor. Given a scene text image to recognize, we retrieve word images from database of synthetic words. The retrieval is ranked based on similarity score. In other words, synthetic words more similar to the scene text word get a higher rank. We use dynamic k-NN with an initial value of $k = 3$ for all the experiments.

5.3.3 Comparison with previous work

We retrieve synthetic word images corresponding to lexicon words and use dynamic k-NN to assign text label to a given scene text image. We compared our method with the most recent previous works related to this task, and also the commercial OCR ABBYY in Table 5.2. From the results, we see that the proposed holistic word matching based scheme outperforms not only our earlier work [109], but also many recent works as [118, 165, 167] on the SVT dataset. On the ICDAR dataset, we perform better than almost all the methods, except [167]. This marginally inferior performance (of about 0.3%) is mainly because our synthetic database fails to model few of the fonts in ICDAR dataset (Fig. 5.3.3). These type of fonts are rare in the street view images. A specific preprocessing or more variations in the synthetic dataset may be needed to deal with such fonts. Fig. 5.4 shows the qualitative performance of the proposed method on sample images. We observe that the proposed method is robust to noise, blur, low contrast and background variations.

In addition to being simple, our method significantly improves the prior art. This gain in accuracy can be attributed to the robustness of our method, which (i) does not rely on character segmentation rather do holistic word recognition; and (ii) learns discriminitiveness of features in a principled way and use this information for robust matching using wDTW.

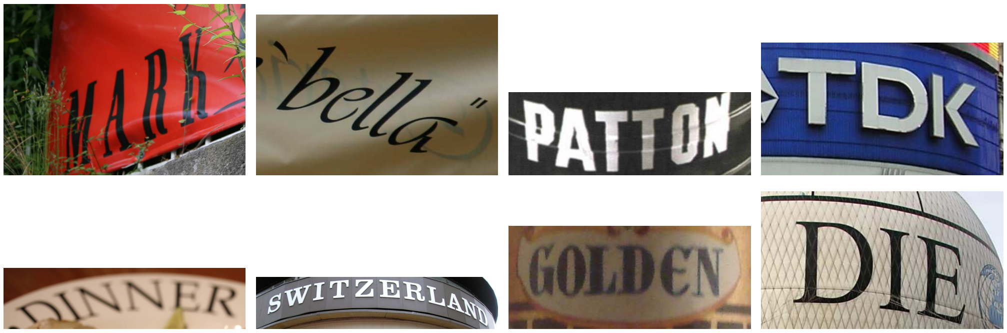


Figure 5.6 A few sample images with text in non-planner surface, e.g., text on cloth banner and curved surfaces. We collected a small subset of such images from NEOCR [12] and our own collection, and evaluated our recognition methods on it.

5.4 Extension to specific cases

We have tested our methods on public benchmark datasets. These datasets contain images from street scenes with different view angle, variable illumination, different contrast and occlusion. However, we have not verified our methods on specific cases such as text on curved surface and cloth banner. This is mainly due to the unavailability of sufficient training data for these specific cases of scene text images.

We conducted a small experiment to demonstrate the generality and extensibility of our recognition methods. To this end, we collected a dataset of 20 images where text is written on non-planer surfaces. These images are manually collected from NEOCR dataset [12] and our personal collection. We have created a lexicon of size 50 corresponding for each word. Few of the images of this dataset are shown in Figure 5.6. We observe that these images have non-linear distortions.

We have tested our two recognition methods on this small test set, namely higher order CRF method (Chapter 4) and holistic recognition method (this chapter). Our methods as such have limited success on recognizing these images. We achieve 50% and 65% word recognition accuracies with these methods respectively. This inferior performance is mainly due to fact that the character and word distortions present in these images are not seen by these methods.

In order to extend our methods to these categories of scene texts, we enriched our dataset with examples of images subjected to non-linear transformations. For this we created synthetic images with ten popular fonts and two different plane to cylinder transformations. We have shown few examples of these synthetic images in Figure 5.5. We added these images to our synthetic database, and the

Attributes	**OCR based**	**Higher order CRF**	**Holistic recognition**
Cursive font	×	×	✓
Uniform background	✓	×	×
Complex background	×	✓	×
Low contrast	×	×	✓
Occlusion	×	×	✓
High Illumination change	×	×	✓
Digital-born images	×	✓	×

Table 5.3 Which method is effective where? Here tick mark under any attribute a and the method m shows that the method m is more effective as compared to others in recognizing scene text having attribute a.

extracted characters of these synthetic words to our character training set. We re-trained our character classifier with this enrichment to the training data. We, then re-evaluated our recognition methods, i.e., higher order CRF method and holistic recognition method. These methods achieved 60% and 80% word recognition accuracy on this set respectively, which is significantly better as compared to original implementation. This shows the robustness and generalization of our methods on scene text with variety of real world challenges. Although our experiment is preliminary with a small number of images, we believe with some more engineering efforts our performance on recognizing specific cases of scene text can be further enhanced.

5.5 Comparison with our other recognition methods

We proposed three effective ways for scene text recognition problem in our thesis: (i) our first approach was the off-the-shelf OCR with our binarization (Chapter 3), (ii) in our second approach of word recognition, we proposed a method where character detection scores and languages priors are integrated in a higher order CRF (Chapter 4), and (iii) finally, in this chapter we proposed a holistic recognition method which bypasses the need of character localization and binarization. We will refer these methods as OCR based method, higher order CRF method and holistic recognition method respectively from here onwards.

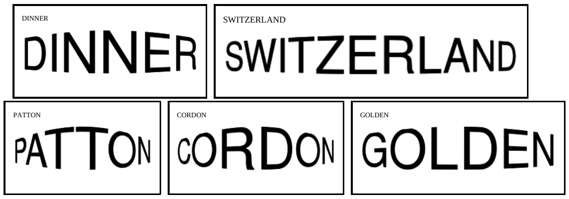


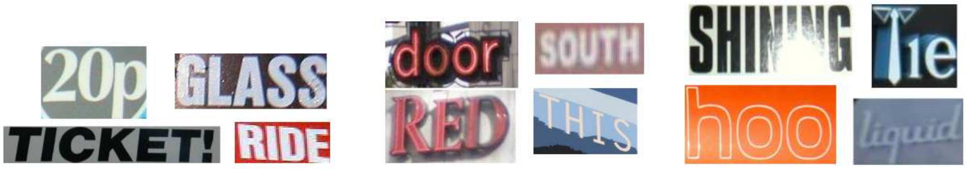
Figure 5.7 We added few text images with non-linear transformation, e.g., plane to cylinder transformation to our synthetic text image database.

For a deeper analysis of the results, we categorized the images from subsets of the datasets we use, based on presence of following attributes : (i) cursive fonts, (ii) uniform background, (iii) complex background, (iii) low contrast, (iv) occlusion, (v) high illumination change, and (vi) digital-born images. We evaluated all our methods separately on images with above categories of attributes. We found that our proposed methods are complementary in nature, and each method is superior to others in addressing a specific challenge, e.g., our OCR based method performs better than others if background is uniform, our holistic recognition methods works better as compared to others in case of occlusion, and so on. Table 5.3 summarizes various attributes present in scene text images, and which of our method is more effective as compared to others in recognition of scene text with specific attributes. In Figure 5.8, we show few example scene text images and the method which is more effective in recognizing them.

We performed another simple experiment on IIIT-5K dataset with small lexicon to validate the complementary nature of our proposed recognition methods. On this dataset our OCR based, higher order CRF and holistic recognition methods achieve 68%, 78%, and 75% respectively. When we further analyze these results and found that 84% of the words are recognized correctly by one of the methods. This implies that a simple combinations of these methods can be used for applications like text-to-image retrieval or suggesting multiple recognition outputs (three in our case) with one being correct with a high rate. Smart ways of combining these recognition methods can also be explored in the future, for example, lexicon reduction methods purposed in [135] on top of our higher order CRF method can be used to reduce the lexicon size with preserving the ground truth in the reduced lexicon. Once lexicon size is reduced our holistic recognition method can be more effectively used to recognize the words.

5.6 Summary

In this chapter, we proposed an holistic method to recognize scene text. Our method neither requires character segmentation nor relies on binarization, but instead performs holistic word recognition. We show a significantly improved performance over the most recent works from 2011 and 2012. The



(a) (b) (c)

Figure 5.8 Images categorized according to the challenges they represent. We observe that for cleaner images, as shown in (a), OCR based method is most effective. Our higher order CRF method have a bias towards a dictionary word, e.g. it recognizes 20p as 200. Moreover, since our higher order CRF method is not trained on special characters, it does mistakes on recognizing words with special characters. On the other hand our holistic recognition method makes mistakes in case of its matching with similar appearance words, e.g., RIDE gets confused with BIKE, GLASS with CLASS. For images with low contrast, complex background images, fancy fonts, as shown in (b), our higher order CRF method performs better than others. For the cursive fonts, missing or fancy characters our holistic recognition method shows better performance. Few such examples are shown in (c).

robustness of our word matching approach shows that the natural extension of this work can be in direction of “text to scene image" retrieval.

We have also compared our holistic method with two other effective recognition methods proposed in Chapter 3 and Chapter 4 respectively. All these recognition methods are complementary in nature, and can be combined effectively in future for higher scene text recognition rates.

Chapter 6

Text2Image Retrieval

In this chapter, we present an approach for the text-to-image retrieval problem based on textual content present in images. Given the recent developments in understanding text in images, an appealing approach to address this problem is to localize and recognize the text, and then query the database, as in a text retrieval problem. We show that such an approach, despite being based on state of the art methods, is insufficient, and propose a method, where we do not rely on an exact localization and recognition pipeline. We take a query-driven search approach, where we find approximate locations of characters in the text query, and then impose spatial constraints to generate a ranked list of images in the database. The retrieval performance is evaluated on public scene text datasets as well as three large datasets, namely IIIT scene text retrieval, Sports-10K and TV series-1M, we introduce. We further boost our performance by using deep character classifier.

6.1 Introduction

In the context of ever-growing large data collections, there are many challenging problems like searching for, and retrieving relevant content. One approach to retrieval uses *text as a query*, with applications such as Google image search, which relies on cues from meta tags or text available in the context of the image. The success of this approach is rather limited by the quality of the meta tags and the contextual text. An alternate approach like Video Google [149] enables image search using *image as a query*, by finding visually similar regions in the database. Although this method exploits the visual content, it may not necessarily be sufficient. For instance, consider four photos of restaurants shown in Figure 6.1. There is very little visual information to suggest that these four images are of restaurants, and thus are unlikely to be retrieved together by such methods. However, the fact that all these images

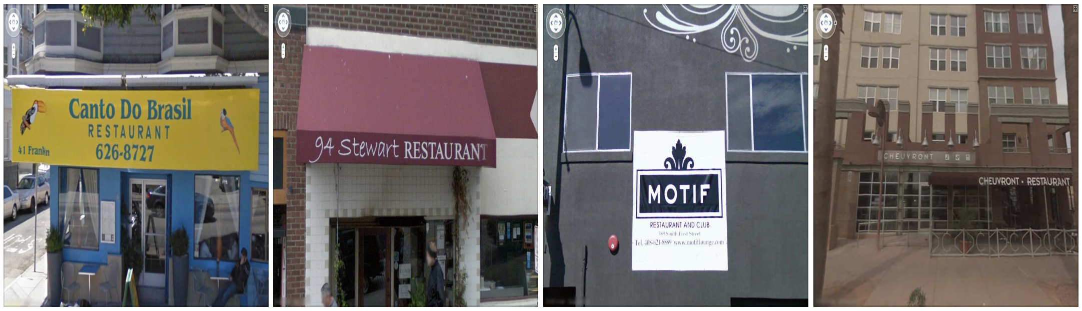


Figure 6.1 Consider an example query for restaurants. Here we show four images of restaurants, which have insufficient visual cues (such as building style) to group them into a single restaurant category. On the other hand, the text “restaurant" appearing on the banner/awning is an indispensable cue for retrieval. We present a text-to-image retrieval method based on the textual content present in images.

contain the word *restaurant* is a very useful cue in grouping them. In this work, we aim to fill this gap in image retrieval with *text as a query*, and develop an image-search based on the textual content present in it.

The problem of recognizing text in images or videos has gained a huge attention in the computer vision community in recent years [29, 40, 109, 143, 145, 165, 166]. Although exact localization and recognition of text in the wild is far from being a solved problem, there have been notable successes. We take this problem one step further and ask the question: *Can we search for query text in a large collection of images and videos, and retrieve all occurrences of the query text?* Note that, unlike approaches such as Video Google [149], which retrieve only similar instances of the queried content, our goal is to retrieve instances (text appearing in different places or view points), as well as categories (text in different font styles).

Plausible approaches. One approach for addressing the text-to-image retrieval problem is based on text localization, followed by text recognition. Once the text is recognized, the retrieval task becomes equivalent to that of text retrieval. Many methods have been proposed to solve the text localization and recognition problems [31, 40, 108, 109, 114]. We adapted two of these methods for our analysis with the implementation from [13, 14]. We transformed the visual text content in the image into text, either with [114] directly, or by localizing with [40], and then recognizing with [109]. In summary, we recognize the text contained in all images in the database, search for the query text, and then rank the images based on minimum edit distance between the query and the recognized text.

Method	**mAP**
Neumann and Matas [114]	23.32
SWT [40]+ Mishra *et al.* [109]	19.25
Wang *et al.* [165]	21.25

Table 6.1 Baseline results for text-to-image retrieval on the street view text dataset [165] are shown as mean average precision (mAP) scores. All the unique ground truth words in the dataset are used as queries. The first two methods, based on the state of the art text localization and recognition schemes, perform poorly. Wang *et al.* [165] is a word spotting method, which detects and recognizes the lexicon words in an image. In comparison, our approach, which does not rely on an exact localization and recognition pipeline, achieves an mAP of 56.24 (see Table 6.3).

Table 6.1 shows the results of these two approaches on the street view text (SVT) dataset. Note that both of them fail to achieve a good performance. This poor show is likely due to the following: (i) The loss of information during localization/recognition is almost irreversible. (ii) The recognition methods are not query-driven, and do not take advantage of the second or the third best predictions of the classifier. (iii) The variation in view point, illumination, font style, and size lead to incorrect word localization and recognition. In other words, these approaches heavily rely on the localization and the recognition performance, making them susceptible to failures in both these phases.

In terms of not relying on an explicit localization, the closest to our work is [165]. Although it is a method for spotting (detecting and recognizing) one of the few ($\sim$50) lexicon words in one image. In contrast, we aim to spot query words in millions of images, and efficiently retrieve all occurrences of query. Thus our goals are different. Furthermore, the success of [165] is largely restricted by the size of the lexicon. We have performed two tests to show that adapting it to our problem is inferior to our proposed approach. (i) Using all the query words as lexicon, it gives a mean AP of 21.25% on the SVT dataset (see Table 6.1). (ii) Using their character detection, and then applying our indexing and re-ranking schemes, we obtain an mAP of 52.12%, about 4% lower than our approach.

Another plausible approach is based on advancements in retrieving similar visual content, e.g. bag of words based image retrieval [149]. Such methods are intended for instance retrieval with *image as a query*. It is not clear how well text queries can be used in combination with such methods to retrieve scene text appearing in a variety of styles.

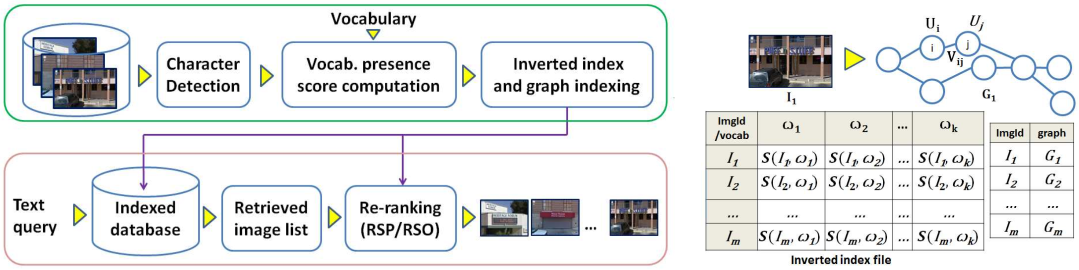
Proposed method. We take an alternate approach, and do not rely on an accurate text localization and recognition pipeline. Rather, we do a query-driven search on images and spot the characters of the words of a vocabulary[1] in the image database (Section 6.2.1). We then compute a score characterizing the presence of characters of a vocabulary word in every image. The images are then ranked based on these scores (Section 6.2.2). The retrieval performance is further improved by imposing spatial positioning and ordering constraints (Section 6.2.3). We demonstrate the performance of our approach on publicly available scene text datasets. For a more comprehensive study, we not only need a large dataset with diversity, but also a dataset containing multiple occurrences of text in different fonts, view points and illumination conditions. To this end, we introduce two video datasets, namely *Sports-10K* and *TV series-1M*, with more than 1 million frames, and an image dataset, IIIT scene text retrieval (STR). To our knowledge, the problem of text-to-image retrieval has not been looked at in such a challenging setting yet.

Another possible way to solve the retrieval performance is to obtain text proposals, recognize them and do a simple text based search. Object proposals are the recent trend in object detection community [179]. Inspired by their success text proposal methods are also devised [62]. These methods provide a ranked list of candidate text regions. These regions with recognition pipeline can significantly boost the retrieval performance of the method. The retrieval results obtained using text proposal based methods can also be ensembled with ours in future.

6.2 Scene text indexing and retrieval

Our retrieval scheme works as follows: we begin by detecting characters in all the images in the database. After detecting characters, we have their potential locations. We assume that a set of vocabulary words, is given to us *a priori*. We then spot characters of the vocabulary words in the images and compute a score based on the presence of these characters. Given our goal of retrieving images from a large dataset, we need an efficient method for retrieval. To achieve this, we create an inverted index file containing image id and a score indicating the presence of characters of the vocabulary words in the image. Initial retrievals are obtained using the inverted index. We then re-rank the top-n initial retrievals by imposing constraints on the order and the location of characters from the query text. Figure 6.2 summarizes our indexing and retrieval scheme.

[1] We define vocabulary as a set of possible query words.



ImgId /vocab	ω_1	ω_2	...	ω_k
I_1	$S(I_1, \omega_1)$	$S(I_1, \omega_2)$	...	$S(I_1, \omega_k)$
I_2	$S(I_2, \omega_1)$	$S(I_2, \omega_2)$	...	$S(I_2, \omega_k)$
...	...	...	...	...
I_m	$S(I_m, \omega_1)$	$S(I_m, \omega_2)$	...	$S(I_m, \omega_k)$

ImgId	graph
I_1	G_1
I_2	G_2
...	...
I_m	G_m

(a) (b)

Figure 6.2 Summary of our indexing and retrieval scheme. (a) In the offline phase, we first detect the characters. We then compute a score indicating the presence of characters from the vocabulary words (vocabulary presence score), and create an inverted index file with this score and image id. In the online phase, user provides a query, which is searched on the indexed database to retrieve images based on the vocabulary presence score. The top-n retrievals are then re-ranked using our re-ranking schemes. (b) After character detection, an image I_m is represented as a graph G_m, where nodes correspond to potential character detections and edges model the spatial relation between two detections. The nodes are characterized by their character likelihood vector U, and the edges by their character pair priors V. This graph is used to prune false positive detections, and also to impose order and position constraints on the characters during the re-ranking phase. See Section 6.2 for details.

6.2.1 Potential character localization

Given a large collection of images or video frames, the first step of our retrieval pipeline is to detect potential locations of characters. We do not expect ideal character detection from this stage, but instead obtain many potential character windows, which are likely to include false positives. To achieve this, we train a linear SVM classifier with HOG features [35]. We then use a sliding window based detection to obtain character locations and their likelihoods. The character localization process is illustrated in Figure 6.3. Note that this is an offline step in our retrieval pipeline.

For a robust localization of characters using sliding windows, we need a strong character classifier. The problem of classifying natural scene characters typically suffers from the lack of training data, e.g. [36] uses only 15 samples per class. It is not trivial to model the large variations in characters using only a few examples. Also, elements in a scene may interfere with the classifier, and produce many false positives. For example, the corner of a door can be detected as the character ‘L’. To deal with these issues, we add more examples to the training set by applying small affine transformations to the original character images.[2] We further enrich the training set by adding many negative examples (non-characters, i.e. background). With this strategy, we achieve a significant boost in character classification.

We use a multi-scale sliding window based detector, which is popular in many applications [35, 164, 165]. Each window is represented by its top-left (x, y) position, width and height in the original scale. Let $\mathcal{K}$ be the set of all character classes, i.e. English characters (A-Z, a-z), digits (0-9) and a background class. Given a window i, we compute the likelihood, $P(l_i|hog_i)$, $_i \in \mathcal{K}$, using Platt's method [124]. Here hog_i denotes the HOG features extracted from the window i. This results in a 63-dimensional vector for every window, which indicates the presence of a character or background in that window. We then perform character-specific non maximal suppression (NMS) to prune out weak windows. Further, since for a given query word, we wish to retrieve all the images where the query word appears either in upper or lower case, we transform the 63-dimensional vector to a 36-dimensional vector by taking the maximum between the upper and lower case likelihoods for every character and dropping the likelihood for background.

6.2.2 Indexing

Once the characters are detected, we index the database for a set of vocabulary words. Consider a set of vocabulary words $\{\omega_1, \cdots, \omega_k\}$, which are given to us *a priori*. In a general setting, k can be as large as the number of words in English or all the words that we are interested in querying.

We first remove a few spurious character windows. To do so, we construct a graph, where each character detection is represented as a node. These nodes are connected via edges based on their spatial proximity. We then use contextual information, window width, size and spatial distance to remove some of the edges. In other words, edges between two neighboring characters are removed if: (i) The width ratio of two neighboring character windows exceeds θ_{width}, or (ii) The spatial distance between two character windows is more than θ_{dist}, or (iii) The height ratio of two neighbouring character windows exceeds θ_{height}. The thresholds $\theta_{width}, \theta_{dist}$ and θ_{height} are estimated from the training set. This may result in isolated nodes, which are discarded. This step essentially removes many false character windows scattered in the image.

Each node of this graph is described by a 36-dimensional vector U_i. Further, assuming these likelihoods are independent, we compute the joint probabilities of character pairs for every edge. In other

[2] Note that the use of affine transformations in training examples is shown to improve classification accuracy [111, 147].

words, we associate a 36×36 dimensional matrix V_{ij} containing joint probabilities of character pairs to the edge connecting nodes i and j (see Figure 6.2(b)).

Now, consider a word from the vocabulary $\omega_k = \omega_{k1}\omega_{k2}\cdots\omega_{kp}$, represented by its characters ω_{kl},$1 \leq l \leq p$, where p is the length of the word. To index an image I_m for this word, we divide the image I_m into horizontal strips, each of height H. We then compute a score denoting the presence of characters from the query in these horizontal strips. This score for an image I_m and a word ω_k, $S(I_m, \omega_k)$, is computed as the maximum over all the horizontal strips of the image. In other words, score $S(I_m, \omega_k)$ is given by:

$$\max_h \sum_{l=1}^{p} \max_j U_j(\omega_{kl}) = \max_h \sum_{l=1}^{p} \max_j P(\omega_{kl}|hog_j), \tag{6.1}$$

where j varies over all the bounding boxes representing potential characters whose top-left coordinate falls in the horizontal strip and h varies over all the horizontal strips in the image. To avoid the dominance of a single character, we modify the score in (6.1) as:

$$S(I_m, \omega_k) = \max_h \sum_{l=1}^{p} \min(\max_j P(\omega_{kl}|hog_j), \tau), \tag{6.2}$$

where τ is a truncation constant.

Once these scores are computed for all the words in the vocabulary and all the images in the database, we create an inverted index file [102] containing image id, the vocabulary word and its score. We also store the image and its corresponding graph (representing character detections) in the indexed database. These graphs and the associated probabilities are used in our re-ranking schemes, which we will describe in the following section.

6.2.3 Retrieval and re-ranking

We use the inverted index file to retrieve the images and rank them based on the score computed in (6.2). This ensures that images containing characters from the query text have a high likelihood in a relatively small area (the horizontal strip of height H) get a higher rank. However, not all relevant images may be ranked well in this step, as it does not ensure the correct ordering and positioning of characters. To address this, we propose two methods to re-rank the results as follows.

Spatial ordering. Character spotting does not ensure that characters are spotted in the same order as in the query word. We address this by proposing a re-ranking scheme based on spatial ordering (RSO). Let $\psi_{total} = \{\sqcup\omega_{k1}, \omega_{k1}\omega_{k2}, \cdots, \omega_{kp}\sqcup\}$ be the set of all the bi-grams present in the query word ω_k, where

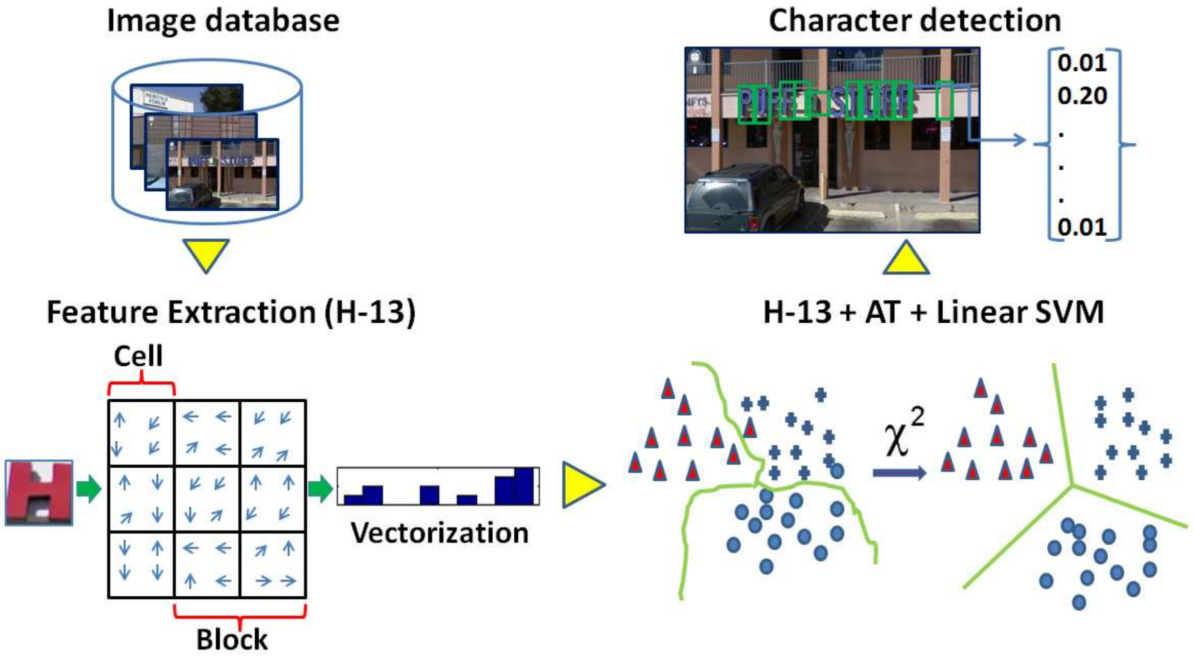


Figure 6.3 Potential character localization. We compute HOG features at various scales for all the images. These features are then represented using the χ^2 kernel. A linear SVM trained on affine transformed (AT) training samples is used to obtain potential character windows. This results in a 63-dimensional vector for every window, which denotes the likelihood of every character/background class in that window.

$\sqcup$ denotes whitespace. We also construct a set $\psi_{present}$ containing the pairs of spatially neighbouring spotted characters. We now define the score of spatial ordering as $S_{so}(I_m, \omega_k) = \frac{|\psi_{present} \cap \psi_{total}|}{|\psi_{total}|}$, where $|\cdot|$ is the cardinality. The score $S_{so}(I_m, \omega_k) = 1$, when all the characters in the query word are present in the image, and have the same spatial order as the query word. We use this score to re-rank retrieval results.

Spatial positioning. The re-ranking scheme based on spatial ordering does not account for spotted characters being in the correct spatial position. In other words, these characters may not have uniform inter-character gap. To address this, we use the graphs representing the character detections in the images, the associated U vectors, and the matrix V to compute a new score. We define a new score characterizing the spatial positioning of characters of the query word in the image as $S_{sp}(I_m, \omega_k) =$

$$\sum_{l=1}^{p} \min(\max_i U_i(\omega_{kl}), \tau) + \sum_{l=1}^{p-1} \max_{ij} V_{ij}(\omega_{kl}, \omega_{kl+1}). \tag{6.3}$$

This new score is high when all the characters and bi-grams are present in the graph in the same order as in the query word and with a high likelihood. Additionally, higher value of new score ensures the correct spatial positioning of the characters. This is because the graph is constructed such that nodes representing characters spatially close to each other are connected. The retrieval results are then re-ranked based on the summation of this score and the score S_{so} obtained from spatial ordering. We refer to this scheme as re-ranking based on spatial positioning (RSP).

6.2.4 Implementation details

Character detection. We use an overlap threshold of 40% to discard weak detection windows in the non maximal suppression stage. The character classifiers are trained on the train sets of ICDAR 2003 character [3] and Chars74K [36] datasets. We harvest 48×48 patches from scene images, with buildings, sky, road and cars, which do not contain text, for additional negative training examples. We then apply affine transformations to all the character images, resize them to 48×48, and compute HOG features. We analyzed three different variations [46] (13, 31 and 36-dimensional) of HOG. To efficiently train the classifier with a large set of training examples, we use an explicit feature map [162] and the χ^2 kernel. This feature map allows a significant reduction in classification time as compared to non-linear kernels like RBF. The performance of this classifier is evaluated in Chapter 4. Additionally, we also evaluate our retrieval performance by using state of the art CNN character classifier [67].

Score computation. We divide the images into horizontal strips of height 30 pixels and spot characters from a set of character bounding boxes, as described in Section 6.2.2. The idea here is to find images where the characters of the vocabulary word have a high likelihood in a relatively small area. We set the truncation parameter $\tau = 0.2$ in (6.2) empirically, and retrieve an initial set of top-100 results with this score and re-rank them by introducing spatial ordering and positioning constraints.

6.3 Datasets

We evaluate our approach on three scene text (SVT, ICDAR 2011 and IIIT scene text retrieval) and two video (Sports-10K and TV series-1M) datasets. The number of images and queries used for these datasets are shown in Table 6.2.

Street view text [4] and ICDAR 2011 [141]. These two datasets were originally introduced for scene text localization and recognition. They contain 249 and 255 images respectively. We use all the unique

Datasets	**# queries**	**# images/frames**
SVT [141]	427	249
ICDAR [141]	538	255
IIIT STR	50	10K
Sports-10K	10	10K
TV series-1M	20	1M

Table 6.2 Scene text datasets (SVT and ICDAR) contain only a few hundred images. We introduce an image (IIIT scene text retrieval) and two video (Sports-10K and TV series-1M) datasets to test the scalability of our proposed approach.

ground truth words of these datasets as queries and perform text-to-image retrieval.

IIIT scene text retrieval dataset. The SVT and ICDAR 2011 datasets, in addition to being relatively small, contain many scene text words occurring only once. To analyze our text-to-image retrieval method in a more challenging setting, we introduce IIIT scene text retrieval (STR) dataset. For this, we collected data using Google image search with 50 query words such as Microsoft building, department, motel, police. We also added a large number of distractor images, i.e. images without any text, downloaded from Flickr into the dataset. Each image is then annotated manually to say if it contains a query text or not. This dataset contains 10K images in all, with 10-50 occurrences of each query word. It is intended for category retrieval (text appearing in different fonts or styles), instance retrieval (text imaged from a different view point), and retrieval in the presence of distractors (images without any text).

Video datasets. To analyze the scalability of our retrieval approach, we need a large dataset, where query words appear in many locations. In this context, we introduce two video datasets. The first one is from sports video clips, containing many advertisement signboards, and the second is from four popular TV series: Friends, Buffy, Mr. Bean, and Open All Hours. We refer to these two datasets as Sports-10K and TV series-1M respectively. The TV series-1M contains more than 1 million frames. Words such as *central*, *perk*, *pickles*, *news*, *SLW27R* (a car number) frequently appear in the TV series-1M dataset. All the image frames extracted from this dataset are manually annotated with the query text they may contain.

Annotations are done by a team of three people for about 150 man-hours. We use 10 and 20 query words to demonstrate the retrieval performance on the Sports-10K and the TV series-1M datasets respectively. All our datasets are available on the project website.

Dataset	Char. spot.	RSO	RSP
With HOG character classifier			
SVT	17.31	46.12	56.24
ICDAR11	24.26	58.20	65.25
IIIT STR	22.11	36.34	42.69
With deep character classifier			
SVT	25.50	50.25	**62.15**
ICDAR11	30.00	64.73	**69.55**
IIIT STR	24.44	39.00	**44.50**

Table 6.3 Quantitative evaluation of text-to-image retrieval. We achieve a notable improvement in mAP with the proposed re-ranking schemes over baseline methods shown in Table 6.1. Another baseline we compare with uses character detections from [165] in combination with our spatial positioning re-ranking scheme, which achieves 52.12% mAP on SVT, over 10% lower than our result.

6.4 Experimental analysis

Given a text query our goal is to retrieve all images where it appears. We aim instance, i.e., text appearing in different view points, as well as category retrieval, i.e., text in different fonts and styles. In this section, we evaluate all the components of the proposed method to justify our choices.

6.4.1 Retrieval results

We first evaluate our retrieval scheme on image datasets. The retrieval performance is quantitatively evaluated using the well-known mean average precision (mAP) measure, which is the mean of the average precision for all the queries. The results are summarized in Table 6.3. We observe that the performance of our initial naive character spotting method is comparable to the baselines in Table 6.1. The re-ranking scheme improves the performance, and we achieve an mAP of 56.24% on SVT and 65.25% on ICDAR. Recall from Table 6.1 that the state of the art localization and recognition based method only achieves an mAP of 23.32% on SVT. Reasonably high performance on IIIT STR, which contains instances (text in different viewpoints), categories (text in different fonts), and distractors (images without any text) shows that the proposed method is not only applicable to retrieve instances and categories of scene texts, but also robust to distractors. Additional gain in mAP due to change of a better performing CNN classifier is also clearly noticed in all the datasets.

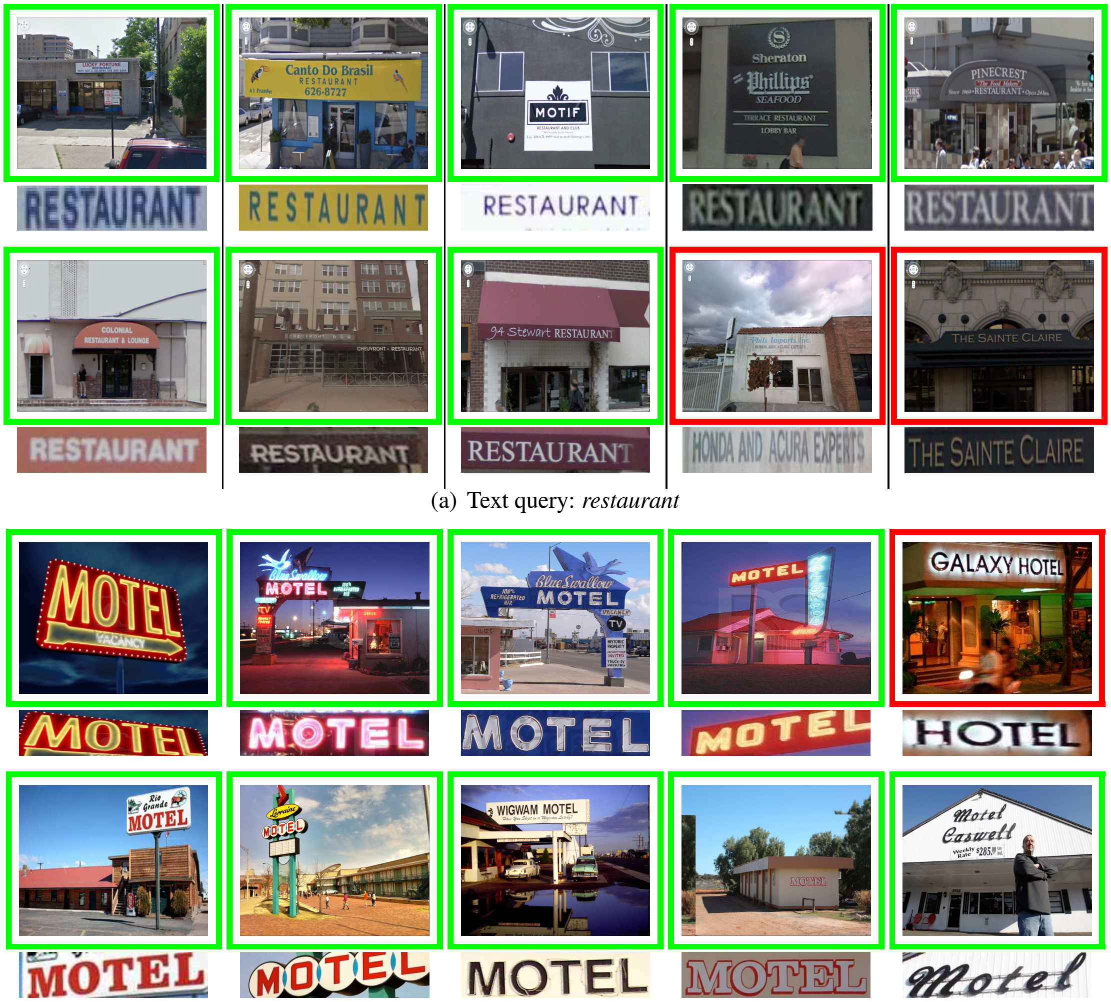
Dataset	**Char. Spot.**		**RSO**		**RSP**	
	P@10	**P@20**	**P@10**	**P@20**	**P@10**	**P@20**
With HOG character classifier						
Sports	26.21	24.26	39.11	38.32	44.82	43.42
TV series	40.22	39.20	58.15	57.21	59.28	59.02
With deep character classifier						
Sports	29.12	26.25	42.24	41.50	47.20	46.25
TV series	44.25	44.00	62.12	61.85	64.15	63.88

Table 6.4 Quantitative analysis of retrieval results on video datasets. We choose 10 and 20 query words for Sports-10K and TV series-1M respectively. We use top-n retrieval to compute precision at n (denoted by P@n).

We then evaluate the scalability of our proposed scheme on two large video datasets. We use precision computed using the top-n retrievals (denoted by P@n) as the performance measure. These results on video datasets are summarized in Table 6.4. The proposed re-ranking scheme achieves P@20 of 43.42% and 59.02% on Sports-10K and TV series-1M datasets respectively. Low resolution videos and fancy fonts appearing in advertisement boards make the Sports-10K dataset challenging, and thus the precision values are relatively low for this dataset.

Our indexing scheme allows us to retrieve images from a large dataset containing 1M images in about 3 seconds. The sliding window based character detection step and computation of index file are performed offline. They take around 9 seconds and 7 seconds per image.

Qualitative results of the proposed method are shown in Figure 6.4 for the query words *restaurant* on SVT, *motel* and *department* on IIIT STR. We retrieve all the occurrences of the query *restaurant* from SVT. The IIIT STR dataset contains 39 different occurrences of the word *motel*, with notable variations in font style, view point and illumination. Our top retrievals for this query are quite significant, for instance, the tenth retrieval, where the query word appears in a very different font. The query word *department* has 20 occurrences in the dataset. Few of these occurrences are on the same building. We observe that, overcoming the changes in the visual content, the relevant images are ranked high. Figure 6.5(a) shows precision-recall curves for two text queries: *department* and *motel* on IIIT STR. Our method achieves AP = 74.00 and 48.69 for these two queries respectively. The method tends to fail in cases where almost all the characters in the word are not detected correctly or when the query text appears vertically. A few such cases are shown in Figure 6.5(b).

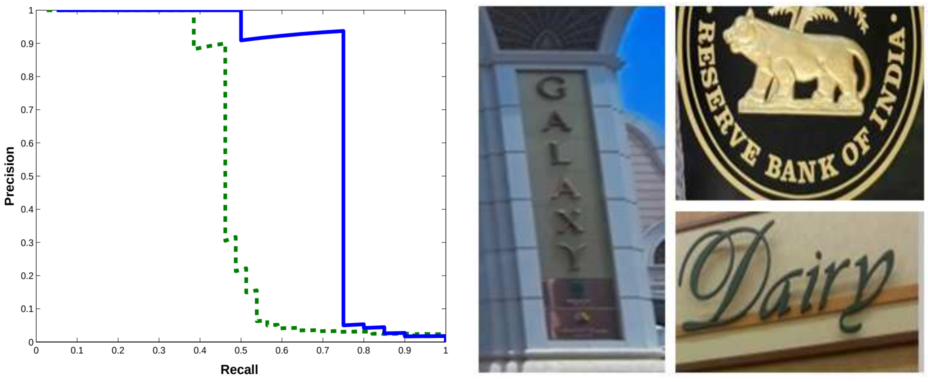


(a) Text query: *restaurant*

(b) Text query: *motel*

Figure 6.4 Text query example: Top-10 retrievals of our method on SVT and IIIT STR are shown. (a) Text query: “restaurant". There are in all 8 occurrences of this query in the SVT dataset. The proposed scheme retrieves them all. The ninth and the tenth results contain many characters from the query like R, E, S, T, A, N. (b) Text query: “motel". There are in all 39 occurrences of query in the IIIT STR dataset, with large variations in fonts, e.g. the first and the tenth retrievals. A failure case of our approach is when a highly similar word (hotel in this case) is well-ranked. These results support our claim of instance as well as category retrieval.

We also show result of our method on Sports-10K video dataset. These results are shown in Figure 6.6. We observe that our method successfully able to retrieve text appearing in wide variety of styles.



(a) (b)

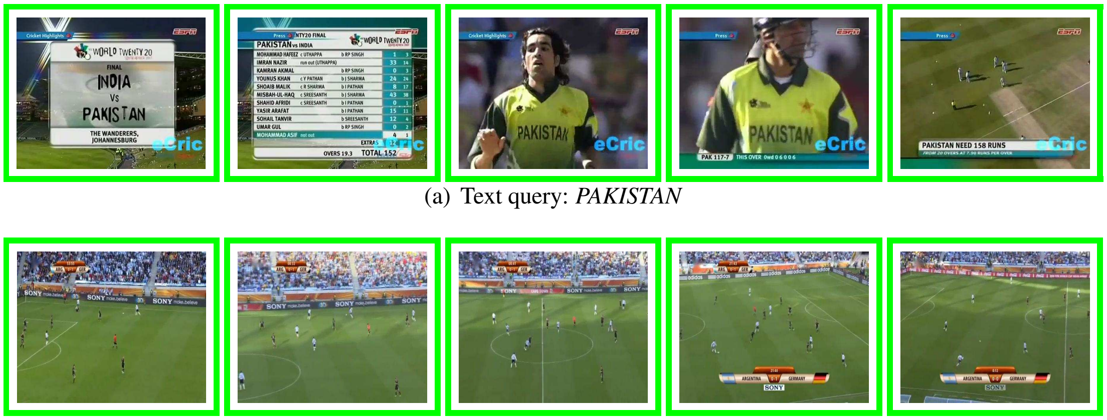
Figure 6.5 (a) Precision-Recall curves for two queries on the IIIT scene text retrieval dataset. The blue (solid) and green (dotted) curves correspond to queries “department" and “motel" respectively. (b) A few failure cases are shown as cropped images, where our approach fails to retrieve these images for the text queries: Galaxy, India and Dairy. The main reasons for failure are: the violation of near horizontal assumption for scene texts (in case of Galaxy and India), or a stylish font (Dairy).

6.5 Summary

We have demonstrated text-to-image retrieval based on the textual content present in images and videos. The query-driven approach we propose outperforms localization and recognition pipeline based methods [40,114]. The benefits of this work over methods based on a localization-recognition pipeline [40, 114] are: (i) It does not require explicit localization of the word boundary in an image. (ii) It is query-driven, thus even in cases where the second or the third best predictions for a character bounding box are correct, it can retrieve the correct result. We also showed that ensembelling the recent methods such as deep character classifier instead of hand crafted feature based character classifier improves the mAP.

The holistic recognition method presented in Chapter 5 can also be explored for text to image retrieval. However, there are two limitations with such an approach, (i) the DTW matching used in holistic recognition are computationally expensive and ill-suited for our large scale image retrieval system, and (ii) the holistic recognition requires accurate word bounding boxes. The method presented in this chapter works without explicit word localization.

There are few recent works where exact localization of objects (texts) is avoided [62, 123], rather a set of text proposals are obtained. We also adopted [62] for our datasets, and fed to modern scene



(a) Text query: *PAKISTAN*

(b) Text query: *SONY*

Figure 6.6 We show top-5 retrieval results of our method from the sports 10K video dataset for two queries (a) PAKISTAN (b) SONY.

text recognition engines (including our higher order CRF method). After obtaining the recognized texts we perform simple text based search. This simple method when tested on one of our datasets namely, Sports-10K achieves significantly high P@10 and P@20 (we obtain 65% and 60% of P@10 and P@20 from this method). Such method can also integrated with our query driven approach for even better retrieval performance in future.

To sum up, we have proposed a robust and scalable solution for text to image retrieval problem. Our method is effective in retrieving both category as well as instance of scene text. We have demonstrated our retrieval results on large-scale image and video datasets.

Chapter 7

Conclusion and Future Work

In this chapter we conclude this thesis by discussing the contributions, impact and comparisons of our proposed approaches with contemporary methods. Finally, we also provide the future directions of this dissertation.

7.1 Discussion

This thesis targets the problem of recognizing text in scene images and retrieving images or frames from a large database using textual cues.

We have presented three effective ways of scene text recognition. First, we presented recognition using robust segmentation (binarization). To this end, we have proposed a principled energy minimization based framework for scene text binarization. The proposed energy formulation uses the color and stroke features of text, and produces clean binary output. The proposed method in combination with the off-the-shelf open source OCR significantly improves the recognition performance on public benchmark datasets. Next, we presented a framework where we bypass hard segmentation and build a CRF model on potential character locations. We seamlessly integrated language model in terms of higher order priors, and efficiently minimize the corresponding energy function to recognize the words. We then presented a holistic framework for word spotting. In this, lexicon words are transformed into synthetic images and the problem of word spotting is posed as matching of scene text words and synthetic words. The proposed method achieves reasonably high performance on challenging benchmark datasets.

Going ahead, we have proposed a novel scheme for text-to-image retrieval. The benefit of our approach are two: (i) it does not require explicit localization of the word boundary in an image. (ii) It is query-driven, thus even in cases where the second or the third best predictions for a character bounding box are correct, it can retrieve the correct result. We have demonstrated our results on public scene text datasets as well as three large datasets, namely, IIIT scene text retrieval, Sports-10K and TV series-1M which we introduce.

Comparisons with contemporary methods. Our work proposed in Chapter 4 belongs to the class of word recognition methods which build on individual character localization, similar to methods such as [59, 112]. In this framework, the potential characters are localized, then a graph is constructed from these locations, and then the problem of recognizing the word is formulated as finding an optimal path in this graph [115] or inferring from an ensemble of HMMs [59]. Our approach shows a seamless integration of higher order language priors into the graph (in the form of a CRF model), and uses more effective modern computer vision features, thus making it clearly different from previous works.

Since the publication of our original work in CVPR 2012 [109] and BMVC 2012 [108] papers, several approaches for scene text understanding (e.g., text localization [60, 65, 104, 114], word recognition [22,65,67,143,168,172] and text-to-image retrieval [16,65,110,135]) have been proposed. Notably, there has been an increasing interest in exploring deep convolutional network based methods for scene text tasks (see [22, 60, 65, 67, 167] for example). These approaches are very effective in general, but the deep convolutional network, which is at the core of these approaches, lacks the capability to elegantly handle structured output data. To understand this with the help of an example, let us consider the problem of estimating human pose [158, 160], where the task is to predict the locations of human body joints such as head, shoulders, elbows and wrists. These locations are constrained by human body kinematics and in essence, form a structured output. To deal with such structured output data, state of the art deep learning algorithms include an additional regression step [160] or a graphical model [158], thus showing that these techniques are complementary to the deep learning philosophy. Similar to human pose, text is structured output data [64]. To better handle this structured data, we develop our energy minimization framework [108, 109] with the motivation of building a complementary approach, which can further benefit methods built on the deep learning paradigm. Indeed, we saw that combining the two frameworks further improves text recognition results (Chapter 4).

Further, our retrieval scheme outperforms localization and recognition pipeline based methods [40, 114]. We have also shown that mAP of our retrieval performance can further be improved with the integration of deep character classification [67] and recent technique of text proposal [62] followed by our recognition.

Impact of this thesis. This thesis advances the field of scene text understanding significantly. The proposed methods achieve noticeable improvements on word and isolated character recognition accuracies, and text binarization performance on multiple benchmark datasets. Many of the methods proposed in this thesis are based on principled frameworks with very minimal assumptions. This allows for the possibility of (i) improving the recognition performance further, and (ii) developing solutions for other languages such as Indian languages scene text recognition where research has not progressed much. The existing datasets related to scene text understanding were either too small or simple, as a part of this work we have introduced multiple scene text benchmark datasets. These datasets are being used by various groups across the globe [67, 133, 172, 174], and are driving research in this area.

7.2 Future directions

This thesis opens many promising avenues for future research as listed below:

- **Exploring CRF framework.** Our proposed higher order CRF framework (Chapter 4) seamlessly integrates multiple cues for word recognition. Some interesting techniques such as automatically learning the appropriate energy functions and the structure of the graph for the word image, are not explored in this thesis. These can be fascinating directions of research in the future.

- **Deep binarization.** There have been recent works on addressing the segmentation in the natural images in deep learning framework. Deep mask [123] is one of the most successful methods among those works. It has shown state of the art results on some image segmentation benchmarks. For a given image, deep mask produces, (i) class-agnostic segmentation mask, and (ii) likelihood of the patch being centered on a full object. We believe such likelihood scores can be integrated with our energy minimization framework for the problem of scene text binarization to further enhance the performance in the future.

- **Integrating deep bigrams and higher order grams classifier scores.** With the availability of immense data for training [66], it is now possible to train deep classifiers for bigrams, trigrams and higher order grams. Such classifier scores can be integrated into our higher order CRF framework (Chapter 4). Such integration will take advantage of both language models as well as appearance of n-grams in a real or synthetic data.

- **Improving efficiency of holistic recognition.** The holistic recognition method which we present in Chapter 5 shows promising performance in recognizing scene text. This method with some modifications can show big gain in recognition accuracy by taking advantage of availability of large synthetic word dataset [66]. One of the limitations of this method is the computational complexity of matching. Improving computation time of this method is one of the potential directions of research.

- **Extension to video.** The techniques proposed in this thesis are designed for recognition in images. Although we show retrieval performance on video datasets in Chapter 6, but treating each frame independently as image. It should be noted that videos have dependency among frames, for example, if a frame contain a word, the next frame is more likely to contain the same word in almost same location. Modeling such dependencies and using inherent spatio-temporal cues in an energy minimization framework is another possible way to develop the proposals of this thesis.

- **Multi-script scene text understanding.** In many countries like India, multiple scripts are used in various regions of the country. Robustly recognizing many of these scripts is still an open ended problem for printed text domain [54, 103, 129], whereas scene text recognition is even more challenging. The reader is encouraged to refer to one of our initial works [148] on this direction where we propose an end-to-end framework for script identification in scene images. Once scripts are identified, some of the techniques in this thesis as such or with some minor modifications can be applied for recognizing scene texts in multiple scripts.

- **Generating image annotations.** Text present in images gives an indispensable cue about the content of the images. Text understanding combined with the object understanding can tell a lot about the content of the image and can significantly improve image annotations. Moreover, questions such as “can text understanding improve object understanding?", and vice versa, are exciting to investigate and answer in future.

- **Integrating textual and visual cues for image retrieval.** In our work we have demonstrated image retrieval using textual cues, and have not explored the use of visual cues. Consider an example where we wish to retrieve all the ambulance images from a large image database. The use of text ambulance written on the vehicle along with visual features of the vehicle is likely to help retrieval system be more effective. Moreover, use of textual and visual features can also be useful in effectively retrieving movie posters, CD covers, etc.

Bibliography

[1] Tesseract OCR, http://code.google.com/p/tesseract-ocr/.

[2] Project website, http://cvit.iiit.ac.in/projects/SceneTextUnderstanding/.

[3] ICDAR 2003 datasets, http://algoval.essex.ac.uk/icdar.

[4] Street View Text dataset, http://vision.ucsd.edu/~kai/svt.

[5] Computer Blindness and Blondness, http://computerblindness.blogspot.in/2010_06_01_archive.html.

[6] H-DIBCO 2012, http://utopia.duth.gr/~ipratika/HDIBCO2012/.

[7] Robust word recognition dataset 2003. http://algoval.essex.ac.uk/icdar/RobustWord.html.

[8] Robust word recognition dataset 2011. http://www.cvc.uab.es/icdar2011competition/.

[9] ICDAR 2015 Competition on Video Script Identification, http://www.ict.griffith.edu.au/cvsi2015/.

[10] ABBYY Finereader 9.0, http://www.abbyy.com/.

[11] Robust word recognition dataset, http://algoval.essex.ac.uk/icdar/RobustWord.html.

[12] Natural Enviorment OCR. http://www.iapr-tc11.org/dataset/NEOCR/neocr_dataset.tar.gz.

[13] http://textspotter.org.

[14] http://www.eng.tau.ac.il/$\sim$talib/RBNR.html.

[15] J. Almazán, A. Gordo, A. Fornés, and E. Valveny. Efficient Exemplar Word Spotting. In *BMVC*, 2012.

[16] J. Almazán, A. Gordo, A. Fornés, and E. Valveny. Word Spotting and Recognition with Embedded Attributes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(12):2552–2566, 2014.

[17] I. Bazzi, R. M. Schwartz, and J. Makhoul. An omnifont open-vocabulary OCR system for english and arabic. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(6):495–504, 1999.

[18] R. Beaufort and C. Mancas-Thillou. A Weighted Finite-State Framework for Correcting Errors in Natural Scene OCR. In *ICDAR*, 2007.

[19] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, 1986.

[20] A. Bianne-Bernard, F. Menasri, R. A. Mohamad, C. Mokbel, C. Kermorvant, and L. Likforman-Sulem. Dynamic and Contextual Information in HMM Modeling for Handwritten Word Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(10):2066–2080, 2011.

[21] L. Bigorda and D. Karatzas. A Fast Hierarchical Method for Multi-script and Arbitrary Oriented Scene Text Extraction. *CoRR*, abs/1407.7504, 2014.

[22] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven. PhotoOCR: Reading Text in Uncontrolled Conditions. In *ICCV*, 2013.

[23] A. Blake, C. Rother, M. Brown, P. Pérez, and P. H. S. Torr. Interactive Image Segmentation Using an Adaptive GMMRF Model. In *ECCV*, 2004.

[24] E. Boros and P. L. Hammer. Pseudo-Boolean optimization. *Discrete Applied Mathematics*, 2002.

[25] Y. Boykov and M.-P. Jolly. Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images. In *ICCV*, 2001.

[26] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(9):1124–1137, 2004.

[27] Y. Boykov, O. Veksler, and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

[28] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*, 2008.

[29] D. Chen, J. Odobez, and H. Bourlard. Text detection, recognition in images and video frames. *Pattern Recognition*, 37(3):595–608, 2004.

[30] D. Chen, J. M. Odobez, and H. Bourlard. Text Segmentation and Recognition in Complex Background Based on Markov Random Field. In *ICPR*, 2002.

[31] X. Chen and A. L. Yuille. Detecting and Reading Text in Natural Scenes. In *CVPR*, 2004.

[32] A. Clavelli, D. Karatzas, and J. Lladós. A Framework for the Assessment of Text Extraction Algorithms on Complex Colour Images. In *DAS*, 2010.

[33] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu, and A. Y. Ng. Text Detection and Character Recognition in Scene Images with Unsupervised Feature Learning. In *ICDAR*, 2011.

[34] T. H. Cormen. *Introduction to Algorithms*. MIT press, 2009.

[35] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, 2005.

[36] T. E. de Campos, B. R. Babu, and M. Varma. Character Recognition in Natural Images. In *VISAPP*, 2009.

[37] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for multi-class object layout. In *ICCV*, 2009.

[38] K. Elagouni, C. Garcia, F. Mamalet, and P. Sébillot. Combining multi-scale character recognition and linguistic knowledge for natural scene text OCR. In *DAS*, 2012.

[39] K. Elagouni, C. Garcia, and P. Sébillot. A comprehensive neural-based approach for text recognition in videos using natural language processing. In *ICMR*, 2011.

[40] B. Epshtein, E. Ofek, and Y. Wexler. Detecting Text in Natural Scenes with Stroke Width Transform. In *CVPR*, 2010.

[41] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *Int'l J. Computer Vision*, 88(2):303–338, 2010.

[42] N. Ezaki, M. Bulacu, and L. Schomaker. Text Detection from Natural Scene Images: Towards a System for Visually Impaired Persons. In *ICPR*, 2004.

[43] J. Feild and E. Learned-miller. Scene Text Recognition with Bilateral Regression. Technical Report UM-CS-2012-021, University of Massachusetts-Amherst, Computer Science Research Center, 2013.

[44] J. L. Feild and E. G. Learned-Miller. Improving Open-Vocabulary Scene Text Recognition. In *ICDAR*, 2013.

[45] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part Based Models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, 2010.

[46] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, 2010.

[47] B. Gatos, K. Ntirogiannis, and I. Pratikakis. DIBCO 2009: document image binarization contest. *IJDAR*, 14(1):35–44, 2011.

[48] B. Gatos, I. Pratikakis, K. Kepene, and S. J. Perantonis. Text Detection in Indoor/Outdoor Scene Images. In *CBDAR*, 2005.

[49] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *CVPR*, 2012.

[50] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(6):721–741, 1984.

[51] V. Goel, A. Mishra, K. Alahari, and C. V. Jawahar. Whole is Greater than Sum of Parts: Recognizing Scene Text Words. In *ICDAR*, 2013.

[52] J. T. Goodman. A Bit of Progress in Language Modeling. Technical report, Microsoft Research, 2001.

[53] S. Gould, T. Gao, and D. Koller. Region-based Segmentation and Object Detection. In *NIPS*, 2009.

[54] V. Govindaraju and S. Setlur. *Guide to OCR for Indic Scripts: Document Recognition and Retrieval*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[55] S. M. Hanif and L. Prevost. Text Detection and Localization in Complex Scene Images using Constrained AdaBoost Algorithm. In *ICDAR*, 2009.

[56] T. Hong and J. J. Hull. Visual Inter-Word Relations and their Use in OCR Postprocessing. In *ICDAR*, 1995.

[57] N. R. Howe. A Laplacian Energy for Document Binarization. In *ICDAR*, 2011.

[58] N. R. Howe. Document binarization with automatic parameter tuning. *IJDAR*, 16(3):247–258, 2013.

[59] N. R. Howe, S. Feng, and R. Manmatha. Finding words in alphabet soup: Inference on freeform character recognition for historical scripts. *Pattern Recognition*, 42(12):3338–3347, 2009.

[60] W. Huang, Y. Qiao, and X. Tang. Robust scene text detection with convolution neural network induced mser trees. In *ECCV*, 2014.

[61] L. G. i Bigorda and D. Karatzas. Scene text recognition: No country for old men? In *ACCV Workshops*, 2014.

[62] L. G. i Bigorda and D. Karatzas. Object proposals for text extraction in the wild. In *ICDAR*, 2015.

[63] H. Ishikawa. Higher-order clique reduction in binary graph cut. In *CVPR*, 2009.

[64] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Deep structured output learning for unconstrained text recognition. *CoRR*, abs/1412.5903, 2014.

[65] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading Text in the Wild with Convolutional Neural Networks. *CoRR*, abs/1412.1842, 2014.

[66] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading Text in the Wild with Convolutional Neural Networks. *Int'l J. Computer Vision*, 116(1):1–20, 2016.

[67] M. Jaderberg, A. Vedaldi, and A. Zisserman. Deep Features for Text Spotting. In *ECCV*, 2014.

[68] C. V. Jawahar, B. Chennupati, B. Paluri, and N. Jammalamadaka. Video retrieval based on textual queries. In *ICACC*, 2005.

[69] S. Jegelka and J. Bilmes. Submodularity beyond submodular energies: Coupling edges in graph cuts. In *CVPR*, 2011.

[70] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to Predict Where Humans Look. In *ICCV*, 2009.

[71] K. Jung, K. I. Kim, and A. K. Jain. Text information extraction in images and video: a survey. *Pattern recognition*, 37(5):977–997, 2004.

[72] D. Karatzas and A. Antonacopoulos. Colour text segmentation in web images based on human perception. *Image Vision Comput.*, 25(5):564–577, 2007.

[73] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. Almazán, and L. de las Heras. ICDAR 2013 Robust Reading Competition. In *ICDAR*, 2013.

[74] T. Kasar, J. Kumar, and A. G. Ramakrishnan. Font and Background Color Independent Text Binarization. In *CBDAR*, 2007.

[75] S. M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. In *IEEE Trans. Acoustics, Speech and Singal processing*, volume ASSP-35, pages 400–401, 1987.

[76] K. Kita and T. Wakahara. Binarization of Color Characters in Scene Images Using k-means Clustering and Support Vector Machines. In *ICPR*, 2010.

[77] J. Kittler, J. Illingworth, and J. Föglein. Threshold Selection based on a Simple Image Statistic. *Computer Vision, Graphics, and Image Processing*, 1985.

[78] P. Kohli, M. P. Kumar, and P. H. S. Torr. P3 & Beyond: Solving Energies with Higher Order Cliques. In *CVPR*, 2007.

[79] P. Kohli, J. Rihan, M. Bray, and P. H. S. Torr. Simultaneous Segmentation and Pose Estimation of Humans Using Dynamic Graph Cuts. *Int'l J. Computer Vision*, 2008.

[80] V. Kolmogorov. Convergent Tree-Reweighted Message Passing for Energy Minimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(10):1568–1583, 2006.

[81] V. Kolmogorov and M. J. Wainwright. On the Optimality of Tree-reweighted Max-product Message-passing. In *UAI*, 2005.

[82] V. Kolmogorov and R. Zabih. What Energy Functions Can Be Minimized via Graph Cuts? *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2):147–159, 2004.

[83] A. Kornai. Language models: where are the bottlenecks? *AISB Quarterly*, 88:36–40, 1994.

[84] J. G. Kuk and N. I. Cho. Feature Based Binarization of Document Images Degraded by Uneven Light Condition. In *ICDAR*, 2009.

[85] D. Kumar, M. N. A. Prasad, and A. G. Ramakrishnan. Benchmarking Recognition Results on Camera Captured Word Image Data Sets. In *DAR*, 2012.

[86] D. Kumar, M. N. A. Prasad, and A. G. Ramakrishnan. MAPS: midline analysis and propagation of segmentation. In *ICVGIP*, 2012.

[87] D. Kumar, M. N. A. Prasad, and A. G. Ramakrishnan. NESP: Nonlinear enhancement and selection of plane for optimal segmentation and recognition of scene word images. In *DRR*, 2013.

[88] D. Kumar and A. G. Ramakrishnan. Power-law transformation for enhanced recognition of born-digital word images. In *SPCOM*, 2012.

[89] S. Kumar and M. Hebert. Discriminative Random Fields: A Discriminative Framework for Contextual Interaction in Classification. In *ICCV*, 2003.

[90] L. Ladicky, P. Sturgess, K. Alahari, C. Russell, and P. H. S. Torr. What, Where & How Many? Combining object detectors and CRFs. In *ECCV*, 2010.

[91] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *ICML*, 2001.

[92] G. Lazzara and T. Géraud. Efficient multiscale Sauvola's binarization. *IJDAR*, 2014.

[93] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[94] A. Levin and Y. Weiss. Learning to Combine Bottom-Up and Top-Down Segmentation. *Int'l J. Computer Vision*, 81(1):105–118, 2009.

[95] S. Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer Publishing Company, Incorporated, 3rd edition, 2009.

[96] D. P. Lopresti and J. Zhou. Locating and recognizing text in WWW images. *Inf. Retr.*, 2(2/3):177–206, 2000.

[97] S. Lu, B. Su, and C. L. Tan. Document image binarization using background estimation and stroke edges. *IJDAR*, 13(4):303–314, 2010.

[98] Z. Lu, Z. Wu, and M. S. Brown. Directed assistance for ink-bleed reduction in old documents. In *CVPR*, 2009.

[99] Z. Lu, Z. Wu, and M. S. Brown. Interactive degraded document binarization: An example (and case) for interactive computer vision. In *WACV*, 2009.

[100] S. Maji and J. Malik. Fast and accurate digit classification. *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-159*, 2009.

[101] C. Mancas-Thillou and B. Gosselin. Color Binarization for Complex Camera-based Images. In *Electronic Imaging Conference of the International Society for Optical Imaging*, 2005.

[102] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[103] M. Mathew, A. K. Singh, and C. V. Jawahar. Multilingual OCR for Indic Scrips. In *DAS*, 2016.

[104] S. Milyaev, O. Barinova, T. Novikova, P. Kohli, and V. S. Lempitsky. Image Binarization for End-to-End Text Understanding in Natural Images. In *ICDAR*, 2013.

[105] S. Milyaev, O. Barinova, T. Novikova, P. Kohli, and V. S. Lempitsky. Image binarization for end-to-end text understanding in natural images. In *ICDAR*, 2013.

[106] S. Milyaev, O. Barinova, T. Novikova, P. Kohli, and V. S. Lempitsky. Fast and accurate scene text understanding with image binarization and off-the-shelf OCR. *IJDAR*, 18(2):169–182, 2015.

[107] A. Mishra, K. Alahari, and C. V. Jawahar. An MRF Model for Binarization of Natural Scene Text. In *ICDAR*, 2011.

[108] A. Mishra, K. Alahari, and C. V. Jawahar. Scene Text Recognition using Higher Order Language Priors. In *BMVC*, 2012.

[109] A. Mishra, K. Alahari, and C. V. Jawahar. Top-Down and Bottom-Up Cues for Scene Text Recognition. In *CVPR*, 2012.

[110] A. Mishra, K. Alahari, and C. V. Jawahar. Image Retrieval using Textual Cues. In *ICCV*, 2013.

[111] M. Mozer, M. I. Jordan, and T. Petsche. Improving the Accuracy and Speed of Support Vector Machines. In *NIPS*, 1997.

[112] L. Neumann and J. Matas. A method for text localization and recognition in real-world images. In *ACCV*, 2010.

[113] L. Neumann and J. Matas. A Real-Time Scene Text to Speech System. In *ECCV workshops*, 2012.

[114] L. Neumann and J. Matas. Real-time scene text localization and recognition. In *CVPR*, 2012.

[115] L. Neumann and J. Matas. On Combining Multiple Segmentations in Scene Text Recognition. In *ICDAR*, 2013.

[116] W. Niblack. *An introduction to Digital Image Processing*. Prentice Hall, 1986.

[117] T. Novikova, O. Barinova, P. Kohli, and V. S. Lempitsky. Large-Lexicon Attribute-Consistent Text Recognition in Natural Images. In *ECCV*, 2012.

[118] T. Novikova, O. Barinova, P. Kohli, and V. S. Lempitsky. Large-Lexicon Attribute-Consistent Text Recognition in Natural Images. In *ECCV*, 2012.

[119] N. Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

[120] J. Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kauffman, 1988.

[121] X. Peng, S. Setlur, V. Govindaraju, and R. Sitaram. Markov Random Field based Binarization for Handheld Devices Captured Document Images. In *ICVGIP*, 2010.

[122] T. Q. Phan, P. Shivakumara, B. Su, and C. L. Tan. A Gradient Vector Flow-Based Method for Video Character Segmentation. In *ICDAR*, 2011.

[123] P. O. Pinheiro, R. Collobert, and P. Dollár. Learning to Segment Object Candidates. *CoRR*, abs/1506.06204, 2015.

[124] J. C. Platt. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In *Advances in Large Margin Classifiers*. MIT Press, 1999.

[125] N. Ponomareva, P. Rosso, F. Pla, and A. Molina. Conditional random fields vs. hidden markov models in a biomedical named entity recognition task. In *RANLP*, 2007.

[126] I. Pratikakis, B. Gatos, and K. Ntirogiannis. ICFHR 2012 Competition on Handwritten Document Image Binarization (H-DIBCO 2012). In *ICFHR*, 2012.

[127] S. Ramalingam, P. Kohli, K. Alahari, and P. H. S. Torr. Exact inference in multi-label CRFs with higher order cliques. In *CVPR*, 2008.

[128] T. M. Rath and R. Manmatha. Word Image Matching Using Dynamic Time Warping. In *CVPR*, 2003.

[129] A. Ray, S. Rajeswar, and S. Chaudhury. OCR for bilingual documents using language modeling. In *ICDAR*, 2015.

[130] K. Rayner and A. Pollatsek. *The psychology of reading*. Routledge, 1989.

[131] X. Ren and D. Ramanan. Histograms of Sparse Codes for Object Detection. In *CVPR*, 2013.

[132] E. M. Riseman and A. R. Hanson. A Contextual Postprocessing System for Error Correction Using Binary n-Grams. *IEEE Trans. Computers*, 23(5):480–493, 1974.

[133] J. Rodriguez and F. Perronnin. Label embedding for text recognition. In *BMVC*, 2013.

[134] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

[135] U. Roy, A. Mishra, K. Alahari, and C. V. Jawahar. Scene Text Recognition and Retrieval for Large Lexicons. In *ACCV*, 2014.

[136] C. Russell, L. Ladicky, P. Kohli, and P. H. S. Torr. Exact and Approximate Inference in Associative Hierarchical Networks using Graph Cuts. In *UAI*, 2010.

[137] D. Sankoff and J. Kruskal. *Time warps, string edits, and macromolecules: the theory and practice of sequence comparison*. Addison-Wesley, 1983.

[138] J. J. Sauvola and M. Pietikäinen. Adaptive document image binarization. *Pattern Recognition*, 33(2):225–236, 2000.

[139] D. Schlesinger and B. Flach. Transforming an arbitrary minsum problem into a binary one. Technical report, Tech Report, 2006.

[140] G. Scot and H. Loguet-Higgins. An algorithm for associating the features of two patterns. *Proc. Of the Royal Society of London B*, 224:21–26, 1991.

[141] A. Shahab, F. Shafait, and A. Dengel. ICDAR 2011 Robust Reading Competition Challenge 2: Reading Text in Scene Images. In *ICDAR*, 2011.

[142] K. Sheshadri and S. K. Divvala. Exemplar Driven Character Recognition in the Wild. In *BMVC*, 2012.

[143] C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, and Z. Zhang. Scene Text Recognition Using Part-Based Tree-Structured Character Detection. In *CVPR*, 2013.

[144] P. Shivakumara, S. Bhowmick, B. Su, C. L. Tan, and U. Pal. A New Gradient Based Character Segmentation Method for Video Text Recognition. In *ICDAR*, 2011.

[145] P. Shivakumara, T. Q. Phan, and C. L. Tan. A Laplacian Approach to Multi-Oriented Text Detection in Video. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(2):412–419, 2011.

[146] J. Shotton, J. Winn, C. Rother, and A. Criminisi. TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context. *Int'l J. Computer Vision*, 81(1):2–23, 2009.

[147] P. Simard, B. Victorri, Y. LeCun, and J. S. Denker. Tangent Prop - A Formalism for Specifying Selected Invariances in an Adaptive Network. In *NIPS*, 1991.

[148] A. K. Singh, A. Mishra, P. Dabaral, and C. V. Jawahar. A Simple and Effective method for Script Identification in the Wild. In *DAS*, 2016.

[149] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[150] E. H. B. Smith. An Analysis of Binarization Ground Truthing. In *DAS*, 2010.

[151] R. Smith. Limits on the Application of Frequency-based Language Models to OCR. In *ICDAR*, 2011.

[152] P. Stathis, E. Kavallieratou, and N. Papamarkos. An Evaluation Technique for Binarization Algorithms. *Universal Computer Science*, 2008.

[153] B. Su and S. Lu. Accurate Scene Text Recognition Based on Recurrent Neural Network. In *ACCV*, 2014.

[154] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. F. Tappen, and C. Rother. A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(6):1068–1080, 2008.

[155] B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin. Learning structured prediction models: A large margin approach. In *ICML*, 2005.

[156] C. Thillou, S. Ferreira, and B. Gosselin. An Embedded Application for Degraded Text Recognition. *EURASIP J. Adv. Sig. Proc.*, 2005(13):2127–2135, 2005.

[157] S. Tian, S. Lu, B. Su, and C. L. Tan. Scene text segmentation with multi-level maximally stable extremal regions. In *ICPR*, 2014.

[158] J. J. Tompson, A. Jain, Y. Lecun, and C. Bregler. Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation. In *NIPS*, 2014.

[159] X. Tong and D. A. Evans. A statistical approach to automatic OCR error correction in context. In *Workshop on very large corpora*, 1996.

[160] A. Toshev and C. Szegedy. DeepPose: Human Pose Estimation via Deep Neural Networks. In *CVPR*, pages 1653–1660, 2014.

[161] M. Valizadeh and E. Kabir. Binarization of degraded document image based on feature space partitioning and classification. *IJDAR*, 15(1):57–69, 2012.

[162] A. Vedaldi and A. Zisserman. Efficient Additive Kernels via Explicit Feature Maps. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(3):480–492, 2012.

[163] A. Vinciarelli. A survey on off-line cursive word recognition. *Pattern recognition*, 35(7):1433–1446, 2002.

[164] P. A. Viola and M. J. Jones. Robust Real-Time Face Detection. *Int'l J. Computer Vision*, 57(2):137–154, 2004.

[165] K. Wang, B. Babenko, and S. Belongie. End-to-End Scene Text recognition. In *ICCV*, 2011.

[166] K. Wang and S. Belongie. Word Spotting in the Wild. In *ECCV*, 2010.

[167] T. Wang, D. Wu, A. Coates, and A. Ng. End-to-End Text Recognition with Convolutional Neural Networks. In *ICPR*, 2012.

[168] J. Weinman, Z. Butler, D. Knoll, and J. Feild. Toward Integrated Scene Text Reading. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(2):375–387, 2014.

[169] J. J. Weinman, E. G. Learned-Miller, and A. R. Hanson. A discriminative semi-Markov model for robust Scene Text Recognition. In *ICPR*, 2008.

[170] J. J. Weinman, E. G. Learned-Miller, and A. R. Hanson. Scene Text Recognition Using Similarity and a Lexicon with Sparse Belief Propagation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(10):1733–1746, 2009.

[171] C. Wolf and D. S. Doermann. Binarization of Low Quality Text Using a Markov Random Field Model. In *ICPR*, 2002.

[172] C. Yao, X. Bai, B. Shi, and W. Liu. Strokelets: A Learned Multi-scale Representation for Scene Text Recognition. In *CVPR*, 2014.

[173] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *CVPR*, 2012.

[174] Q. Ye and D. Doermann. Text Detection and Recognition in Imagery: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 99:1–20, 2014.

[175] J. S. Yedidia, W. T. Freeman, and Y. Weiss. *Understanding belief propagation and its generalizations*, pages 239–269. Morgan Kaufmann Publishers Inc., 2003.

[176] C. Yi, X. Yang, and Y. Tian. Feature Representations for Scene Text Character Recognition: A Comparative Study. In *ICDAR*, 2013.

[177] D. Zhang, D. Wang, and H. Wang. Scene text recognition using sparse coding based features. In *ICIP*, 2014.

[178] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional Random Fields as Recurrent Neural Networks. In *ICCV*, 2015.

[179] C. L. Zitnick and P. Dollár. Edge Boxes: Locating Object Proposals from Edges. In *ECCV*, 2014.