Methods for text segmentation from scene images

A Thesis

Submitted for the Degree of

Doctor of Philosophy

in the Faculty of Engineering

by

Deepak Kumar

Department of Electrical Engineering
INDIAN INSTITUTE OF SCIENCE
BANGALORE – 560 012, INDIA

JANUARY 2014

ಅರ್ಪಣೆ

ಶ್ರೀಮತಿ ಕಮಲಮ್ಮ

ಮತ್ತು

ಶ್ರೀ ಪುಟ್ಟೇ ಗೌಡ

Acknowledgements

I consider it a privilege to express my sincere gratitude and respect to all those who guided and inspired me. I express my special thanks to my guide, Prof. A. G. Ramakrishnan, for his encouragement, inspiration, support and guidance throughout my research work.

I express my thanks to Prof. P. S. Sastry, Prof. K. R. Ramakrishnan and Prof. M. Narasimha Murty, for their inspiration, support and guidance throughout my course work.

I express my thanks to the research students of MILE laboratory, Thotreingam Kasar, Suresh Sundaram, Shiva Kumar H. R. and M. N. Anil Prasad, for their support throughout my research work.

I express my thanks to the members of MILE laboratory, Shanti Devaraj, Shanti S. and Saraswathi S., for annotating the images that are used in the experiments. I express my thanks to Technology Development for Indian Languages (TDIL), Department of Information Technology, Government of India for funding the above staff.

I express my thanks to my sisters, Sowmya Shree and Bhanu Shree, for their support throughout my research work.

I express my thanks to Prof. K. Sankara Rao for his feedback on the initial draft of my thesis.

I thank everyone, who have been directly or indirectly supportive during the course of my research work.

Abstract

Recognition of text from camera-captured scene/born-digital images help in the development of aids for the blind, unmanned navigation systems and spam filters. However, text in such images is not confined to any page layout, and its location within in the image is random in nature. In addition, motion blur, non-uniform illumination, skew, occlusion and scale-based degradations increase the complexity in locating and recognizing the text in a scene/born-digital image.

Text localization and segmentation techniques are proposed for the born-digital image data set. The proposed OTCYMIST technique won the first place and placed in the third position for its performance on the text segmentation task in ICDAR 2011 and ICDAR 2013 robust reading competitions for born-digital image data set, respectively. Here, Otsu's binarization and Canny edge detection are separately carried out on the three colour planes of the image. Connected components (CC's) obtained from the segmented image are pruned based on thresholds applied on their area and aspect ratio. CC's with sufficient edge pixels are retained. The centroids of the individual CC's are used as nodes of a graph. A minimum spanning tree is built using these nodes of the graph. Long edges are broken from the minimum spanning tree of the graph. Pairwise height ratio is used to remove likely non-text components. CC's are grouped based on their proximity in the horizontal direction to generate bounding boxes (BB's) of text strings. Overlapping BB's are removed using an overlap area threshold. Non-overlapping and minimally overlapping BB's are used for text segmentation. These BB's are split vertically to localize text at the word level.

A word cropped from a document image can easily be recognized using a traditional optical character recognition (OCR) engine. However, recognizing a word, obtained by manually cropping a scene/born-digital image, is not trivial. Existing OCR engines do not handle these kinds of scene word images effectively. Our intention is to first segment the word image and then pass it to the existing OCR engines for recognition. In two aspects, it is advantageous: it avoids building a character classifier from scratch and reduces the word recognition task to a word segmentation task. Here, we propose two bottom-up approaches for the task of word segmentation. These approaches choose different features at the initial stage of segmentation.

Power-law transform (PLT) was applied to the pixels of the gray scale born-digital images to non-linearly modify the histogram. The recognition rate achieved on born-digital word images is 82.9%, which is 20% more than the top performing entry (61.5%) in ICDAR 2011 robust reading competition. In addition, we explored applying PLT to the colour planes such as red, green, blue, intensity and lightness plane by varying the gamma value. We call this technique as Nonlinear enhancement and selection of plane (NESP) for optimal segmentation, which is an improvement over PLT. NESP chooses a particular plane with a proper gamma value based on Fisher discrimination factor. The recognition rate is 72.8% for scene images of ICDAR 2011 robust reading competition, which is 30% higher than the best entry (41.2%). The recognition rate is 81.7% and 65.9% for born-digital and scene images of ICDAR 2013 robust reading competition, respectively, using NESP.

Another technique, midline analysis and propagation of segmentation (MAPS), has also been proposed. Here, the middle row pixels of the gray scale image are first segmented and the statistics of the segmented pixels are used to assign text and non-text labels to the rest of the image pixels using min-cut method. Gaussian model is fitted on the middle row segmented pixels before the assignment of other pixels. In MAPS, we assume the middle row pixels are least affected by any of the degradations. This assumption is validated by the good word recognition rate of 71.7% on ICDAR 2011 robust reading competition for scene images. The recognition rate is 83.8% and 66.0% for born-digital and scene

images of ICDAR 2013 robust reading competition, respectively, using MAPS. The best reported results for ICDAR 2003 word images is 61.1% using custom lexicons containing the list of test words. On the other hand, NESP and MAPS achieve 66.2% and 64.5% for ICDAR 2003 word images without using any lexicon. By using similar custom lexicon, the recognition rates for ICDAR 2003 word images go up to 74.9% and 74.2% for NESP and MAPS methods, respectively.

In place of passing an image segmented by a method, manually segmented word image is submitted to an OCR engine for benchmarking maximum possible recognition rate for each database. The recognition rates of the proposed methods and the benchmark results are reported on the seven publicly available word image data sets and compared with these of reported results in the literature.

Since no good Kannada OCR is available, a classifier is designed to recognize Kannada characters and words from Chars74k data set and our own image collection, respectively. Discrete cosine transform (DCT) and block DCT are used as features to train separate classifiers. Kannada words are segmented using the same techniques (MAPS and NESP) and further segmented into groups of components, since a Kannada character may be represented by a single component or a group of components in an image. The recognition rate on Kannada words is reported for different features with and without the use of a lexicon. The obtained recognition performance for Kannada character recognition (11.4%) is three times the best performance (3.5%) reported in the literature.

Contents

List of Tables

List of Figures

Notation and Abbreviations

$f,\ f_{in},\ f_{out}$ – Pixel values of an image

$x,\ y$ – Pixel coordinates

$\rho,\ \theta$ – Pixel polar coordinates

$w,\ w_c$ – Weight

$\mu,\ \mu_c,\ \mu_{ct},\ \mu_T$ – Mean

$\sigma_B^2,\ \sigma^2$ – Variance

T – Threshold

h – Hypothesis

$p,\ P_{OB},\ \overline{P}_{OB}$ – Precision

$r,\ R_{OB},\ \overline{R}_{OB}$ – Recall

t_r, t_p – Recall and Precision threshold

V_{ij} – Interaction potential

C_{nm} – Cosine function

V_{nm} – Angular and radial basis function

A_m – Angular basis function

R_n – Radial basis function

GT_i – Fraction of ground-truth pixels in a scene word image

B_i – Benchmark value of a scene image

AR_{ji} – Algorithm Result of j^{th} algorithm on i^{th} image

AS_j – Algorithm Strength

ART: Angular Radial Transform

BDI: Born-Digital Image

Block DCT: 8x8 Block Discrete Cosine Transform

CC: Connected Component

DCT: Discrete Cosine Transform

DIBCO: Document Image Binarization Contest

HOG: Histogram of Oriented Gradients

ICDAR: International Conference on Document Analysis and Recognition

IJDAR: International Journal on Document Analysis and Recognition

JPEG: Joint Photographic Experts Group

Lex: Lexicon

MAPS: Midline Analysis and Propagation of Segmentation

MAST: Multi-script Annotation toolkit for Scene Text

MAST-CH: Multi-script Annotation toolkit-character-level

MASTER: Multi-script and scene text reading

MILE: Medical Intelligence and Language Engineering Laboratory

MPEG: Moving Picture Experts Group

MKL: Multiple Kernel Learning

MRRC: Multi-script Robust Reading Competition

NESP: Non-linear enhancement and selection of plane

OCR: Optical Character Recognition

OTCYMIST: Otsu–Canny Minimum Spanning Tree

PAMI: Pattern Analysis and Machine Intelligence

PLT: Power-law transform

Q-Q: Quantile-Quantile

SIFT: Scale-Invariant Feature Transform

SVT: Street View Text

Chapter 1

Scene text

Summary

Text detection in scene images is a complex task. Even when text exists in a scene image, its localization is not trivial. Apart from the text localization problem, recognition of the individual characters and the word pose additional problems. This chapter surveys the literature and presents the structure of the thesis.

1.1 Introduction

The field of document analysis and recognition has a history of more than one hundred years. The time itself indicates the complexity involved with the problems in the field, and several researchers have contributed to the progress in the field. Optophone instrument [78] was invented by Dr. E. E. Fournier d'Albe of Birmingham University in 1912, using optics to read characters and convert it to a phone, finally to help blind people. However, it still remains as an unsolved complex problem in terms of matching the ability of a human being.

Document analysis and recognition has expanded rapidly, adding new sub-field with each invention in the process of evolution. The major inventions that added new problems to the field are printing, photocopying, scanning and image capturing technologies.

If you wish to have a digital copy of a published article in a newspaper, then you

capture the QR code of that article present on the paper and provide it to a smart phone application available from the newspaper publisher. The application will fetch the digital copy of that article with additional information into your smart phone. This is a sample application in the field of document analysis and recognition. There are a large number of other applications that have been developed by researchers, with various advancements in technology. However, every research in the field may not lead to an application.

Photocopying, a Xerox invention, revolutionized the field of documents [87]. However, the idea of completely recognizing the content of a document is amazing and is a task still difficult to achieve by a machine, unlike the human brain. The basic version of a photocopying machine does not possess this capability. The present day optical character recognition (OCR) engines reasonably perform this task by creating a soft copy of the document.

A computer, an achievement of mankind to perform computation, which is not possible by a single person or a group, has ability to process and store textual information (Roman characters) in bits. If judiciously managed, textual information stored in a computer can have a larger life expectancy than printed material as in books, journals and magazines. The scanner was invented to convert printed, handwritten and historical documents into digital format. This triggered the conversion process in archives. A single scanned image consumes a large storage space due to the required scanning resolution. The ratio of storage space consumed by a scanned image to that of the same information stored as coded text is huge. This motivated researchers to work on methods to recognize the textual content of a scanned image. Thus, the field of pattern recognition taught the complexities to recognize Roman characters. As mentioned earlier, the thought of recognizing the content created a breakthrough in the form of document/character recognition.

The explosion of digital documents gave birth to the analysis of actual content in digital documents such as information retrieval, text or data mining (as search), text summarization (as knowledge) and relation between documents (as relational database management systems). Machine learning approaches have been incorporated to mimic the ability of humans.

Figure 1.1: Sample scene images from ICDAR 2003 robust reading competition text localization data set with non-uniform illumination, perspective deformation and motion blur.

The availability of low-cost cameras and mobile phones with a camera, created camera-captured documents. They overcome the limitations of a scanner and also increased the range of documents. An example for the limitation of a scanner is imaging the text on a notice board with a glass frame. On the other hand, the mobility of cameras make it feasible to capture the contents of a notice board from any reasonable distance. Other examples of camera-captured documents are those of billboards and signboards.

Generally a camera-captured image is likely to cover more of scene than text. In fact, textual information may or may not be present in many scenes. Detecting the presence or absence of text in a scene image is known as the text detection problem. It is almost equivalent to asking a blind person to analyze the scene. In order to reduce the complexity of text detection, the problem itself was broken down into parts such as text localization (locating the text) and word recognition (recognizing the individual words). Only scene images with textual information, which are known as scene text images, are analyzed for text localization and word recognition. Sample billboard, signboard and notice board images captured with the scene are shown in Figure 1.1.

Image editing softwares are used for enhancing, filtering and reconstruction of images. These softwares are used ubiquitously to create banners, logos, sign-ages and advertisements. Here, images are modified or artificially created and are known as born-digital images. Sample born-digital images are shown in Figure 1.2. In this thesis, methods are developed to tackle the problems involved in the analysis of scene/born-digital text images.

Figure 1.2: Sample born-digital images from ICDAR 2011 competition data set.

1.2 Page layout analysis

Scanned or camera-captured documents have a definite underlying format/layout. The segmentation of these images into text and non-text blocks is known as page layout analysis. Every book, magazine or journal has its own page layout. In particular, magazines have varying page layouts mixing graphics and text. Non-uniform illumination within a scanner degrades a document image. Page curl, skew, occlusion, motion blur, gloss (due to glass frame) or non-uniform illumination appear as degradations in a camera-captured document image. A method should be efficient to handle the different kinds of page layouts along with these degradations. Page layout analysis is important to analyze the structure of a document, which provides the ability to segment the text blocks and recognize the characters.

1.3 Text localization

Text localization is an easier task on document images, since the underlying structure can be analyzed using a page layout algorithm. In a scene/born-digital image (hereafter, only scene/born-digital text images are considered), there does not exist any structure to perform page layout analysis. Thus, the task of locating text and placing a bounding box around each word in the image, becomes complex. A camera-captured/born-digital image may have randomly positioned text. A text localization mask and a bounding box around each word are shown on a scene image in Figure 1.3. To match human capability, a text localization method needs to locate randomly placed text in any image with degradations.

Figure 1.3: Illustration of the distinction between text localization and segmentation tasks. (a) A sample scene image from ICDAR 2003 data set. (b) Text localization mask. (c) Red coloured bounding box placed around each word. (d) Segmented text.

Since the resolution of text varies from image to image, it is also important and acts as degradation based on scale, apart from other degradations.

1.4 Text segmentation

Text segmentation is the process of segmenting/extracting the text pixels from a scene or born-digital image. This is similar to text localization task, but differs in locating the text at the pixel-level. Segmented text (ground-truth) is shown in Figure 1.3(d) for a sample image. A method can be developed independently to address the text segmentation or localization problem or both. Text localization results in non-text pixels within a located bounding box, but the text segmentation task targets in segregating the text pixels only.

1.5 Recognition of scene word images

Locating the text or a word in a scene/born-digital image is complex. This complexity is totally skipped in word recognition tasks by manually cropping the word from a scene/born-digital image. However, the degradation which exists in a scene/born-digital image cannot be avoided in cropped images. Though the complexity is reduced in the amount of non-text pixels compared to its parent image, it is not completely removed. A few samples of cropped word images are shown in Figure 1.4. Here, the recognition or segmentation may be used to aid the other task. Thus, a better segmented word image yields better recognition (bottom-up/feed-forward approach) and better recognition may

Figure 1.4: Sample cropped word images from ICDAR 2011 competition data set.

Figure 1.5: Kannada and English character image samples from Chars74k dataset [77].

be used to improve segmentation (top-down/feedback approach).

1.6 Recognition of characters cropped from scene images

Scene images are cropped manually at the character level. Each character image has degradation similar to that encountered in a generic scene image. Some samples of manually cropped characters are shown in Figure 1.5. The intention behind manually cropping the characters is to train a supervised classifier. An enormous number of training samples is required to cover the different types of variations of characters that occur in a scene image. It is possible, but not advisable, since segmenting a character and choosing a nearest training sample for recognition is easier for a technique.

1.7 Handwritten characters

Handwritten characters also may appear in a scene image. Similarly, artistic characters are used to design banners which appear in a born-digital image. A few images with

Figure 1.6: Sample scene images with handwritten text and artistic font from ICDAR 2003 competition data set.

handwritten and artistic characters are shown in Figure 1.6. Handwriting recognition is classified into two categories based on the kind of information used in the training stage. If temporal information is used to train a classifier, then the recognition is known as on-line handwriting recognition; else off-line handwriting recognition.

1.8 Robust reading competitions

Several methods have been published in reputed journals and conferences to tackle sub-problems in the field. How does one choose one method over another based on the requirement of an application? How do we know which method is superior to another? Hence, competitions are organized to determine state-of-the-art algorithms for a particular problem with the same data set.

In early 1990s, Roman character recognition competitions were organized. The document images had a simple page layout. Thus, the page layout analysis is avoided and an algorithm had to recognize the characters by segmenting the text blocks into text lines and further, the words. This character recognition competition was continued till late 1990s [72].

By the year 2000, character recognition had reached more than 95% for different methods. Thus, since 2000, page segmentation competitions have been organized [5]. This competition involves different types of page layout that are commonly observed in published books, conferences, journals, magazines and newspapers. An algorithm needs to effectively segment the document into text and non-text blocks. This competition is

now regularly conducted in ICDAR conferences on different kinds of document images.

In ICDAR 2003, a new type of competition known as 'robust reading competition' on scene text was organized. This competition had three tasks, namely text localization, word recognition and character recognition. Only text localization task received entries and surprisingly there were no entries for word or character recognition tasks. In ICDAR 2005, only text localization competition was organized [22]. The state-of-the art algorithms on ICDAR 2005 dataset use edge information to locate text in images [11, 15]. Robust reading competition was not conducted in ICDAR 2007 and 2009. It got restarted in ICDAR 2011 with two challenges: one was on scene images and the other was on born-digital images. In ICDAR 2013, this competition has also included text localization challenge in videos [24]. ICDAR robust reading datasets are publicly available after the respective competitions as IAPR TC11 Reading Systems-Datasets [21].

1.8.1 Performance evaluation

This thesis is devoted to scene/born-digital text analysis. The evaluation procedures used in ICDAR robust reading competitions are described below. These procedures are also used in the thesis to evaluate the developed methods.

Text localization task

The evaluation procedure for text localization task cannot stop with simple area-overlap-based precision and recall measures. Lucas et.al proposed a way to measure the precision and recall for each word bounding box [52]. The final score is obtained by matching one-to-one the bounding boxes obtained by the method with those of the ground-truth. Precision and recall values are computed as [52]:

$$p = \frac{\sum_{r_e \in E} m(r_e,\ T)}{|E|} \tag{1.1}$$

$$r = \frac{\sum_{r_t \in T} m(r_t,\ E)}{|T|} \tag{1.2}$$

where, T and E are the sets of target (r_t) and estimated (r_e) rectangles, respectively. $m(r_e, T)$ or $m(r_t, E)$ is the best match for a rectangle r_e or r_t in a set of rectangles T or E, respectively. $m(r_e, T)$ and $m(r_t, E)$ are computed as:

$$m(r_e,\ T) = \max m_p(r_e,\ r_r) \mid r_r \in T \tag{1.3}$$

$$m(r_t,\ E) = \max m_p(r_t,\ r_r) \mid r_r \in E \tag{1.4}$$

The stricter norms of one-to-one matching in the performance evaluation are used to penalize methods that detect a text-line rather than the individual words. Wolf and Jolion [86] proposed another evaluation procedure based on the object count. Apart from one-to-one match, they considered one-to-many and many-to-one matches with little penalty. Each bounding box that results from a method is evaluated for precision and recall at the bounding box level according to the expressions given below [86]:

$$R_{OB}(G,\ D,\ t_r,\ t_p) = \frac{\sum_i Match_G(G_i,\ D,\ t_r,\ t_p)}{|G|} \tag{1.5}$$

$$P_{OB}(G,\ D,\ t_r,\ t_p) = \frac{\sum_j Match_D(D_j,\ G,\ t_r,\ t_p)}{|D|} \tag{1.6}$$

where, G and D are the sets of ground-truth (G_i) and detected (D_j) rectangles, respectively. $Match_G$ and $Match_D$ are match functions calculated from area overlap measures; t_r and t_p are thresholds for recall and precision, respectively.

The precision and recall values of a method for the entire data set is obtained by summing up the precision and recall values of the individual bounding boxes, and normalizing by the total number of bounding boxes resulting from a method and present in the ground-truth, respectively. The recall and precision values for a data set of N images

are computed as:

$$\overline{R}_{OB}(\overline{G},\ \overline{D},\ t_r,\ t_p) = \frac{\sum_k \sum_i Match_G(G_i^k,\ D^k,\ t_r,\ t_p)}{\sum_k |G^k|}, G^k \in \overline{G}, k = 1, ..., N \quad (1.7)$$

$$\overline{P}_{OB}(\overline{G},\ \overline{D},\ t_r,\ t_p) = \frac{\sum_k \sum_j Match_D(D_j^k,\ G^k,\ t_r,\ t_p)}{\sum_k |D^k|}, D^k \in \overline{D}, k = 1, ..., N \quad (1.8)$$

Text segmentation task

The text segmentation task was introduced as a separate aspect of ICDAR 2011 robust reading competition. However, it existed earlier for scanned documents as Document Image Binarization Contest (DIBCO) [16]. The evaluation procedure has been to calculate the precision, recall and f-score values. The precision and recall values are calculated between the result of the method and the ground-truth for each image at the pixel-level. The average values of precision and recall for the data set are used to rank the method with harmonic mean as the f-score.

Clavelli et. al [12] introduced another evaluation procedure for segmented images at connected component (CC) level. In this approach, each ground-truth CC is compared with the output of the algorithm to determine whether the component is well-segmented, merged, broken or lost and classified into their respective group. The number of components in each group are normalized for each image by dividing by the total number of components in the ground-truth image. Then, the average value is obtained from these normalized values for the entire data set to rank the algorithm.

Character recognition task

The character classification accuracy is calculated by dividing the number of correctly recognized characters by the total number of characters. This procedure is followed in character recognition competitions. Manually cropped characters were evaluated using this procedure in Chars74k data set [77].

Word recognition task

The word recognition accuracy is calculated by dividing the number of correctly recognized words by the total number of words in the data set. Another parameter known as Levenshtein edit distance [49] is used for the evaluation apart from the word recognition accuracy. The edit distance is calculated between the recognized and the ground-truth word and normalized by the number of characters in the ground-truth word. All the normalized edit distances are summed up to obtain the total edit distance for the data set. The edit distance measure was introduced in ICDAR 2011 robust reading competition for the word recognition task.

1.9 Structure of the thesis

Chapter 2 describes Otsu–Canny Minimum Spanning Tree (OTCYMIST) method, proposed for text segmentation task, which participated in ICDAR 2011 Robust reading competition challenge 1. Born-digital images were used for the competition. Text localization task was performed by grouping the connected components, resulting from the OTCYMIST method. This method won the first place for its performance on text segmentation task. ICDAR 2011 robust reading competition on scene images did not have text segmentation task. Now, ICDAR 2013 robust reading competition has included text segmentation task for both scene and born-digital images. OTCYMIST method was placed in the third position for its performance on text segmentation task in ICDAR 2013 robust reading competition for born-digital images.

Chapter 3 describes Power-law transform (PLT), Non-linear enhancement and selection of plane (NESP) and Midline analysis and propagation of segmentation (MAPS) methods, developed for pixel level segmentation of manually cropped word images. These methods were all developed post ICDAR 2011 period and have statistically different approaches. The recognition rates achieved by the methods are state-of-the-art on publicly available word image data sets. All these methods were positioned in the second and the third places for their performance on word recognition task in ICDAR 2013 robust

reading competition for both scene and born-digital images.

Chapter 4 describes the recognition of English and Kannada characters and words. MILE Kannada OCR samples were used for training the classifier. A word recognizer is developed for Kannada word images. This work is first of its kind in reporting on scene word recognition in Kannada language.

Appendix A describes an annotation tool known as 'Multi-script annotation toolkit for scene text' (MAST) and creation of 'Multi-script and scene text reading' (MASTER) database. The annotation toolkit is a MATLAB program [56] developed by us to annotate multi-script scene images. A virtual keyboard was designed to input text in Indic scripts. An option exists in the annotation program to add any new script and a suitable virtual keyboard. This annotation program was used to annotate the MASTER database. MASTER database includes multi-script scene images captured by MILE members.

Appendix B describes the details of multi-script robust reading competition organized by MILE laboratory as part of ICDAR 2013 competitions [24]. This competition ran in open mode and had four tasks, namely, script-independent text localization, script-independent text segmentation, English word recognition and Kannada word recognition. Thirty people registered for the competition, but only three participants, from three different countries, actually submitted their results.

1.10 Conclusion and future work

Scene/born-digital images are analyzed for the different tasks, namely, text localization, text segmentation and word recognition. Robust reading can be used to build blind-aids, unmanned navigation systems to tag geographical locations, spam filters and as part of information retrieval systems. QR code and bar code readers are extensively used in ticketing, billing and payments. Video text, which consists of multiple scene text frames, can also be analyzed. Motion in videos may arise due to the movement of a human or a vehicle. Since a frame in the video may or may not have text, the text detection and word recognition problems should be solved simultaneously. ICDAR 2013 has started a new challenge in robust reading competition for text localization in videos [24].

Chapter 2

OTCYMIST: Otsu–Canny minimum spanning tree for text segmentation from born-digital images

Summary

Text segmentation and localization methods are proposed for the born-digital image data set. Image segmentation and edge detection are separately carried out on the three colour planes of the image. Connected components (CC's) obtained from the segmented image are pruned based on their area and aspect ratio thresholds. CC's with sufficient edge pixels are retained. The centroids of the individual CC's are represented as nodes of a graph. A minimum spanning tree is built using these nodes of the graph. Long edges are broken from the minimum spanning tree of the graph. Pairwise height ratio is used to remove likely non-text components. The CC's are grouped based on their horizontal proximity to generate bounding boxes (BB's) of text strings. Overlapping BB's are removed using an overlap area threshold. Non-overlapping and minimally overlapped BB's are used for text segmentation. Each such BB is vertically split, if required, to localize text at the word level. The proposed method is applied on born-digital test data set of ICDAR 2011 and ICDAR 2013 competitions. The values of precision, recall and H-mean are tabulated and compared with the different techniques submitted as entries in the competition.

Figure 2.1: Sample born-digital images from ICDAR 2011 competition data set.

2.1 Introduction

Images with text superimposed by a software are known as born-digital images. They are created using software such as Adobe Photoshop [2], Gimp [17] and Microsoft Paint [57]. Born-digital images are used in web pages and e-mail for names, logos or advertisements. A competition [29] was organized on reading text from born-digital images in ICDAR 2011. Figure 2.1 shows sample images from the data set. The very low resolution of text present in the images and anti-aliasing of the text with the background, form the major differences between born-digital and scene images. Otsu–Canny Minimum spanning tree (OTCYMIST) method [46] is proposed for text segmentation and localization from born-digital images. OTCYMIST is mainly proposed for extracting text from born-digital images and not from scene images.

The proposed method is the winner of text segmentation task in the competition organized during ICDAR 2011 [23]. It consists of binarization, edge detection, minimum spanning tree formation and horizontal text grouping process [46]. Each process is independent and acts as a module in a chain.

2.2 Segmentation

As an initial experiment, k-means clustering was used to form the clusters. The three (Red, Green and Blue) plane pixel values were used as feature vectors in k-means clustering. In the training data set, due to low resolution characters, merging of small width characters into the background cluster was observed. It is undesirable, since missing

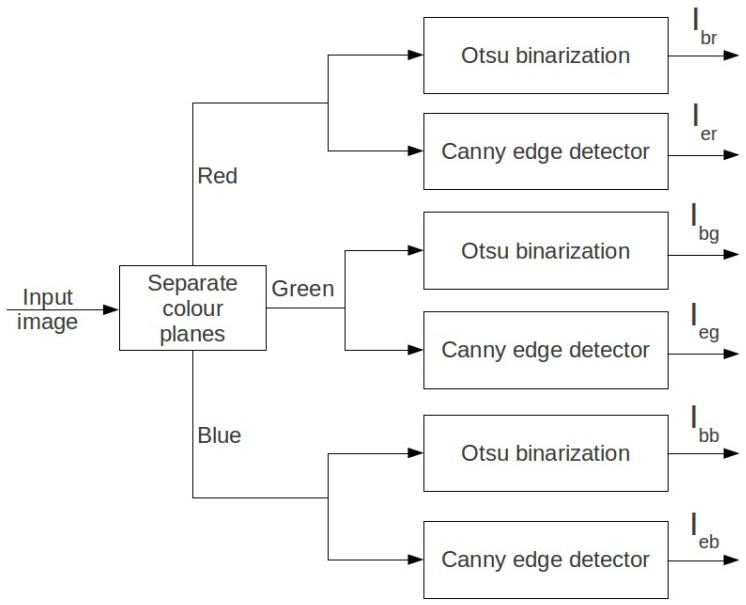


Figure 2.2: The first part of the proposed OTCYMIST method for text localization and segmentation. This comprises the binarization and edge map modules for the individual colour channels.

characters at an early stage impacts the method. Therefore, a thresholding algorithm was chosen instead. Each of the R, G and B colour planes is separately segmented using 'Otsu' global thresholding technique [68]. Figure 2.2 shows the block diagram of the initial part of the proposed method. In Otsu's method, the threshold calculation is posed as an optimization problem. It is used in binarization of document images, and works well on good quality images. Born-digital images, being digitally created rather than captured by a camera, have less variation in lighting or colour both in the foreground and the background of regions. Otsu binarization is an effective method for image segmentation when the variations in lighting and colour are minimal.

2.2.1 Otsu's method

The histogram of an image is used to arrive at the threshold that maximizes the discrimination value. The values in the histogram are normalized before calculating the discrimination value. At each gray value, the histogram is split into two parts. The mean and weight of each histogram part are calculated, and also the discrimination value. This

Figure 2.3: Results of Otsu's binarization of colour channels for the first image in Fig. 2.1. (a) Binarized red plane (I_{br}). (b) Binarized green plane (I_{bg}). (c) Binarized blue plane (I_{bb}).

process is repeated for all gray values. The gray value at which a peak is found for the discrimination value is used as the global threshold. This threshold splits the histogram into two parts. The discrimination value is calculated as follows,

$$\sigma_B^2(k*) = \max_k \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]} \tag{2.1}$$

where,

$$\omega(k) = \sum_{i=1}^{k} p_i \tag{2.2}$$

$$\mu(k) = \sum_{i=1}^{k} i p_i \tag{2.3}$$

$$\mu_T = \sum_{i=1}^{L} i p_i \tag{2.4}$$

Here, L is the total number of gray levels, p_i is the normalized probability distribution obtained from the histogram of the image and $k*$ is the optimal threshold value obtained for that image.

The connected components (CC's) in the binarized image are labeled. The complement version of binarized image is also considered for possible text components, to account for the presence of inverse text.

Edges are detected for individual colour planes using Canny's [10] edge detection algorithm. In this algorithm, a Gaussian filter is applied on an image. Sobel horizontal

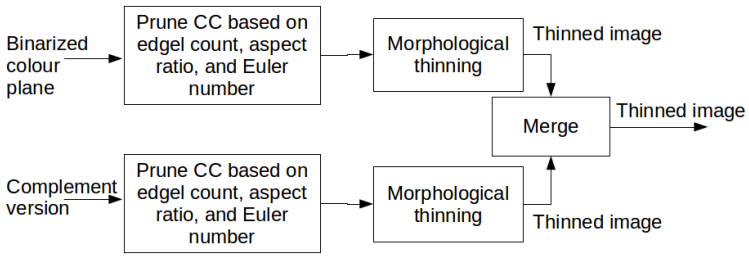


Figure 2.4: Second section of OTCYMIST method separately processes each binarized colour plane (I_{br}, I_{bg} and I_{bb}) and its complement to form a thinned image.

and vertical edge operators are applied on this Gaussian filtered image. The maximum gradient value is obtained from the edge operated image. A threshold is calculated from the gradients, which is specific to that image. This threshold value is applied to the edge operated image to obtain the edge pixels. These edge pixels are termed as edgels and are used to select the CC's from the binarized image.

The proposed method is split into sections for ease of description. Some parts of the method are performed repeatedly on individual colour planes. Figure 2.3 shows the binarized result for each of the colour channels, for one of the sample images shown in Figure 2.1.

2.3 Pruning of connected components

Figure 2.4 shows the second section of the OTCYMIST method. This part of the processing is carried out on each of the binarized planes and their complements. Complement form is used to identify the text of inverse polarity, in case it exists in an image. The area of the CC and the aspect ratio of the bounding box are thresholded to remove possible non-text components. Aspect ratio is defined as the ratio of width to height. Aspect ratio of each CC is calculated. The range of valid aspect ratio is fixed as 0.01 to 20. During validation, several stray pixels were observed in the binarized image due to thresholding. The CC's with more than 5 pixels were only retained, thereby removing the stray pixels. The CC's which have pixels on the image boundary are removed; this ensures that broken characters do not appear in the next step of the method.

Figure 2.5: Pruning and thinning of connected components in each colour plane. First column: Results of Otsu's binarization of colour channels for the last image in Fig. 2.1. It can be seen that the text that is lost in binarization in one colour plane is captured in some other colour plane. Second column: Complement version of the results of the first column. Part of the text that was merged with the background becomes foreground in the complement image and can be successfully captured. Third column: Combination of each binarized plane and its complement version after connected component analysis. Fourth column: Thinned images of the results of the third column.

In the case of low resolution text characters, there will be a displacement of one pixel in the Canny edge operation. Hence, the CC's are morphologically thickened. The thickened CC's are placed on the edge map of the colour plane to count the number of edgels present within the thickened CC. Non-text components may not have a high density of edgels on the boundary. The CCs with edgel count less than 6 are removed. The training data set consists only of English characters. The maximum Euler number for English characters is 2. Hence, the CC's whose Euler number is greater than 2 are also removed. The CC's preserved in the binarized plane and its complemented version are morphologically

Figure 2.6: Thinned planes obtained after the merging operation in second section, where the text and non-text components are easily identified. (a) Thinned red plane (I_{tr}). (b) Thinned green plane (I_{tg}). (c) Thinned blue plane (I_{tb}).

skeletonized and clubbed together to form a single thinned image plane. Figure 2.5 shows the intermediate results of this section. Figure 2.6 shows the resulting individual thinned image planes for the first sample image shown in Figure 2.1. The components in these thinned image planes are referred to as thinned connected components (TCC's).

2.4 Minimum spanning tree

Delaunay triangulation is used for page layout analysis of a document image. It provides good segregation between text and non-text sections in a document image. Nourbakhsh et. al [64] have used them to extract text paragraphs from a document image. Extraction of a text line is more complex from a hand-written document than from a printed document. However, Yin and Liu have showed that minimum spanning tree has the ability to extract arbitrary handwritten lines from Chinese documents [88]. Compared to non-text components, the character components are closer in a text string. This characteristic of text can be better utilized in a minimum spanning tree (MST). A MST is generated for the graph using the algorithm proposed by Prim [70]. Prim's algorithm uses the shortest distance between the nodes and bridges the nodes to generate spanning tree for the graph. The centroid of the TCC's in each of the thinned planes are used as the nodes of a graph. The centroid of the TCC's is calculated by averaging the pixels in TCC's. The length of the edge connecting two nodes is given by the Euclidean distance between the corresponding centroids. Isolated nodes with long edges are broken using a length threshold

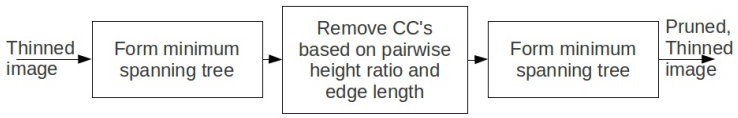


Figure 2.7: Minimum spanning tree and pairwise height ratio check in the proposed OTCYMIST method. This section operates on each of the three thinned planes (I_{tr}, I_{tg} and I_{tb}) obtained from the previous section.

2.5 times the mean value of edges present in the spanning tree. The average length of all the edge in the spanning tree may be approximately equal to the length of adjacent characters. Suppose some middle or adjacent character is missed during segmentation due to degradation. In order to retain the last or first character, we need to allow at least twice the length. Thus, a marginally higher value of 2.5 is specified as threshold for retaining the components. The isolated nodes are presumed to be non-text components appearing in the thinned image. Figure 2.7 shows this section of the method, which is applied separately on each of the thinned images obtained from the three colour channels. Figure 2.8 shows the intermediate results of this section. The removal of non-text components can be observed in Figure 2.8 after each pass of minimum spanning tree.

A text string has characters of comparable heights. The preserved nodes of spanning tree are subjected to mutual height ratio test. The height of each TCC is compared with other TCC's. TCC's with more than two associated height ratios in the range 0.5 to 2, are retained. The retained TCC's are converted into nodes and another pass of minimum spanning tree is carried out. The mean value of the edges is reduced after the first round, if non-text TCC's have been removed. Some of the non-text components with heights similar to that of text components may not be removed by the above check. These non-text components are usually far away from the string of characters. Second pass of minimum spanning tree is to ensure the removal of these non-text components also.

The result of this section is used to retrieve the actual CCs from the binarized image. The CCs from all the three binarized images are grouped into a single plane for text segmentation and localization.

Figure 2.8: Analysis of minimum spanning tree on thinned colour planes. First column: Thinned colour channels for the last image in Fig. 2.1. Second column: Results after the first pass of minimum spanning tree. A few non-text components are removed. Third column: Results after the second pass of minimum spanning tree. Non-text components, which were still present, have now been removed. Fourth column: Connected components retrieved after the second pass of minimum spanning tree in each plane before merging into a single plane.

2.5 Grouping of text components in horizontal direction

Figure 2.9 shows the final section of the method. The bounding box and centroid value of each CC are obtained from the combined plane. Since most of the training images have only horizontally oriented text, CC's are grouped into words based on their proximity in the horizontal direction. These CC's are sorted in the ascending order, based on the earliest occurring top row of the CCs. The CC's with centroids falling within the vertical range of the bounding box of a candidate CC, are grouped together. The grouping

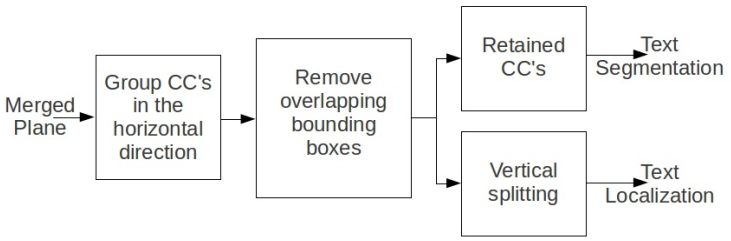


Figure 2.9: The final section of the OTCYMIST method which segments and localizes the text. Vertical splitting block locates the individual words from a group of words.

process is repeated until all CC's have been grouped. Figure 2.10 shows CC's before and after this horizontal grouping process. Some non-text components which pass through the minimum spanning tree module overlap largely between one another. Hence, the bounding box values of grouped CC's are tested for overlap, and those with higher percentage of overlap are removed. We found that a bounding box should be removed if the area of its overlap with another bounding box is more than 20 percent of its own area. Bounding boxes containing a single CC are also removed, since we are interested only in bounding boxes containing words.

The CC's from the selected bounding boxes are preserved as segmented text at the level of pixels. Bounding boxes containing more than 4 CC's are tested for the presence of multiple words. If the maximum value of gap between two neighboring CC's is higher than the mean value of the inter-component gaps by more than 3 pixels, then a threshold of (max value − 2) pixels is used to split the bounding box into two. However, if the height of the bounding box of the horizontally grouped CC's is less than 11 pixels, a fixed

Figure 2.10: CC's before grouping. After grouping, each group is replaced by a white mask; thirteen distinct groups can be seen. Most of the non-text components are removed due to the overlap, which does not happen for the text components.

Figure 2.11: Result of the proposed OTCYMIST method for the first born-digital image in Fig. 2.1. (a) Segmented text. (b) Localized text. (c) Text Localization mask.

threshold of 3 pixels is used to split the bounding box into words. These thresholds have been obtained by analyzing the 420 training images given; an exact relationship between the font size of the characters (determined approximately using the average height of CCs in a text string) and word gap could not be determined due to large variations of the word gap value in the training images. This splitting process to form individual words is termed as vertical splitting. After this splitting, the bounding box values are recalculated and the text localization details are created at the output stage.

This final section of the OTCYMIST method gives two outputs. Selected CC's form the segmented text at the pixel level. Splitting generates the final bounding boxes. The bounding box list provides the text localization information. Figure 2.11 shows the results of segmentation and localization of the text for the first example image shown in Figure 2.1. Figure 2.12 shows the results of text and inverse text obtained using our method for the second example image shown in Figure 2.1 .

Figure 2.12: Segmentation and localization results of OTCYMIST method on the second image in Fig. 2.1, which has text of both polarities.

Table 2.1: Text localization results (%) of ICDAR 2011 Robust Reading Competition: Challenge-1 on BDI [23] evaluated using Wolf and Jolion method.

Method	Recall	Precision	H-mean
TDM IACAS	69.70	85.83	76.93
TH-TextLoc	73.06	80.39	76.55
Textorter	69.08	85.54	76.43
Baseline Method	69.94	83.92	76.30
OTCYMIST	75.65	63.85	69.25
SASA	64.91	67.38	66.12
TextHunter	58.43	75.52	65.88

2.6 Performance evaluation

The training set of the ICDAR 2011 data set has a total of 420 images [29]. The ground-truth of training data set is in pixel format for text segmentation and bounding box format for text localization. Normally, a document image has some definite structure and requires page layout analysis before passing the processed image for OCR. But in these training images, the location and the number of words and characters vary randomly. The ground-truth of training images was used for tuning parameters in the cross validation process. A set of 102 images were provided during the testing phase by the competition organizers.

In ICDAR 2011 competition, text localization performance is evaluated as described by Wolf and Jolion [86]. ABBYY OCR software was used as the baseline method in the performance evaluation [1]. The performance of the baseline method was comparable to the best performing algorithm during the competition period. Table 2.1 gives the competition results for the text localization task [23].

Text segmentation performance of any technique is evaluated using Clavelli's method

Table 2.2: Text segmentation results (%) of ICDAR 2011 Robust Reading Competition: Challenge-1 on BDI [23] evaluated using Clavelli's method.

Method	Well Segmented	Merged	Lost	Recall	Precision	H-mean
OTCYMIST	64.14	15.69	20.15	80.62	72.06	76.10
Textorter	58.12	9.50	32.37	65.23	63.63	64.42
SASA	41.58	10.97	47.43	71.68	55.44	62.52

Table 2.3: Text localization results (%) of ICDAR 2013 Robust Reading Competition: Challenge-1 (on BDI) evaluated using Wolf and Jolion method.

Method	Recall	Precision	Hmean
USTB_TexStar	82.38	93.83	87.74
TH-TextLoc	75.85	86.82	80.96
I2R_NUS_FAR	71.42	84.17	77.27
Baseline	69.21	84.94	76.27
Text Detection	73.18	78.62	75.81
I2R_NUS	67.52	85.19	75.34
BDTD_CASIA	67.05	78.98	72.53
OTCYMIST	74.85	67.69	71.09
Inkam	52.21	58.12	55.00

[12]. This method verifies whether the skeleton of the estimated connected component is fully contained in the ground truth CC. This is used to quantify minimal covering of the output of the segmentation algorithm. Then, using the stroke width of the ground truth component, the skeleton is dilated and then superimposed on the ground truth to quantify maximal covering of the result of the algorithm being tested. Maximal and minimal covering information are used to classify the extracted text components as well-segmented, merged, broken or lost. This is used as the primary evaluation for judging the performance of the algorithms. Clavelli's method is an improvisation of the technique proposed by Ntirogiannis et. al [66] to rank document image binarization algorithms. The precision and recall measures at the pixel level are used as the secondary indices of segmentation performance.

Table 2.4: Text segmentation results (%) of ICDAR 2013 Robust Reading Competition: Challenge-1 (on BDI) evaluated using pixels and atoms.

Method	Pixel Level			Atom Level		
	Recall	Precision	F-score	Recall	Precision	F-score
USTB_FuStar	87.21	79.98	83.44	80.01	86.20	82.99
I2R_NUS	87.95	74.40	80.61	64.57	73.44	68.72
OTCYMIST	81.82	71.00	76.03	65.75	71.65	68.57
I2R_NUS_FAR	82.56	74.31	78.22	59.05	80.04	67.96
Text Detection	78.68	68.63	73.32	49.64	69.46	57.90

Figure 2.13: Segmented text, localized text and localized bounding box mask obtained by OTCYMIST method for test images from the born-digital data set.

2.7 Results and Discussion

Table 2.2 lists the results of the OTCYMIST method for the text segmentation task and compares it with those of the other entries in ICDAR 2011 competition [23]. OTCYMIST method is the winner of text segmentation task in the competition and is ranked fourth in text localization task. The effectiveness of Otsu binarization resulted in the highest number of text components being segmented properly in ICDAR 2011 born-digital image dataset and eventually winning the competition for the text segmentation task. Columns titled 'well segmented', 'merged' and 'lost' in Table 2.2 indicate recognition performance on the segmented image. 'Background' components are not considered in the evaluation process. However, including a penalty for the background components detected as text components is preferable to improve the assessment of the quality of a method.

As a continuation of Robust Reading Competition in ICDAR 2013 [24], same training set of the ICDAR 2011 data set was provided in ICDAR 2013 [30]. A set of 141 images were provided during the testing phase by the competition organizers, which is an increase

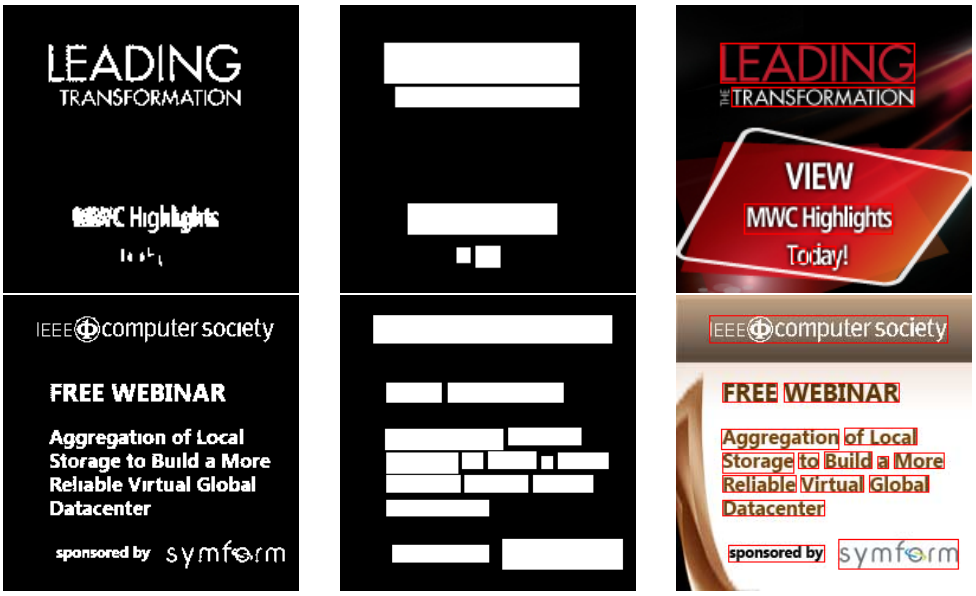


Figure 2.14: Segmented text, localized text and localized bounding box mask obtained by OTCYMIST method for test images from the born-digital data set.

of 40%. The competition organizers used the same evaluation procedure, which was used in ICDAR 2011 Robust Reading Competition, for evaluting the submitted algorithms. Table 2.3 gives the competition results for the text localization task [30]. Table 2.4 lists the competition results for the text segmentation task [30]. OTCYMIST method has consistent performance even with additional 40% images in the test set. OTCYMIST method is placed in third position for text segmentation task.

OTCYMIST approaches text localization by segmenting the image, which is totally different from the known algorithms such as [11, 15] used for text localization on ICDAR 2005 dataset. Epshtein et. al [15] use Canny edges obtained from a scene image to construct stroke width transform. Stroke width of each pixel is used to group adjacent pixels and find letter candidates, which resemble connected components. These letters are filtered to obtain text lines and words. Chen et. al [11] use region growing approach to form maximally stable extremal regions. The stable regions are considered as character candidates or connected components and further geometric filtering is performed to group character candidates as text lines and words. Chen et. al use edge information which is similar to stroke width transformation during the filtering stage. An image can be

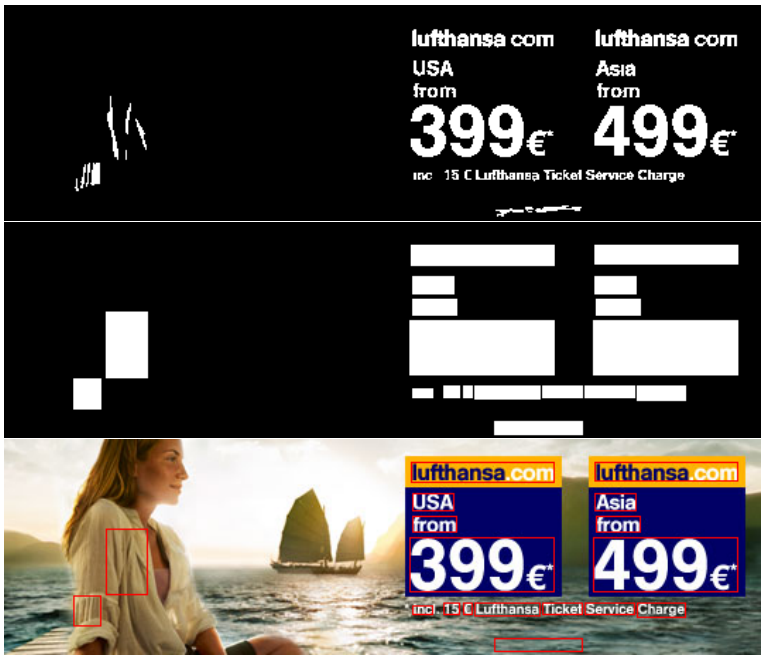


Figure 2.15: Segmented text, localized text and localized bounding box mask obtained by OTCYMIST method for a test image from the born-digital data set.

analyzed using edges, regions, or segments. The algorithms popular for scene images use edges and regions for image analysis, but OTCYMIST proposed mainly for born-digital images uses global threshold. OTCYMIST performance for text localization on ICDAR 2011 and 2013 scene image datasets is poorer, since it is optimized for text segmentation on born-digital images.

A few of the born-digital images have slowly varying illumination in the background. Otsu's method segments the slowly varying background into two regions and merges the actual text part with anyone of them. Thus, the slowly varying illumination part in the image, if it exists, needs to be identified. Dealing with the varying illumination in an image is our future work for improving the method. Figures 2.13, 2.14, 2.15 and 2.16 give a number of born-digital images and the results of segmentation and localization by OTCYMIST technique. Figure 2.17 shows some sample test images, on which our method fails at the level of binarization or thresholding the CC's.

Figure 2.16: Segmented text, localized text and localized bounding box mask obtained by OTCYMIST method for a test image from the born-digital data set.

2.8 Conclusion

A novel method has been proposed for the segmentation and localization of text in born-digital images. The performance of OTCYMIST method on bipolar text is good. Minimum spanning tree is used in our method. Earlier, minimum spanning tree has been used for line extraction in handwritten documents. Image segmentation is approached by splitting the different colour channels of an image.

Only segmenting or locating text in an image is not sufficient. We need to recognize the actual text from either segmented text components or located bounding box. In the next chapter, we propose different approaches for word recognition from the cropped word images.

Figure 2.17: Images from the born-digtial competition data set, where OTCYMIST method fails. The first image has low contrast. The second image has a single character, which gets eliminated during the grouping process, in the proposed method.

Chapter 3

Recognition of scene word images through segmentation

Summary

The task of word recognition is considered as a part of object recognition in the field of computer vision. A word cropped from a document image can be recognized using a traditional optical character recognition (OCR) engine. However, recognition of a word image, even when it is manually cropped from a scene or born-digital image, is not trivial. Existing OCR engines do not effectively handle such images. In the literature, the word recognition task is addressed in a top-down approach using descriptors such as histogram of oriented gradients or directional features. A custom lexicon is used in this top-down approach. The word recognition rate for few data sets is not good even with such lexicons. Our intention is to use the existing OCR engines for recognition. In two aspects, it is advantageous: (i) it avoids building a character classifier from scratch and (ii) it reduces the word recognition task to the word segmentation task. Here, we propose two elegant bottom-up approaches for the task of word segmentation. These approaches choose different features at the initial stage of segmentation. Experiments are conducted on robust reading, sign evaluation and street view text data sets. In place of passing an image segmented by a method, manually segmented word image is submitted to an OCR engine for benchmarking maximum possible recognition rate for each data set. Our results are reported along with

the benchmark results. The recognition rate of proposed methods without the use of any lexicon are comparable or better than all the methods in the literature for four of the standard data sets.

3.1 Introduction

Image segmentation is an active research area for object detection and recognition. In document analysis systems, early research was focused on segmentation and recognition of text from scanned documents. Often, the recognition of characters (text) is performed using OCR engines. Segmentation of a document image into its constituent parts such as lines or words and recognition of words are necessary steps in the process of document digitization. Several good character recognition (OCR) engines are available [1, 2, 67, 76] for the Roman script.

The major problem in trying to directly recognize scene word images without going through the complex step of text localization is the degradation present in it. It is difficult to segment a degraded word image. During the segmentation process, several characters get merged due to minimal character gaps, thus reducing the recognition rate on word image data sets. The minimum character gap in the born-digital word data set is 1 pixel width, which was found during the cross validation process. Power-law transform, which is used to correct gamma values in display systems, is applied to prevent the merging of characters, thus increasing the possible word recognition rate for each word image data set. Here, methods are developed to segment word images and the trial version of Omnipage OCR [67] is used to recognize the characters in the segmented image.

3.2 Competition on word recognition

In ICDAR 2003, a competition was organized [52] for text localization on camera-captured images and recognition from the word images extracted by placing bounding boxes on the images. Some camera-captured word image samples extracted from scene images are shown in Figure 3.1. Only five entries were received for text localization and none for word

Figure 3.1: Sample cropped word images from ICDAR 2011 data set.

recognition. In continuation, Lucas et. al conducted only text localization competition in ICDAR 2005 [53]. One may assume that the bounding box information of a word is sufficient for any OCR to recognize the word. However, the best performing method only has a word recognition rate of 61.1% on ICDAR 2003 test set [60], even though it uses a customized lexicon derived from the set of test images themselves. In a real scenario, since the text that may appear in a scene cannot be predicted, depending on such custom lexicons might limit the scope of word recognition. ICDAR 2011 Robust Reading challenge 2 reports that the best word recognition rate is only 41.2% [75]. This recognition rate is improved to 56.4% using a customized lexicon [65].

Another robust reading challenge was included in ICDAR 2011 to recognize born-digital word images [29]. For the competition, these images were collected from web pages and email. Most of the words present in the data set are horizontally oriented. The reason behind horizontal placement of text may be the simplicity involved in creating the born-digital image, using standard softwares. Sample born-digital word images cropped from born-digital images are shown in Figure 3.2. Low resolution of text and anti-aliasing are the main issues to be tackled in born-digital images, whereas illumination changes is a difficult problem in the case of camera-captured images. Figures 3.1 and 3.2 show degradation due to illumination changes in camera-captured images and low resolution in born-digital images, respectively. This competition received only one entry for word image recognition, whose performance was low and did not match even up to that of ABBYY Fine Reader. This inspired us to explore as to why the word recognition rate was low. Since the resolution of the given images themselves is very low, the words extracted from

Figure 3.2: Sample cropped word images from born-digital data set.

born-digital images are still lower in resolution.

In ICDAR 2013 [24, 30], the task of recognizing born-digital and scene word images were separately organized. The data set used in scene word images is combination of the test and the training word images that were used in ICDAR 2003 competition. The number of scene word images used in this competition is approximately close to the number of word images in the first competition. However, the number of born-digital word images has been increased by fifty percent compared to its earlier competition, which was held in ICDAR 2011.

3.3 Survey of related literature

Methods have been proposed in the literature for segmenting word images, such as conditional random fields (CRFs) to form super pixels by KAIST AIPR [28], maximally stable extremal regions (MSER) by Neumann [55, 62] and Markov random fields (MRFs) by Mishra et. al [58]. Other methods explored are clustering and combining different segmentation techniques [32, 33, 79, 80, 89].

After segmentation of word images, recognition is performed using either a standard OCR engine or a classifier built for the purpose. KAIST AIPR system classifies super pixels and passes them to INZI soft OCR engine [25]. Similarly, TH-OCR system uses its own OCR engine [50]; Mishra et. al use ABBYY OCR reader [1]; Zeng et. al use Omnipage

OCR reader [67] and Neumann and Matas [62] classify detected characters in the image using multi-class support vector machines (SVM) based on the character contour features. Due to variations in the contour feature caused by noise or scaling, Neumann's method was ranked low in ICDAR 2011: Robust Reading Competition Challenge 2. This indicates that better features are required in the classification stage.

Words can be recognized without using any segmentation procedure, but this procedure requires a training data set. Wang et. al, Mishra et. al and Novikova et. al [60, 82, 65, 84] use this approach for word recognition. A classifier is trained on selected features using the training data, which is available with the data set. Apart from the training, they provide lexicon as a support to their methods (available as a part of the Street View Text (SVT) data set) [82], in a top-down approach.

3.4 Methods proposed for text segmentation

The proposed methods use bottom-up approach for the segmentation of word images. Table 3.1 lists the three proposed methods with their distinctions.

Power-law transform (PLT) and Non-linear enhancement and selection of plane for optimal segmentation (NESP) are proposed to counter degradations by operating on the pixel values. Gray values are obtained from the colour word image to decrease the amount of computation involved in processing the pixels. PLT method applies a non-linear transformation to the gray values of an image before segmentation. NESP method is an enhanced version of PLT method, which analyzes all the colour pixel values for optimal segmentation.

Middle line analysis and propagation of segmentation (MAPS) method was proposed to counter degradations by first operating on the middle row pixels of the image. A concept which is not violated in the word images is a regular interval between text and non-text pixels in the middle row of a word image. The set of middle row pixels in an image is considered as a sub-image. MAPS method starts its initial segmentation procedure on the sub-image with an assumption that middle row pixels are least affected by any kind of degradation.

Table 3.1: Distinction between the three approaches proposed for segmentation of word images. MAPS method begins by binarizing the middle line pixels, whereas PLT and NESP methods operate on the gray and colour values of the pixels, respectively.

Method	Initial input to the method	Segmentation approach
PLT	Gray values of an image	Non-linear enhancement before applying Otsu's threshold
NESP	Colour values of an image	Independent non-linear enhancement of each plane before applying Fisher discriminant and Otsu's threshold
MAPS	Gray values of only the middle row of an image	Foreground and background segmentation using Min-cut/Max-flow algorithm

3.4.1 PLT: Power-law transform

During segmentation, the adjacent characters merge in images with poor contrast, which is a kind of degradation. The pixel values are modified to avoid the merge between adjacent characters. The concept of power-law transformation is introduced to non-linearly modify the pixel values, which increases the contrast of an image and helps in better segmentation. The basic form of power-law transformation is [18],

$$f_{out}(x,y) \;=\; C f_{in}^{\gamma}(x,y) \tag{3.1}$$

where $f_{in}(x,y)$ and $f_{out}(x,y)$ are the pixel intensities of the input and output images, respectively; C and γ are positive constants.

A variety of devices used for image capture, printing, and display respond according to a power-law. By convention, the exponent in the power-law equation is referred to as gamma. Hence, the process used to correct the response of these devices is called gamma correction. Gamma correction is used in displaying an image accurately on a computer screen. In our experimentation, γ is varied in the range of 1 to 5. Figure 3.3 shows the change in the appearance of the histogram of a sample word image for different γ values.

Figure 3.4 shows a gray scale image from the born-digital word image data set, after power-law transformation with different gamma values. The word 'GARNIER', which

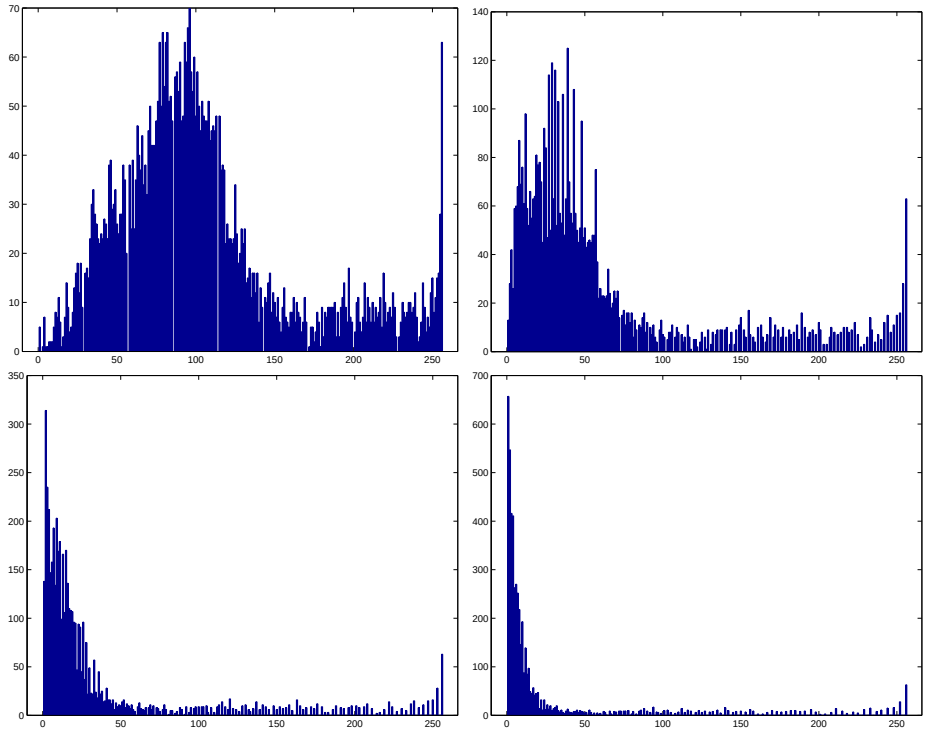


Figure 3.3: Histogram plots obtained for the sample word image 'Ticket,' shown in Fig. 3.2 after power-law transformation with $\gamma = 1, 2, 3$ and 4, respectively. The modified pixel values change the histogram significantly.

exists in the image has yellow background for letters 'G' and 'A'. The gradual improvement in contrast can be clearly identified when the gamma value is increased. Thus, the application of PLT increases the contrast in the image resulting in improved image segmentation. By applying a PLT, the image is enhanced non-linearly when compared to linear operations like contrast stretching. PLT improves the quality of an edge by increasing the edge strength. So, any anti-aliased image with lower edge strength can be passed through PLT to increase the edge strength of the image. Otsu's method [68] is used for segmenting an image, after it is enhanced by PLT.

Figure 3.5 shows the segmentation result and the OCR output for a sample input image, for different values of γ. The recognized result is erroneous when the characters are merged. The merge between adjacent characters can be observed clearly in the left top image in Figure 3.5. With increase in γ value, the merged characters split, resulting in better segmentation and OCR output. As discussed, due to anti-aliasing, characters

Figure 3.4: Contrast enhancement after power-law transformation with $\gamma = 1, 2, 3$ and 4, respectively. The gray patch behind the letters 'G' and 'A' gradually decreases and becomes invisible for $\gamma = 4$. Bottom row: Original image is shown for comparison.

get merged into a single connected component. CC's are eroded non-linearly by applying power-law transform. Pixels at the boundary of the CC get eroded first. The cause for merging is anti-aliasing, and this effect on the text components in the image is removed.

The exact value of γ is difficult to determine for a word image. Each word image has a distinctly different histogram to start with, and hence responds slightly differently for a particular value of gamma. As γ is increased to a large value, individual text components may split further into multiple components. This leads to a poor performance in the next stage, namely OCR in our experiment and reduces the recognition rate for the word image data set. Figure 3.6 shows the plot of word recognition rate as a function of γ, for the entire born-digital image data set of 918 images, with γ shown in log scale. As γ value is increased, there is an improvement in word recognition rate. The word recognition rate then decreases for values of γ higher than 1.6.

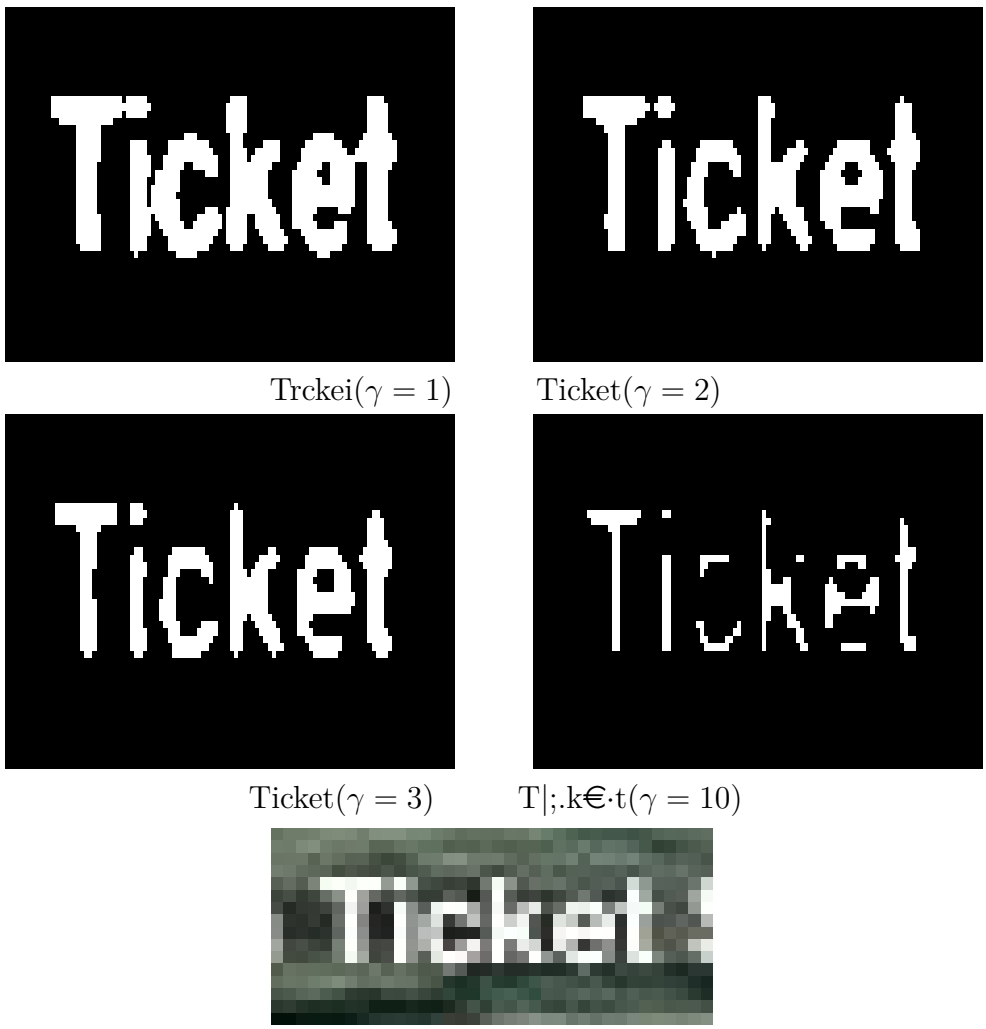


Figure 3.5: Segmented outputs and their respective OCR results for different values of γ. To begin with, increasing the value of γ yields proper output and further increase in the γ value deteriorates the character stroke width. Bottom row: Original image for reference.

An adaptive γ value is estimated to use PLT method for individual images in real applications [44]. Initially γ value is set to '1' and an image is segmented. The average of the horizontal run lengths of text pixels in the segmented image is calculated as a very approximate estimate of the stroke width. If this value is less than 8 pixels, then the segmented image is directly passed to OCR engine for recognition. If the average run length is more than 8 pixels, then gamma value is increased in steps of 0.2. This procedure of power-law transformation, segmentation and run length calculation is repeated until the average run length of segmented image is less than 8 pixels. However, for the competitions,

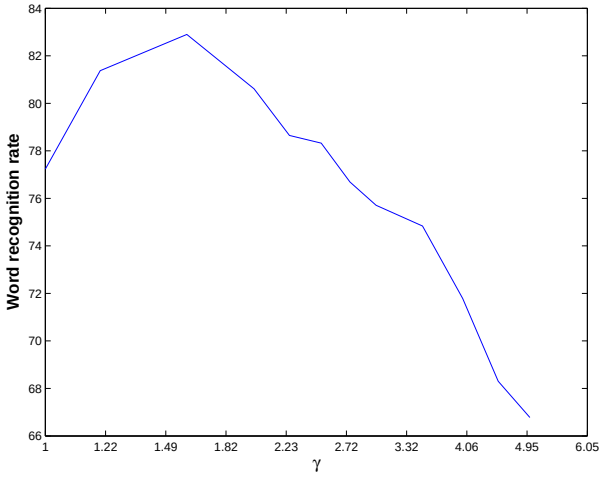


Figure 3.6: Plot of word recognition rate against $\log \gamma$ on the complete ICDAR 2011 born-digital word image data set (918 images in total). Gamma value is fixed in each run to estimate the recognition rate on the entire data set.

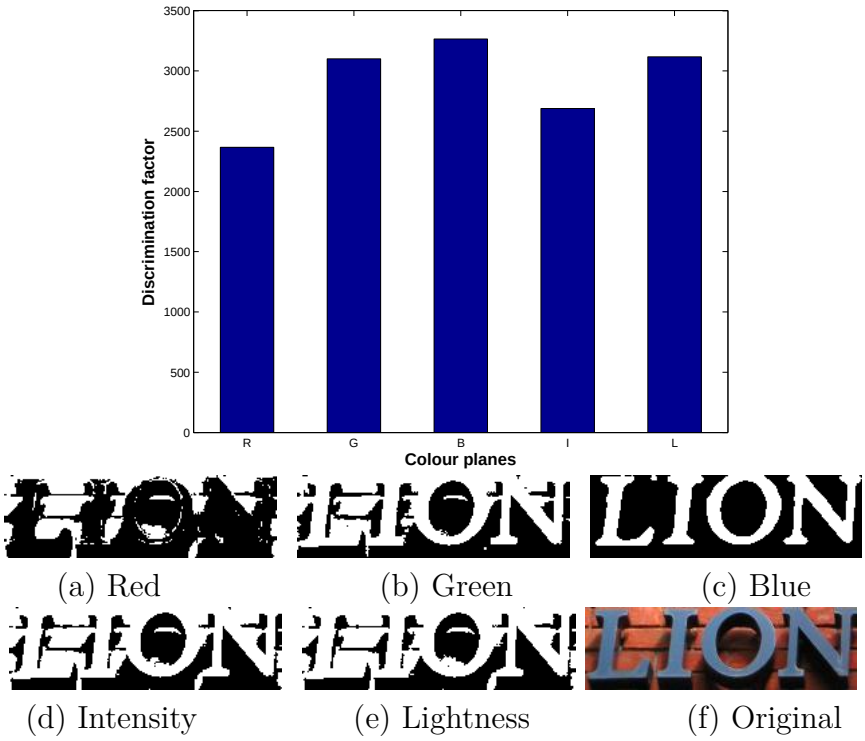
since there is a separate training set, a common value for gamma is chosen based on its best performance on the training set and this value is simply applied on the test dataset also.

3.4.2 NESP: Non-linear enhancement and selection of plane

A word image is split into Red, Green, Blue, Gray and Lightness (CIE Lab) components [18], which we refer to as planes. In PLT method, only the gray values of an image are considered. Usually, gray values of an image have reduced information compared to the colour image. When a colour image is converted to a gray image, the foreground and background gray levels may not be distinct due to the effect of illumination, a kind of degradation. However, they may be separable in one of the other colour planes.

Selection of plane for segmentation

When multiple images are used as input during experimentation, an unique objective metric is required to select the best image [43]. The metric should be in such a way that it can be applied across all the planes derived from an image. When Niblack's method [63] was tried for segmentation, there is no unique metric for each plane due to the local



(a) Red (b) Green (c) Blue

(d) Intensity (e) Lightness (f) Original

Figure 3.7: Illustration of the effectiveness of plane selection in the NESP approach. The top panel shows the discrimination factors computed for different colour planes of the 'LION' word image shown in Fig 3.1. The second and third rows show the images segmented using different planes, which illustrates the effect of selection of the plane using the discrimination factor as an objective measure. In the case of this image, no non-linear enhancement has been applied.

window approach for segmentation. Canny edges [10] can also be used to obtain a measure. However, the degradations in the image provide undue weighting towards degraded edge pixels. This results in the selection of the degraded plane as the "best" plane. To have generality and uniqueness, Fisher discrimination function is applied as proposed by Otsu [14, 68]. Hence, the discrimination factor is chosen as the fitness measure for a plane.

Further, it is not possible to infer a priori the plane in which the text is best separable from the background. Each plane is considered to have both the classes of interest and the problem of best plane selection is posed as the maximization of two-class Fisher discriminant function across all the planes. This measure is calculated for each plane, as:

$$d_c^2(k*) = \max_k \frac{[\mu_{ct}\omega_c(k) - \mu_c(k)]^2}{\omega_c(k)[1-\omega_c(k)]}, \quad c \in [R,\ G,\ B,\ I,\ L] \tag{3.2}$$

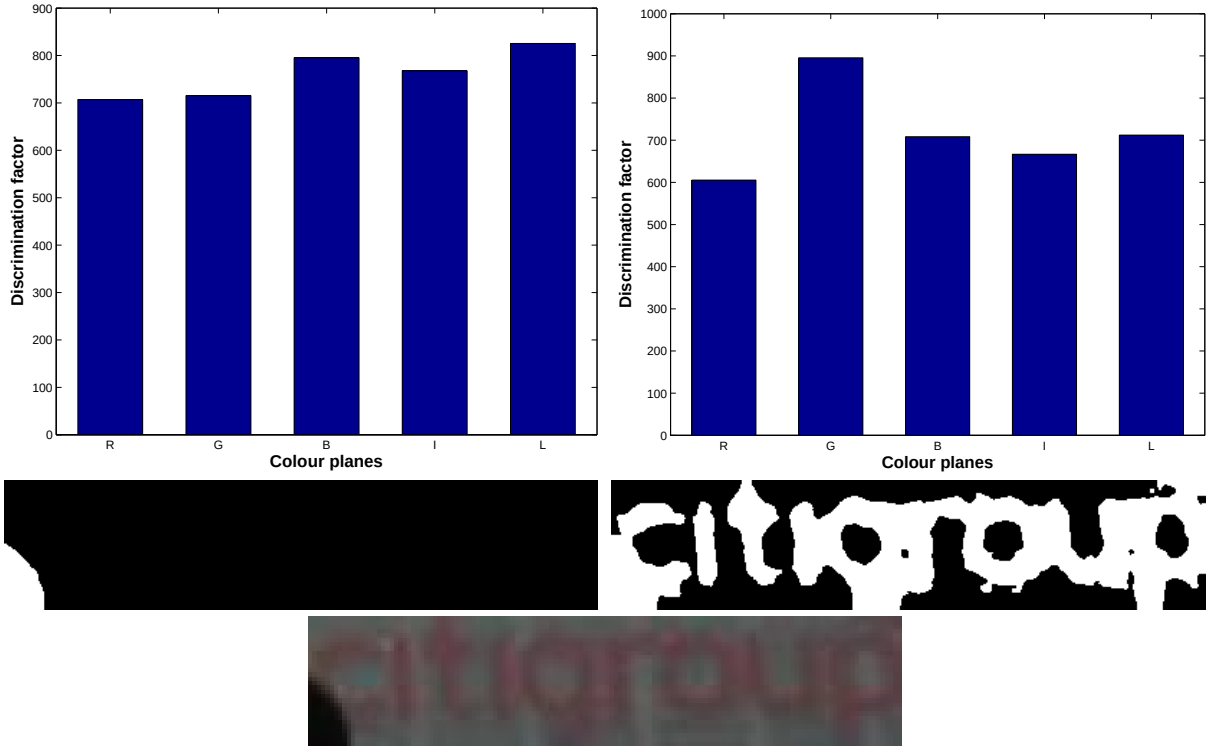


Figure 3.8: Illustration of the effect of non-linear enhancement. Top row: Discrimination factors computed by our method for different colour planes before and after power-law transformation ($\gamma = 1.6$) for 'citigroup' word image shown in Fig 3.1. Second row: Results of segmenting the selected plane. Bottom row: Original image for reference.

where,

$$\omega_c(k) = \sum_{i=1}^{k} p_{ci} \tag{3.3}$$

$$\mu_c(k) = \sum_{i=1}^{k} ip_{ci} \tag{3.4}$$

$$\mu_{ct} = \sum_{i=1}^{N} ip_{ci} \tag{3.5}$$

Here, N is the total number of gray levels and p_{ci} is the normalized probability distribution from the histogram of the c^{th} colour plane. $\omega_c(k)$ and $\mu_c(k)$ are the zero-th and the first order cumulative moments of the histogram up to k^{th} level, respectively and μ_{ct} is the mean of the complete histogram. The plane with maximum discrimination factor is segmented using the corresponding threshold.

NESP method has to decide a plane from the range of red, green, blue, intensity, and luminous plane and also choose a particular gamma value from the range of 1, 1.2, 1.4, 1.6, 1.8, and 2. NESP method has to be illustrated using both plane wise and gamma wise. Figure 3.7 shows the individual effectiveness of the selection of plane alone from the list of planes and Figure 3.8 shows the effectiveness of non-linear enhancement in achieving better text segmentation from scene word images.

Figure 3.7 shows the discrimination factor for each plane without PLT and segmentation of the word image for the selected plane, respectively, for two sample images from ICDAR 2011 data set. This illustrates the advantage of plane selection with an objective criterion in improving binarization. In the case of heavily degraded images, plane selection alone may not achieve the desired improvement, and PLT performs the additional job. The top left panel in Figure 3.8 shows the values of the Fisher discrimination factor obtained for each plane for a very low contrast image. The left image in the middle row is the binarized image obtained by segmenting the plane (Lightness plane) with the highest discrimination factor and it is clear that the binarization is not effective in segmenting the text. The top right panel shows the values of the discriminant factor after the proposed non-linear transformation with a gamma value of 1.6. The enhanced segmentation achieved by the power-law transformation is obvious.

3.4.3 MAPS: Midline analysis and propagation of segmentation

This method [41] starts by segmenting the middle row pixels selected as a sub-image. The gray values of an image are processed in this method. The middle row is segmented independently using Niblack and Min-Max methods. Then, the statistics derived from the midline pixels belonging to the two classes are utilized to segment the rest of the pixels in the image.

Niblack Method

Niblack proposed a method [63] to calculate a local threshold for each pixel by moving a rectangular window over the whole image. Here, Niblack's method is applied to the one

dimensional sub-image that is the middle row of an image. The mean and the standard deviation values of all the pixels within the window are used to calculate the threshold. Thus, the threshold is computed as:

$$T_i = \mu_i + \ k_n \ * \ \sigma_i \tag{3.6}$$

$$\mu_i = \frac{1}{N_w} \sum_{j \in N_w} f(x_{i+j}, y) \tag{3.7}$$

$$\sigma_i^2 = \frac{1}{N_w} \sum_{j \in N_w} (f(x_{i+j}, y) \ - \ \mu_i)^2 \tag{3.8}$$

Here, $f(x_{i+j}, y)$ is the gray value at pixel position x_{i+j} in the middle row y. The user has to fix the values for k_n and window size. We have used $k_n = 0.1$ and $N_w = min(height, width)/2$. In our experiments, these values are fixed for all the tested word images.

Min-Max Method

Locally adaptive methods make use of local statistics to infer the presence or absence of the two classes within the window. In the case of a bimodal distribution, the minimum and maximum values belong to different distributions. These minimum and maximum values are not dependent on the representation of each distribution within the window. But in the presence of noise, these minimum and maximum values get heavily biased. By designing a Min-Max filter with a carefully chosen size of window, it is possible to mitigate the effect of noise. The maximum values are obtained for two windows placed to the left ($f_L(x_i, y)$) and to the right-side ($f_R(x_i, y)$) of position i on the middle row pixels. T_{max} is estimated as the minimum of these maximum values. Similarly, maximum operation is performed on the minimum values in the left and right-side windows to obtain T_{min} .

$$f_L(x_i, y) = [f(x_{i-N_m+1}, y), ..., f(x_i, y)] \tag{3.9}$$

$$f_R(x_i, y) = [f(x_i, y), ..., f(x_{i+N_m-1}, y)] \tag{3.10}$$

$$T_{max} = \min(\max_{N_w} f_L(x_i, y), \max_{N_w} f_R(x_i, y)) \tag{3.11}$$

$$T_{min} = \max(\min_{N_w} f_L(x_i, y), \min_{N_w} f_R(x_i, y)) \tag{3.12}$$

$$T = (T_{min} + T_{max})/2 \tag{3.13}$$

where, N_w is the size of the moving window obtained as $N_w = min(height, width)/2$. The window size was fixed in our experiment, since varying the size did not improve the segmentation.

The plot of gray values of the middle row from a degraded image is shown in Figure 3.9. The local thresholds obtained by Niblack and Min-Max methods are also plotted. The adaptation of both the thresholds to the gradual change in the gray values of the image is seen. However, the variation of threshold in Niblack method is more than that of Min-Max method. Niblack method calculates the threshold by averaging pixel values in a window. When the window slides, this value varies. But in Min-Max method, when we slide a window, the variation of maximum and minimum values is less. Hence, the threshold of Min-Max method is more stable than that of Niblack method.

Classification of other pixels

The threshold obtained using either Niblack's or Min-Max method segments the middle row pixels into two classes. The labels obtained from middle row segmentation are used to estimate the means and variances of the two classes.

$$\mu_0 = \frac{1}{N_0} \sum_{i \in C_0} f(x_i, y) \tag{3.14}$$

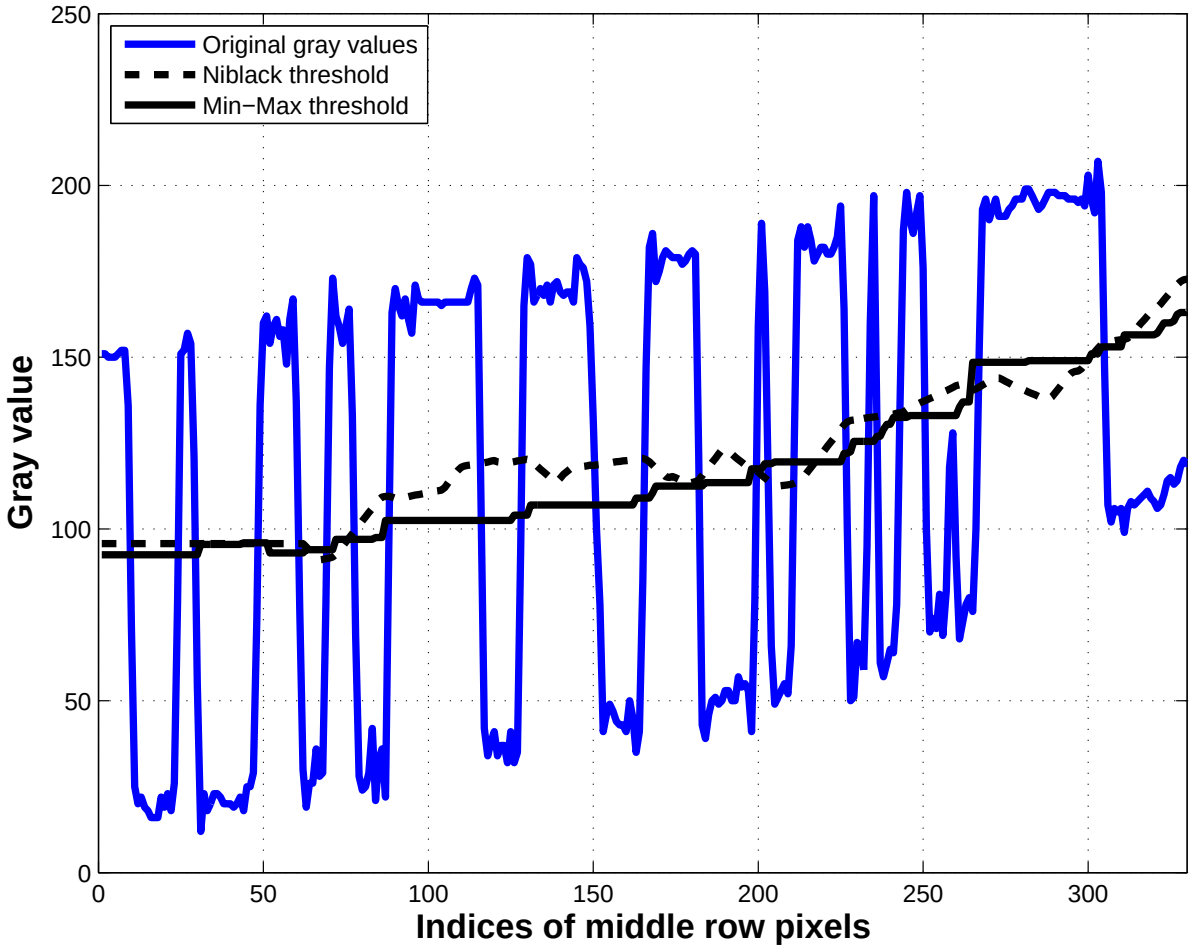


Figure 3.9: A plot of gray values of the middle row of an image is shown, together with the binarization thresholds given by Niblack and Min-Max methods. Both methods slowly adapt to the gradual change in gray levels, but the fluctuations are less in Min-Max method.

$$\sigma_0^2 = \frac{1}{N_0} \sum_{i \in C_0} (f(x_i, y) - \mu_0)^2 \tag{3.15}$$

where μ_0 is the mean of class C_0, which has N_0 labels. σ_0^2 is the variance of class C_0. Similarly, the mean μ_1 and the variance σ_1^2 for class C_1 are calculated.

Only the middle row is considered for initial segmentation and parameter estimation. Other pixels need to be labeled through classification. Again, two different approaches are followed for classification, namely, Bayesian classification and Min-Cut/Max-Flow algorithm. The purpose behind two separate segmentations is to examine any variation in

segmentation, based on computational aspect involved. The worst case running time complexity for Bayesian classification is O(XY), whereas for Min-Cut/Max-Flow algorithm, it is O($(XY)^3$). Here, X and Y are the number of rows and columns in the image.

Bayesian classification

In Bayes binary classification [14], the posterior probability of sample $f(x, y)$ belonging to class C_0 is given as:

$$p(C_0|f(x,y)) \;=\; \frac{p(f(x,y)|C_0)p(C_0)}{p(f(x,y)|C_0)p(C_0) + p(f(x,y)|C_1)p(C_1)} \tag{3.16}$$

where $f(x, y)$ is the gray value at pixel position (x, y). The prior probability is

$$p(C_0) = N_0/N \tag{3.17}$$

For classification,

$$h(f(x,y)) : \begin{cases} p(C_0|f(x,y)) \;\geq\; p(C_1|f(x,y)), & f(x,y) \;\in\; C_0 \\ p(C_0|f(x,y)) \;<\; p(C_1|f(x,y)), & f(x,y) \;\in\; C_1 \end{cases} \tag{3.18}$$

The class-conditional density is assumed as Gaussian for binary classification [14], given by

$$p(f(x,y)|C_0) \;=\; \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\{-\frac{(f(x,y)-\mu_0)^2}{2\sigma_0^2}\} \tag{3.19}$$

Other non-middle row pixels in the image are classified using Equation 3.18. Then, the classified pixels represent the segmented word image.

Min-cut/Max-flow algorithm

In Bayesian method, individual pixels are considered at the time of classification and neighborhood pixel gray values are completely ignored. In the min-cut/max-flow algorithm, graph cut originally proposed for image segmentation is used to add the neighborhood gray values as smoothness term at the classification stage. Pixels are represented

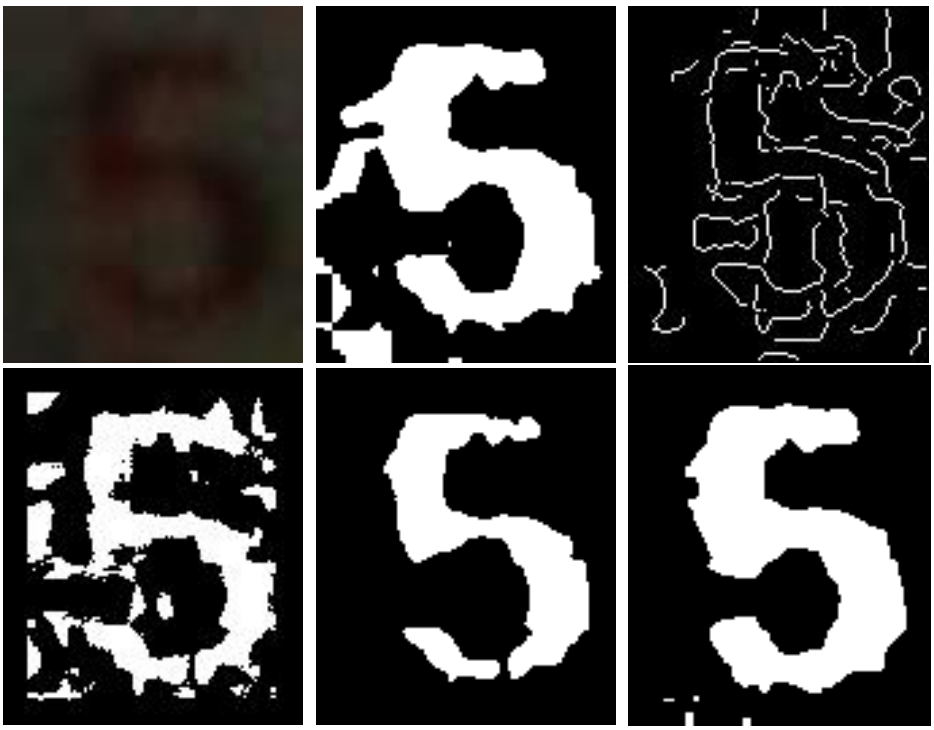


Figure 3.10: Segmentation of a sample word image from ICDAR 2003 data set. (a) Original image. (b) Output of Otsu's method [68]. (c) Edges by Canny's method [10]. (d) Segmented output of Niblack's method [63]. (e) Output of MAPS technique using Min-Max method for segmentation and Max-flow algorithm for classification. (f) Output of NESP technique is better than Otsu's and MAPS results.

as graph nodes with edges connected to other pixels. The energy function of the Potts model [9], which has to be minimized, is given as

$$E(L) = \sum_{i \in \mathcal{I}} D(f_i, L_i) + \sum_{i \in \mathcal{I},\ j \in \mathcal{N}(i)} V(f_i, f_{(i+j)}, L_i, L_{(i+j)}) \tag{3.20}$$

where $L = L_i | i \in \mathcal{I}$ is a labeling of the image $\mathcal{I}$, $D(.)$ is a data penalty function, $V(.)$ is an interaction potential, and $\mathcal{N}(.)$ denotes the neighborhood of i.

The energy minimization is performed by incorporating the means and variances as parameters into data penalty and interaction potential functions by Boykov et. al [9]. The minimum energy corresponds to the segmented image.

The result for a sample image drawn from ICDAR 2003 data set is shown in Figure 3.10, along with segmentation results by Otsu's [68], Canny's [10], Niblack's [63],

and NESP methods. From the results, it is evident that even degradations like uneven illumination and low contrast are effectively handled by our methods.

3.5 Recognition and evaluation of segmented images

The segmented word images are fed to a recognition engine. There are several OCR engines available such as Abbyy Fine Reader [1], Adobe Reader [2], Omnipage [67] and Tesseract [76]. These engines are used to digitize machine printed documents. The quality of OCR performance is good for all the above engines and hence any one of them can be used to recognize the word from a segmented word image. In our work, the trial version of Omnipage OCR engine is used. The OCR result is used for evaluating the performance of our techniques. The performance of a method is quantified by counting the number of properly recognized words. A standard edit distance metric [49] is also used for the evaluation of the recognized word. Equal weights are given for additions, substitutions and deletions. Normalized edit distance is calculated between the transcriptions of the ground-truth and the method.

3.6 Data sets used for the experiments

We have considered seven publicly available word image data sets, namely ICDAR 2003, PAMI 2009, SVT 2010, Born-digital 2011, ICDAR 2011, Born-digital 2013, and ICDAR 2013 data sets for experimentation. These data sets differ in the creation of the word images and the definition of the boundary. Further, these data sets cover all types of degradations.

3.6.1 ICDAR 2003 data set

Robust reading competition [52] was first conducted in ICDAR 2003. Mishra et. al [58] showed 52% word recognition on the sample ICDAR 2003 data set, but not on the test set. This result indicates that more attention must be paid to solve the task of word

recognition. ICDAR 2003 test set consists of 1110 word images.

3.6.2 PAMI 2009 data set

This sign evaluation data set was prepared by Weinmann et. al [85]. This and the SVT data sets were originally used to demonstrate the ability of top-down approach for character and/or word recognition. A lexicon provides information from the top layer to the middle layer during the recognition stage. This limited lexicon, formed from the words of test set, is used to derive N-gram statistics for recognition.

The test set of 215 word images consists only of horizontally aligned characters, except for one or two. The degradation in the images is also minimal. Hence, the recognition rates reported by different methods are all close.

3.6.3 SVT 2010 data set

Wang and Belongie introduced street view text (SVT) data set [83] obtained as a part of Google Street View project. The SVT test set consists of 647 labeled word images obtained from businesses around a location. It also provides a synthetic lexicon created out of the ground-truth, in which each word image has an associated custom dictionary consisting of 49 distracting words and the actual word. Apart from the other degradations, these word images have also undergone motion blur.

This data set consists only of name and location information of businesses. The bounding box tagged by Amazon's mechanical turk is not perfect. A rough bounding box is placed around the spotted word, which was listed by the Google search engine. This imprecise bounding box itself provides an additional layer of difficulty for locating text within the annotated bounding box. The erroneous tagging of bounding boxes has led to lower recognition rate using both open source and proprietary OCR engines.

3.6.4 Born-digital 2011 data set

In born-digital images, the text has been placed by an user in an interactive fashion through a software. Hence, the system generated fonts are only used to create the text pixels. Thus, when the recognizer uses a similar font, the recognition rate can be higher for born-digital images. This is evident from the fact that the baseline recognition result itself is 63%. This data set has better word boundary definition than others. A background margin of four pixels exists around the text bounding box to provide the context of the image.

BDI 2011 test set consists of 918 word images. Only one participant competed in the ICDAR 2011 word recognition competition. Abbyy Fine Reader (applied on the image without any processing, but software set to low resolution image mode) was used as the baseline method by the competition organizer to compare the performance of the submitted algorithm. Since only TH-OCR algorithm [50] competed in the competition and could not beat the baseline method, it was mentioned as a honorary entry.

3.6.5 ICDAR 2011 data set

This data set given for ICDAR 2011 Robust reading challenge Task 2 is almost a subset of ICDAR 2003 data set, where repeated words have been removed and a few additions have been made. All the images removed are from scene images and are not considered either in the testing or training sets of ICDAR 2011 competition. The test set consists of 716 word images, which do not have any background pixels around the word boundary, unlike the born-digital 2011 data.

3.6.6 Born-digital 2013 data set

This data set was used in the ICDAR 2013 competition [30]. This data set is superset of earlier Born-digital 2011 data set. There is an improvement of fifty percent in the number of images compared to the previous edition. The state-of-the-art recognition rate is 82.2%. This data set consists of 1439 word images.

3.6.7 ICDAR 2013 data set

As mentioned in the competition section, this data set is combination of the test and the training images of ICDAR 2003 data set [30]. A background of four pixels is present to have similarity with Born-digital 2013 data set. This data set consists of 1095 word images.

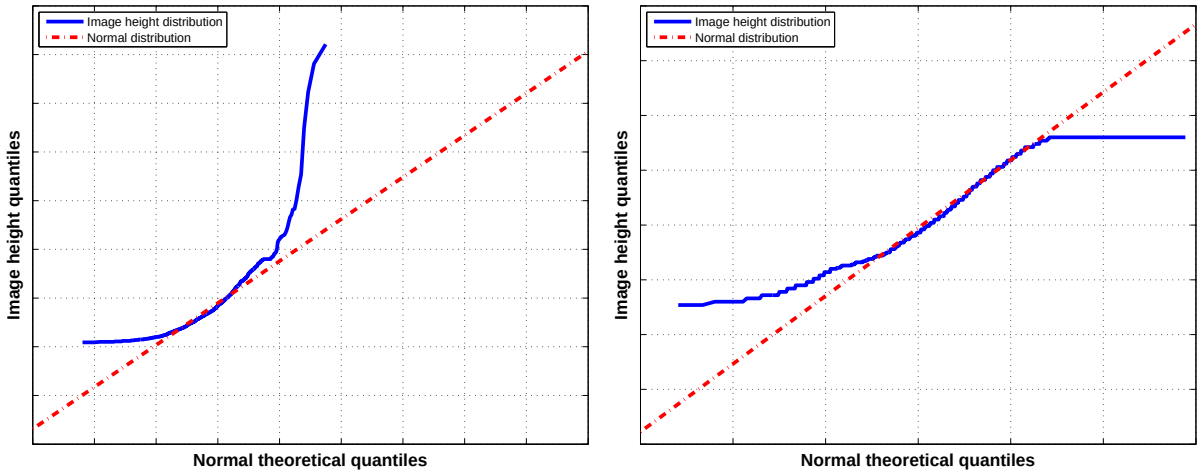
3.6.8 Preprocessing of word images

Each image in a data set is preprocessed before passing it to the segmentation stage and post-processed before passing it to a recognition engine. These processing steps are described in this and the next sub-section.

Apriori, we do not have any kind of information on the stroke width of characters in the data set but have access only to the height and width of the word images. A normality test [19] was conducted on the heights of the images in the data set. Figure 3.11 shows the Q-Q plots of image heights for ICDAR 2011 data set. A close observation of Figure 3.11(i) reveals that the image heights in the data set are not normally distributed. So, image scaling is performed to normalize the image heights in the data set. Images are scaled by bi-cubic interpolation preserving the aspect ratio. In order to minimize the variance in the stroke width, the height of an image is modified to lie within a range. The height range is calculated from the normality test. Accordingly, the rules for rescaling are:

- Rule 1: If the height of an image is less than 60 pixels, rescale by a factor of 3.
- Rule 2: If the height lies between 60 and 180 pixels, do not rescale.
- Rule 3: If the height exceeds 180 pixels, scale it down to a height of 180 pixels.

After height scaling based on these rules, the Q-Q plot of modified image heights is again shown in Figure 3.11(ii). The image height distribution plot is approximately close to normally distributed data set, except for a little deviation at both the ends.



(i) (ii)

Figure 3.11: Q-Q plots of image heights for ICDAR 2011 data set. The actual image height quantiles are plotted against the normal theoretical quantiles. (i) Before height normalization of images. (ii) After height normalization of images.

al OFFI

Figure 3.12: Foreground-background ambiguity. For the binarized images shown, the background is broken, where the text connected components touch the boundary of the word image. This creates ambiguity for the OCR in determining the text polarity. Recognized results obtained from the OCR are also shown.

3.6.9 Post-processing of segmented images

Figure 3.12 shows some sample segmented images. Here, black and white pixels denote background (0) and foreground (1) respectively. Several text components touch the boundary. Such segmented images result in wrong recognition when passed through an OCR engine, since both foreground and background pixels touch the image boundary. To eliminate the foreground-background ambiguity in dealing with such images, the polarity of the background and foreground pixels is detected. Polarity of the text is inverted and/or background padding is then carried out, as required.

Text polarity is detected by examining the following three conditions:

- Is the ratio of number of white pixels along the boundary of segmented word image

IttentitIO (i) ROGERS (ii)

Figure 3.13: Post-processing the segmented image by padding background pixels to eliminate foreground-background ambiguity. (i) The text connected components touch the boundary. (ii) Foreground and background are clearly separated after background padding, leading to improved OCR results.

to the total length of boundary greater than 0.5?

- Is the ratio of number of white pixels on the vertical sides of the segmented word image to the total length of the side walls greater than 0.5?
- Is the ratio of maximum widths of 'white' to 'black' connected components in the segmented word image greater than 1?

If two out of these three conditions are true, then polarity needs to be inverted, after which the text pixels will be white.

During segmentation, uneven illumination may cause salt and pepper noise in the output image. Hence, median filtering, with a structuring element of size 5x5, is performed on the segmented images. The images rescaled using Rule 1 are excluded from median filtering, since they are of low resolution and filtering may degrade the segmented image.

To prevent the text connected components from touching the boundary of the image, background pixels (zeros) are padded, vertically by half the number of rows and horizontally by half the number of columns. An example is shown in Figure 3.13. In Figure 3.13(i), the text connected components touch the word boundary. After background padding, foreground and background pixels are clearly distinct. Post-processed word image after background padding is then fed to a recognition engine. The improvement in word recognition due to this post-processing step is clearly seen.

For word images from BDI 2011 data set, after segmentation, components touching the boundary are removed. Figure 3.14 shows an example of such component removal. Three non-text components, which can be identified in the segmented word image, are

Figure 3.14: Removal of non-text components touching the boundary from born-digital images. Segmentation of a sample word image from born-digital 2011 data set using PLT. Few non-text components (the characters of neighboring words) appear as broken along the boundary.

removed before any post-processing on the segmented image. The non-text components touching the boundary crop up due to the retaining of four boundary pixels on all sides of the born-digital word image.

3.6.10 Benchmarking for upper bound on recognition rate

In order to investigate the recognition rate that can be achieved when an ideal segmented word image is passed to an OCR, analyzed data sets are semi-automatically segmented to create ideal segmented word images. An open-source MAST-CH toolkit [39] developed in MILE laboratory is used for manual segmentation of word images to create ground-truth at pixel level for these data sets (see Appendix A for more details on the toolkit). The earlier version of MAST toolkit was designed to annotate scene images at the pixel level [31]. Additionally, character segmentation from word images is included in the new version. These ground-truth images are passed to an OCR and the recognition accuracy is recorded as benchmark [40].

Each word in the data set has been appropriately segmented in such a way that there is minimal visual distortion with respect to the original image. These segmented images are available for download from MILE website [54]. In all the data sets, only the testing set is considered for the segmentation and recognition experiments. One can possibly improve character segmentation using word images from the training set.

Table 3.2 compares the word recognition rates (WRR) for images segmented by MAST-CH, MAPS and NESP methods with the best results in the literature and baseline - for the seven publicly avasilable word image data sets. The reported benchmark results and those of NESP and MAPS methods have been obtained using Nuance Omnipage OCR

Table 3.2: Comparison of word recognition rates (WRR) of images segmented by MAST-CH, MAPS and NESP methods with the best results in the literature and baseline - for the seven publicly available word image data sets. The baseline results have been obtained by running Omnipage on the word images without segmentation. One of the best reported results [85] uses a limited lexicon and three others [59, 60, 65] use a synthetic custom lexicon, derived from the ground-truths of the respective test sets.

Data sets	ICDAR 2003	PAMI 2009	SVT 2010	BDI 2011	ICDAR 2011	BDI 2013	ICDAR 2013
Benchmark (MAST-CH+ BkGnd padding)	83.9	89.3	79.6	88.5	86.7	—	—
NESP method	**66.2**	80.9	35.2	79.4	**72.8**	81.7	65.9
MAPS method	64.5	80.0	39.6	**82.8**	71.7	**83.8**	66.0
Best result in the literature	61.1 [60]	**86.1** [85]	**73.6** [59]	61.5 [29]	56.4 [65]	82.2 [30]	**82.8** [30]
Baseline (Omnipage)	41.0	50.2	27.7	63.0	31.4	61.0	45.3

on the pixel level segmented word images. Definitely, the numbers may slightly differ if any other standard OCR is used and hence the benchmark results reported indicate a rough level of recognition that can be achieved, rather than the exact maximum value attainable in current circumstances. The baseline results reported have been obtained by supplying the raw, coloured word images to Omnipage OCR without any preprocessing or segmentation. Further, some of the results compared from the literature have been obtained using custom lexicons, synthetically created from the ground-truth. The results reported on NESP and MAPS methods and MAST-CH segmented images have been obtained without the use of any such custom lexicons.

The result reported by Mishra et. al [60] is based on only 829 images, a subset of ICDAR 2003 test set. Hence, the reported result is scaled to the total number of images in the test set for comparison in Table 3.2. Further, the net performance of 61.1% has been obtained using a synthetic custom lexicon derived from the ground-truth of the test set, whereas the performance of 64.5% by MAPS method, 66.2% by NESP method and the benchmark result of 83.9% have been obtained without the use of any such lexicon.

This is resorted to, because we think that, in a real application, it is unlikely that such customized lexicons will be available or practicable.

In Table 3.2, the benchmark recognition rate of 89.3% on PAMI 2009 data set is the highest among all the data sets. As compared to the best result of 86.1% obtained by using a limited lexicon, MAPS and NESP segmentations achieve a reasonable performance of above 80%, without the use of any lexicon.

Mishra et. al show a high recognition rate of 73.6% by top-down approach with higher order language priors on SVT 2010 data set [59]. However, this result is based on the customized synthetic lexicon for each word, built from the test word ground-truth. With a limited lexicon, one can hit upon the proper word more easily than with full lexicon, as discussed by Weinmann et. al [85]. Mishra et. al use bi-gram probabilities extracted from this custom lexicon to improve recognition.

The poor results of 35.2% and 39.6% obtained by NESP and MAPS methods, respectively indicate clearly that our methods are not suited for SVT data set; the bounding boxes are much bigger than the enclosed words in many cases, and hence, the key assumption made by the MAPS method that the middle line contains both foreground and background information is not satisfied and the number of non-text pixels increases with larger bounding boxes resulting in improper segmentation by NESP method. If bounding boxes had been properly defined, it might have resulted in a higher recognition rate. Further, the images also contain motion blur, which cannot be adequately handled by the segmentation approaches used by the MAPS or NESP methods.

The resolution of born-digital word images are low compared to scene word images. Further, they are affected by anti-aliasing. Due to these two factors, characters in the word images merge when a global threshold is applied. Power-law transform is applied to remove the affect of anti-aliasing. By varying the γ value in the power-law transform, it is shown that the merged characters can be split. Thus, the best recognition result of 82.9% is obtained for PLT method [44]. The performance of MAPS method is very close, with a value of 82.8%. As against this, the best reported result (61.5%) in the ICDAR 2011 competition is less than the baseline performance of Omnipage (63%) on the raw

images. The performance of our methods (71.7% and 72.8%) on ICDAR 2011 data set far exceed that of the best result in the literature of 56.4% (with a custom lexicon).

The recognition rate of 83.8% obtained for MAPS method is far better than PhotoOCR [30] and the recognition rate of 81.7% obtained for NESP method is comparable on born-digital 2013 data set. The recognition rate of 66% and 65.9% for MAPS and NESP methods, respectively, is far behind the state-of-the-art PhotoOCR [30] on ICDAR 2013 data set. PhotoOCR performs over segmentation of word images, over-segmented patches are classified as characters by trained deep neural network, and word for each image is constructed using n-gram and local beam search.

A four pixel margin is provided for contextual information for each word image in born-digital 2013 and ICDAR 2013 data sets. The addition of background pixels inadvertently for each word image affected the performance of MAPS and NESP method during the ICDAR 2013 competition period [30]. The definition of four pixel margin is not uniform across word images in the data set and the background margin for some images is improper which was observed while individual images of these data sets were examined at the segmentation level. The results reported for born-digital 2013 and ICDAR 2013 data set are after removing the four pixel margin. The recognition rate for MAPS method on born-digital 2013 data set crossed the state-of-the-art method by removing background margin.

3.7 Lexicon based correction for comparison

Searching an exact word from a vocabulary, covering up to a million words, is time consuming. If constraints are applied on the vocabulary, then the search list is reduced drastically.

Hangman is an English word guessing game. A random word from a category is drawn and a player has to guess the word. Animals, fruits and numbers are a few examples for the category. The number of letters is fixed for a game. At the first stage, the number of letters as a single constraint reduces the number of possible words in the category. In successive stages, as and when a guessed letter is proper, the position(s) of the letter(s)

in the word is/(are) revealed that further reduces the search. In a few steps, a player reaches the solution, if successive guesses are correct.

Jumble letters is another English word game. A player has to unscramble letters presented in a game. Here, the number of letters and the possible letters in the word are fixed, except for their positions. All permutations can be used to search for an exact word from the list. N-gram statistics can avoid wrong permutations to form a word and help in reducing the word list.

Hangman and Jumble letters are educational games to test the vocabulary of a person. If one wants to develop a method to solve these games, edit distance can be used as a metric to sort the words in the list. Shortest edit-distant word may be picked as the right word.

The number of constraints for Jumble letters makes the game easier than Hangman. In a segmented image, a few letters may be broken or merged. Broken or merged letters result in wrong recognition, and the method may miss the possible word. Based on the confidence of other recognized characters, one can use a constrained lexicon to improve the recognition of the word.

Wang and Belongie use a custom lexicon for recognizing a word [83]. A list of 49 random words are presented along with the right word as the customized lexicon for each word image. This task is easier than Jumble letters, but the word image is not easier to segment. Ho et. al [20] use the knowledge on the occurrence of characters within words in the lexicon for recognition of degraded words. Word images are degraded either during capture or in the generation process. If a few parts in a word image are segmented properly, then a constrained lexicon can boost the recognition of words.

Object recognition by humans is considered to be more often top-down with information being fed from primary visual cortex to lateral geniculate nucleus (LGN) cells using the situational context [3]. Consider the process of recognizing a word from an image, where some letters are badly degraded beyond recognition. Exact location of other letters in the word and a person's domain knowledge reduces the search time in figuring out the word. A closely resembling word derived from knowledge simplifies the searching process.

Table 3.3: Comparison of word recognition rates (WRR) of images segmented by MAPS and NESP method with and without lexicon for the seven publicly available word image data sets. A synthetic custom lexicon is used for SVT data set and a limited lexicon for other data sets, both derived from the ground-truths of the respective test sets.

Data sets	ICDAR 2003	PAMI 2009	SVT 2010	BDI 2011	ICDAR 2011	BDI 2013	ICDAR 2013
NESP method	66.2	80.9	35.2	79.4	72.8	81.7	65.9
NESP + Lex	**74.9**	**92.1**	56.4	88.2	**80.5**	89.1	76.6
MAPS method	64.5	80.0	39.6	82.8	71.7	83.8	66.0
MAPS + Lex	74.2	88.8	63.5	**89.9**	79.2	**90.0**	76.7
Best result in literature	61.1	86.1	**73.6**	61.5	56.4	82.2	**82.8**

In order to make a fair comparison with the results in the literature, we also invoke the use of a finite lexicon. However, unlike most researchers, we use a fixed lexicon for all the word images of a particular data set. This lexicon is derived as the union set of all the ground truth words of the test set. Edit distance is measured between the recognized word and the lexicon. Words in the lexicon are sorted in ascending order of edit distance. The word from the list with the smallest edit distance is declared as the lexicon based recognized word. Edit distance can be said to approximately simulate the word recognition process by a person. For each data set, a data set specific lexicon is built, except for SVT data set, where a custom lexicon of 50 words is available for individual images. Table 3.3 lists the performance of NESP and MAPS methods on each of the seven data sets, with and without the use of lexicon.

Wang and Belongie approach of using lexicon to improve the recognition rate of a word image data set works, provided the lexicon used is realistic [83]. Several papers have been published using a similar theme [60, 84]. However, one cannot expect such a custom lexicon whenever word images are added. Thus, the custom lexicon dependent methods cannot generalize and require manual update of the lexicons with the addition of every new word image.

3.8 Discussion

Each of our processing steps plays an important role in improving the recognition rate on ICDAR data sets. The performances of NESP and MAPS methods exceed those of others. These methods are bottom-up approaches in that image statistics is used while choosing the plane with maximum discrimination in NESP method and the middle row in MAPS method.

The definition of boundaries for cropping word images is not uniform across the different data sets. For example, the cropped images in ICDAR 2011 data set are tight or closely bounded, which sometimes causes the characters to touch the boundary. In SVT 2010 data set, each cropped word image has non-text pixels enclosed to form a bigger boundary for the word. The recognition rate of MAPS method is better for such images than NESP method due to the better estimation of background statistics.

NESP method successfully countered the adverse effects of low illumination and low resolution of word images. The plane selection criterion is improved by power-law transformation. However, it is not affected by image scaling.

The major factor in MAPS method is handling illumination variation during the segmentation stage. This is achieved by picking up the middle row as the sub-image for segmentation. Thus, the improvement in segmentation boosts the word recognition rates. To verify the validity of our assumption of minimal degradation in the middle line of an image, middle line segmentation results are compared with the ground-truth for each data set [39]. It is found that nearly 90% of the images in the data set match with our segmentation outputs.

Comparison with the lexicon dependent methods is to show that the methods based on segmentation is better. One may ask a question as to whether any new word image can be recognized. The answers is probabilistic in nature. If the n-gram statistics obtained from the limited lexicon satisfies the new word, then the word image may be correctly recognized. Since the lexicon tries to fit the word image with its dictionary, this approach is not feasible always. We proposed NESP and MAPS methods that are useful in building a generalized word recognition system. These segmentation methods use the word image

statistics and do not depend on the training data. There is no requirement for a lexicon during segmentation.

The choice of one of the proposed methods of segmentation for a given scene image is not straightforward. For scene images, where both background and foreground are fairly uniform, PLT and NESP are better, since they address the binarization of the whole image globally. However, when the degradation is more and non-uniform, if one can assume that the middle line is relatively less degraded, then MAPS is a better option for such images. However, if the question is which of them is the best in general, then the number of images used to evaluate the segmentation methods should be very large (close to a million) to arrive at a conclusion. We do not have a single dataset in the field, which consists of a million images (huge dataset). We are the first to cover the seven datasets and report it in this thesis and the number of images used in the analysis is around 5000. We have demonstrated that the performance of the three proposed methods are reasonably comparable on the tested databases. It may take several years to create a sufficiently large dataset. All the methods proposed in the literature should then be evaluated on this created large dataset to determine the best method for segmentation.

Figure 3.15 shows two sample word images, where the proposed method fails to recognize the word correctly. Strong illumination and low contrast appear in the middle row, which violate our assumption, resulting in failure of the MAPS method in those images. Artistic characters also pose a challenge; even if the segmentation is proper, the recognition is poor due to the unusual nature of the font used (unfamiliar to the OCR).

3.9 Conclusion and future work

We have proposed elegant methods for segmentation of text pixels from camera-captured and born-digital word image data sets. The results in Table 3.2 show that it is not always a trivial task to recognize the text, even from a manually segmented word image. This could be due to artistic and hand-written font, severe degradations and/or varying stroke width in the word image. The performance of our methods on three of the data sets far exceeds the performance of the other techniques. This is achieved by breaking the whole

Figure 3.15: Word images, for which the proposed methods fail to recognize the words.

segmentation and recognition chain into constituent parts.

Approximately, an additional 10% improvement is observed using a lexicon. With the use of limited lexicon, the comparison with the techniques in the literature becomes fair and our techniques have the best performance for the fourth dataset also (PAMI 2009). The use of lexicon reduces the edit distance measure, which is used as the parameter for performance evaluation in ICDAR 2013 competition. It is observed that the edit distance for the state-of-the-art algorithm on ICDAR 2013 competition data sets is less compared to similar recognition rates.

One has to incorporate uniformity in character stroke width for further improvement in recognition rates. However, in this work, we have not performed any direct operation to have uniformity of stroke width of characters; only normalization of the image heights is carried out on each data set.

NESP method picks the right plane and the right value for gamma to process a word image. The recognition rates of NESP method on ICDAR 2003 and 2011 data sets for different planes and different values of gamma have been reported in [43]. MAPS method explored two extreme computational methods. The variation in the recognized results with different varieties of the method is less than 1% as reported in [41].

In our future work, we plan to improvise on the MAPS method by (i) using multiple rows at the sub-image segmentation stage to avoid dependency only on the middle line; (ii) including colour information at the classification stage; (iii) estimating accurate stroke width to reduce the effect of degradations caused to text pixels. We shall also attempt to combine the best aspects of NESP and MAPS methods.

In this chapter, only the segmentation of a word image is considered, since good OCR engines for Roman scripts are available. In the case of Indian scripts, a few OCR engines have been developed. The aim of those engines is to analyze the page layout and recognize the word in a segmented line. The performance of these engines is similar to that of Roman

script OCR engines but the recognition rate is low when word recognition is carried out on a cropped scene image (comparison in similar context). One of the reasons may be that the Roman OCR engines have been trained with multiple fonts, which may not be the case with Indic script OCRs.

In the next chapter, we apply the above segmentation method on camera captured scene images containing Kannada text. Since good commercial OCRs are not available for Kannada, a classifier is also trained to recognize Kannada characters from segmented scene images.

Chapter 4

Kannada word recognition from scene images

Summary

A method is proposed for feature extraction and classification of manually isolated characters from scene images. Characters in a scene image may be affected by low resolution, uneven illumination or occlusion. Discrete cosine transform is used to extract features from characters after translation and scale normalization. We have evaluated our method on the test set of Chars74k dataset for Roman and Kannada scripts consisting of handwritten and synthesized characters, as well as characters extracted from camera-captured scene images. Only synthesized and handwritten characters from this dataset are used as training set and nearest neighbor classifier is used for classification in our experiments. Since the number of classes for Kannada characters is large and the labeling is erroneous and redundant, MILE laboratory Kannada OCR samples are used for training the classifier. The recognition accuracy has a big leap with MILE laboratory training samples. Recognition of the Kannada words from the MRRC data set is also performed using the same classifier. Segmented word images are decomposed into characters or symbols before classification.

Figure 4.1: Kannada and English image samples from Chars74k dataset [77].

4.1 Introduction

Character recognition in a non-degraded document image is a solved problem and several open source and commercial recognition engines are available as OCR [1, 67, 76]. However, there is still enough room for development of novel methods to tackle the new problems encountered in scene images. In general, most of the scanned documents are in gray scale, while camera-captured scene images have characters with additional colour information and complexity in the form of degradations. Figure 4.1 shows a few samples of manually cropped characters or symbols from Chars74k dataset [77]. To counter the degradations and to improve recognition, scene images are segmented and features are extracted from the characters and a classifier is built.

4.2 Description of Chars74k dataset

Chars74k dataset [13, 77] has been used for our experiments, which contains both English and Kannada characters. It contains nearly seventy four thousand characters. Most of the characters for English (63K) are generated artificially using synthetic fonts and the remaining have been obtained by manual cropping from camera-captured scene images. Kannada is one of the ancient Dravidian languages, with a history of more than two thousand years. Table 4.1 gives the details on the number of classes and samples for Roman and Kannada scripts in Chars74k dataset. The training dataset for English has 254 synthetic fonts, and there are four samples in each font, corresponding to normal, bold, italic and bold-italic types. In addition, there are 55 handwritten samples for each of the

Figure 4.2: Kannada and English handwritten samples from Chars74k dataset [77].

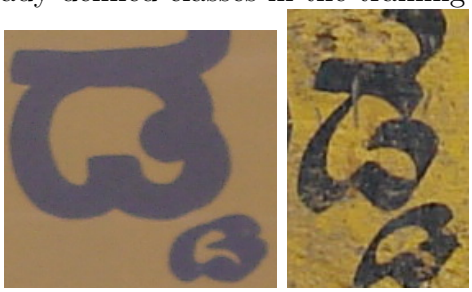
Table 4.1: The number of character classes, the number of samples per class and the total number of samples in Chars74k dataset for *Fnt*, *Hnd* and *Img* categories for Roman and Kannada scripts

Category	Description	No. of classes	Samples/ class	Total No. of samples
English *Fnt*	Synthetic fonts	62	1016	62992
English *Hnd*	Handwritten images	62	55	3410
English *Img*	Segmented from scene images	62	—	12503
Kannada *Hnd*	Handwritten images	657	25	16425
Kannada *Img*	Segmented from scene images	990	—	5135

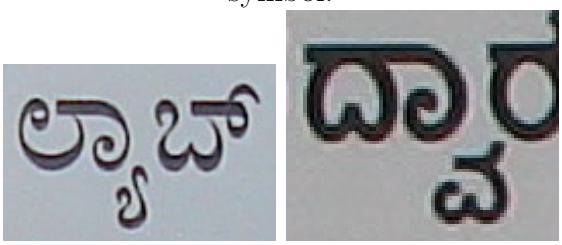
62 classes (Uppercase, Lowercase and Hindu_Arabic numerals). For Kannada, there are 25 handwritten samples for each of 657 classes. Figure 4.2 shows a few of the handwritten samples from Chars74k dataset for both Kannada and Roman scripts. Variation in handwritten characters is much higher than in font based (synthesized) characters. Further, the training with handwritten characters gives rise to the possibility of recognizing handwritten characters found in camera-captured scene images. The number of classes present in Chars74k *Img* dataset for English and Kannada are 62 and 990, respectively. Also, the number of samples per class varies in *Img* dataset.

It is to be noted that the Kannada *Hnd* and *Img* datasets have been poorly segmented and casually labeled, without the involvement of people with knowledge in Kannada. Because of this, different samples of the same character have been given different labels in some cases (see Figure 4.3(b)). In other cases, a complete word or a set of characters have been segmented as a character and given a distinct character label (see Figure 4.3(c)). Because of such imperfections, the number of classes in Kannada *Hnd* category is 657 whereas it is 990 for the Kannada *Img* category.

(a) Images labeled as a separate new class in the testing set even though it is part of the already defined classes in the training set.



(b) Two separate new classes are added in the test set for distinct samples of the same symbol.



(c) Classes are defined for words.

Figure 4.3: Erroneous tagging of Kannada class labels in Chars74k test set [77].

4.3 MILE Kannada OCR training samples

The difference in the number of classes between Kannada *Hnd* and *Img* is huge, which makes Kannada symbol recognition more complex than that of English. The symbols or class labels in Chars74k dataset are tagged erroneously.

Commonly observed errors that exist in the data set are shown in Figure 4.3. This tagging error forms a barrier in achieving the required recognition rate. Three hundred and eighty seven unique symbols are designed for Kannada character or word recognition in MILE laboratory. The training samples of these symbols are used in Kannada word recognition. The errors shown in Figure 4.3 can be removed by tagging the characters or words in Kannada Unicode.

4.4 Related work

Our main interest lies in recognition of Kannada characters from natural images [45] and this is the first public dataset that makes it possible to perform experiments on Kannada character recognition. On the theme of bag-of-visual-words technique, feature vectors are extracted to build visual vocabulary [13]. Six different types of local features are calculated, namely, shape context [7], geometric blur [8], scale-invariant feature transform (SIFT) [51], spin image, filter response and patch descriptor. These features generally tend to capture the edge information of a shape to build a visual vocabulary. The feature vectors calculated from the images are local and most of them are dependent on the edges of the training characters.

In our experiments, 2-D transform based descriptors are used to extract the features. An advantage of these features is that the length of the feature vectors used for training is less than that used in other methods.

4.5 MAPS based binarization of scene character images

The colour images are converted to gray and the gray values of pixels are used for binarization. Here, MAPS method is used to segment manually cropped characters. In this approach, the middle row of the image is segmented using the Min-Max method and Min-cut/Max-Flow algorithm is used to segment the whole image. Figure 4.4 shows the plot of middle row gray values and the result of Min-Max method for the character image sample shown in Figure 4.5.

Figure 4.5 shows the superior binarization result of MAPS method compared to those of global thresholding by Otsu [68], Niblack's local thresholding [63] and Sauvola's [74] methods for a sample Kannada character image from Chars74k *Img* dataset. A window of size 32x32 is used for local thresholding algorithms. Noise in the form of stray pixels is observed in other methods due to the presence of texture in the background.

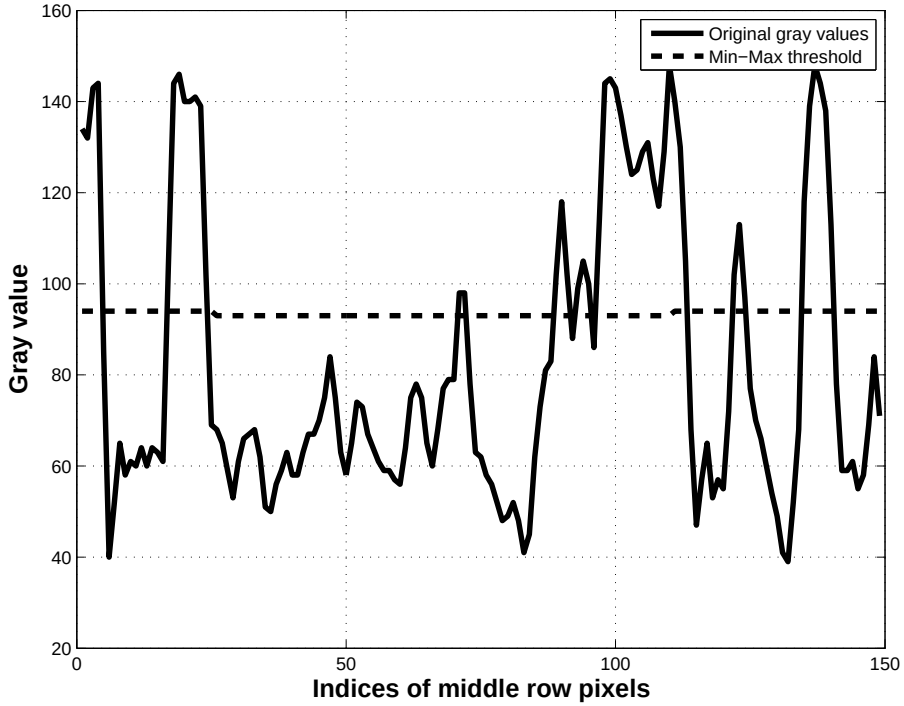


Figure 4.4: Segmentation of the middle row of a degraded character image shown in Figure 4.5 using Min-Max method.

4.6 Exploration of transform features

Here, the effectiveness of orthogonal transformations used in JPEG and MPEG standards [27, 61] is explored. Discrete cosine transform (DCT) is used in JPEG image compression. DCT features have been used for the recognition of machine printed Tamil characters by Aparna and Ramakrishnan [6], Kannada characters by Vijay Kumar and Ramakrishnan [37, 38], Arabic word recognition by AlKhateeb et.al [4] and script identification by Pati and Ramakrishnan [69]. Angular radial transform (ART) has been used by Kasar and Ramakrishnan [34] as a feature vector for matching points while mosaicing images. Lavrenko et.al [47] and Rath et.al [71] have used Discrete Fourier transform (DFT) features for word spotting in historical handwritten images.

Figure 4.5: Comparison of outputs of different binarization techniques. (a) A sample image. (b) Gray scale image. (c) Classification of pixels using energy minimization function (MAPS). (d) Otsu's global thresholding. (e) Niblack's local thresholding. (f) Sauvola and Pietäikinen's local thresholding.

4.6.1 Discrete cosine transform

Discrete cosine transform was originally suggested for energy compaction [27]. Only a third of the coefficients have significant values, which can be retained as the feature vector of a character. The two dimensional DCT equation is defined as [37],

$$\begin{gathered} C(u,v) = \alpha(u)\alpha(v) \\ \sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2M}\right] \end{gathered} \tag{4.1}$$

for $0 \leq u \leq (N-1)$,$0 \leq v \leq (M-1)$ and

$$\alpha(k) = \begin{cases} \sqrt{\frac{1}{P}} & k = 0 \\ \sqrt{\frac{2}{P}} & otherwise \end{cases} \tag{4.2}$$

where, $k = u, v$ and $P = NorM$.

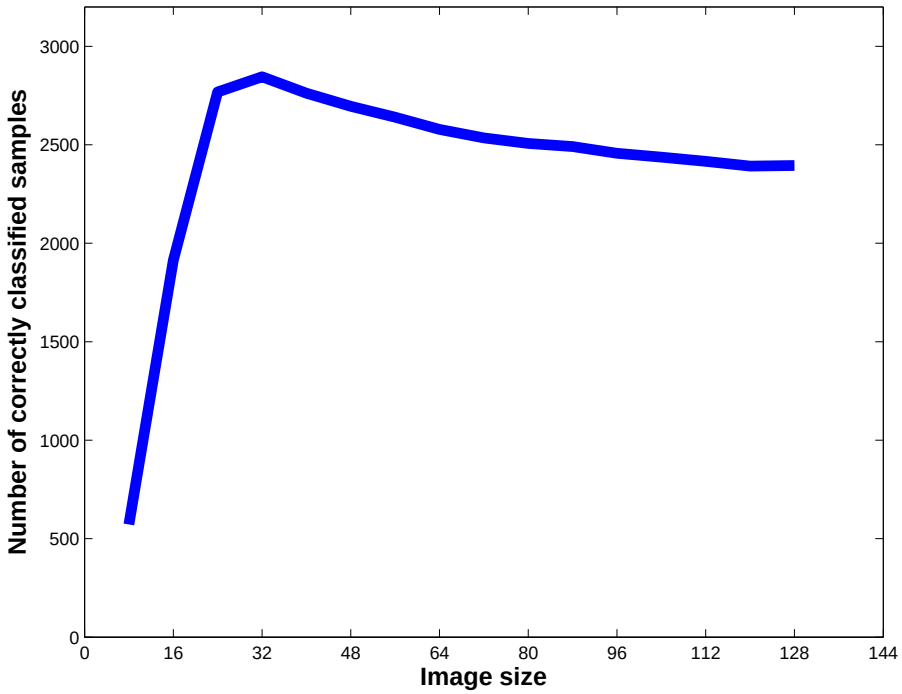


Figure 4.6: Plot of number of correctly classified Kannada character samples as a function of normalization size of the image.

There arises a need to determine the proper size for normalization and further, estimation of feature vectors for training. In our experiment, different normalized sizes of binarized image, in multiples of 8, from 8x8 to 128x128 are explored. Figure 4.6 shows the plot of number of correctly classified handwritten Kannada test symbols as a function of normalization size of the image. A peak occurs at 32x32 and from 40x40 to 128x128, there is minimal variation in the number of correctly classified samples.

The connected components of the input image are scaled to 32x32 size for unit normalization after the image is binarized. 153 DCT coefficients are retained, which is equal to 15% of the coefficients in zigzag sequence, similar to JPEG method [27]. 8x8 block-based DCT is also performed and 15% of the coefficients are retained in every block. This method of block processing is referred to as 'block DCT' and the other as 'global DCT.' Figure 4.7 shows the original and the two reconstructed images from the retained block and global DCT coefficients, for a sample scene character image.

Figure 4.7: Original image of a Kannada character 'th' and images reconstructed using global DCT and block DCT.

4.6.2 Recognition by nearest neighbor classifier

Since the number of training samples available per class is very small (5 to 25) for *Hnd* and *Img* datasets, it is not possible to use training data intensive classifiers such as artificial neural network (ANN) and support vector machine (SVM). Hence, we resorted to nearest neighbor classification.

4.7 Component segmentation by VP, CCA and overlap threshold

A character in any script can generally be separated into three horizontal regions namely ascender, middle and descender region. Kannada characters are generally labeled as 'base', 'ottu', 'dheergha', 'matra' or 'numeral'. The base symbols of Kannada are modified by 'ottu', 'dheergha' and 'matra'. Kannada 'ottu's appear in the descender region and others in the middle region.

4.7.1 Vertical projection

Vertical projection (VP) of text pixels in the image is carried out and a small threshold splits the word into segments. Figure 4.8 shows two manually segmented sample word images in the top row and red vertical lines in the bottom row shows the splitting of words based on the threshold into segments with one or multiple connected components.

Figure 4.8: Top row: Two word images, which are manually segmented. Bottom row: Red lines split the word images into components, which are obtained by a small threshold on the VP.

Figure 4.9: The segmented image has a base and a ottu component. The split occurred between the base and ottu components due to low overlap. The base and other components are shown in the middle and last columns, respectively.

4.7.2 Connected component analysis and overlap threshold

The first segment of first image in Figure 4.8 has multiple components, even though it is a single base character. The last segment of second image has two components and they are base and ottu symbols. In order to separate base and ottu from these segments, connected component analysis is performed on each segments.

The bounding box information of each CC in the segment is calculated and it is used to obtain pairwise overlap between the CC's. The pairwise horizontal overlap is normalized by the width of the considered CC which may split the base and ottu components. A threshold of 1.5 is applied on this normalized overlap to split the components in our experiment. Figure 4.9 shows the split of base and ottu components into two separate parts for the sample segmented image shown in Figure 4.8.

4.8 Kannada label to Unicode generation

Unicode 6.0 character code charts [81] are available for scripts that are not included in ASCII code. A glyph in a script is represented by one or more Unicode(s). The Unicode range for Kannada is 0x0C80–0x0CFF (hexadecimal numbers).

4.8.1 Construction of labels

Kannada words may have components in all the three (ascender, middle and descender) regions and all possible combination of symbols for Kannada characters are considered to determine the unique labels. The necessary and sufficient symbols used in the construction of characters are termed as unique labels and they are used in Kannada OCR for training the classifier. The samples for these labels have been generated by the members of MILE laboratory. Segmented components from the previous section are classified into these Kannada labels using the training samples. The sequence of generated labels is then mapped to a Kannada Unicode word string.

4.8.2 Mapping Kannada labels to Unicode

A mapping is required for Unicode generation from the label sequence, since combining Kannada labels is not always straight forward. So, a complex mapping is carried out from these labels to Kannada Unicode. The flowchart of mapping labels to Unicode is shown in Figure 4.10. A check is carried out to figure out whether the present label is a numeral. If the check is valid, then numeral Unicode is appended. If the check is not valid, then the checks for ottu, dheergha or matra are carried out. If any of these checks are valid, then the last Unicode may be modified or a new Unicode is appended into the array. A single Kannada grapheme may be represented by one or two Unicodes. An example of a linear mapping is shown in Figure 4.11(a). Further, the order of the Unicode sequence is different from that of the symbol sequence in some cases. For example, see the word given in Figure 4.11(b) for complex mapping. The corresponding Unicode sequence shows that when the symbol ಕ್ (called arkaottu) occurs, its Unicode may be advanced by two or

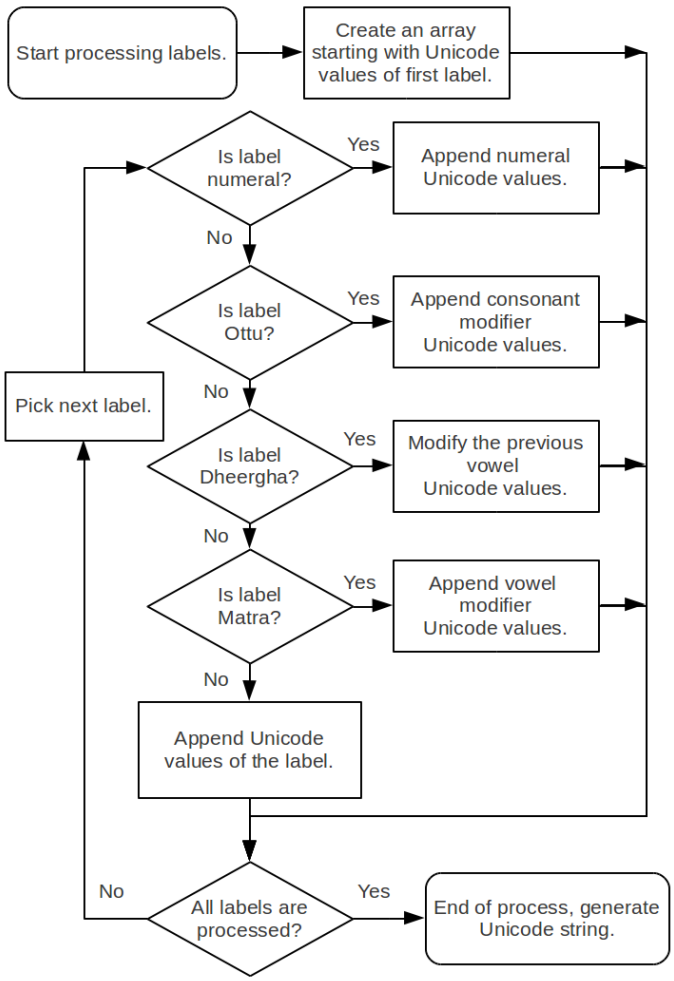
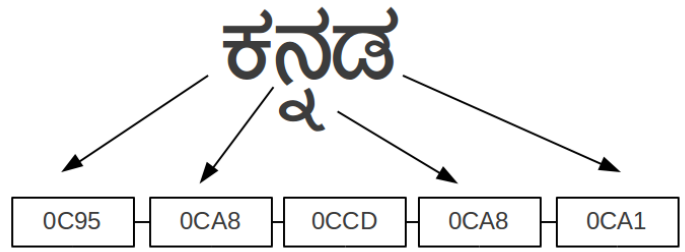


Figure 4.10: The flowchart of Kannada labels to Unicode conversion. The four important sections in the flowchart check the different combinations of previous and present labels that modify the generated Unicode sequence.

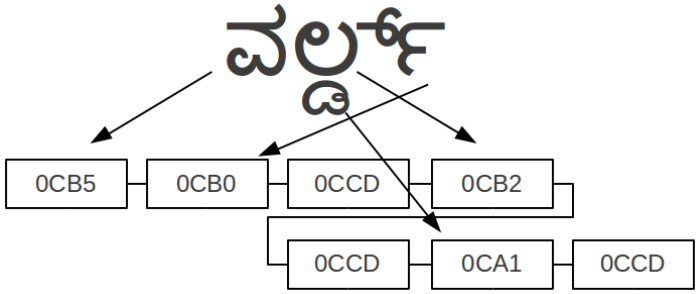
more positions, as shown by the example in Figure 4.11.

4.9 Experimental results on Chars74k dataset

Manually extracted English characters from Chars74k dataset, namely English *Img* dataset, are tested with different types of training samples, namely, synthetically generated, handwritten and cropped scene characters. In all the cases, training samples are binarized using the MAPS method. Features are extracted from the segmented image and feature vectors



(a) Linear mapping.



(b) Complex mapping.

Figure 4.11: Mapping of Kannada symbols to their respective Unicode during the generation of a Kannada word.

for the training samples are stored as a kd-tree to accelerate nearest neighbor classification. Similarly, Kannada characters from Chars74k dataset (Kannada *Img* dataset) are tested with two different types of training samples. One is handwritten Kannada symbols from Chars74k dataset and the other is MILE laboratory Kannada OCR training samples. In both the cases (English *Img* and Kannada *Img*), the test suite provided by de Campos et.al is used for our experiment.

4.9.1 Results using *Img* dataset for training

English *Img* dataset consists of 12503 characters. de Campos et.al [13] created two training sets, namely Chars74k-5 and Chars74k-15, with 5 and 15 training samples, respectively. Fifteen samples per class from English *Img* dataset are used for testing. Wang et.al [83] generated HOG features from the training samples and used nearest neighbor classifier. Table 4.2 shows the results on the test set with two separate columns for Chars74k-5 and Chars74k-15. Even for Chars74k-5 dataset, which has very few training samples, DCT

Table 4.2: The classification results (%) on English *Img* dataset, using *Img* for training. Nearest neighbor classifier is used since the number of training samples available per class is limited (5 for Chars74k-5 and 15 for Chars74k-15).

Feature vector	Chars74k-5	Chars74k-15
Block DCT	**47.7 ± .4**	**58.3**
Global DCT	**47.7 ± .7**	57.9
HOG [82]	45.3 ± 1.0	57.5
MKL [13]	—	55.3
Shape Context [13]	26.1 ± 1.6	34.4
Geometric Blur [13]	36.9 ± 1.0	47.1
Patches [13]	13.7 ± 1.4	21.4
MR8 [13]	6.9 ± 0.7	10.4

Table 4.3: The classification results on English *Img* dataset, using *Fnt* for training. The number of classes is 62 and there are 1016 training samples per class.

Feature vector	Recognition rate (%)
Block DCT	**66.4**
Global DCT	66.3
Shape Context [13]	44.8
Geometric Blur [13]	54.3
SIFT [13]	11.1
Patches [13]	7.8

based classification accuracy is 2% more than the results reported by Wang et.al. using HOG features. The classification results will be higher for case insensitive recognition, as reported by de Campos et.al [13].

4.9.2 Results using *Fnt* dataset for training

English *Fnt* dataset consists of 254 different kinds of fonts with the following type faces: normal, bold, italic and italic bold. This synthesized character dataset is used for training. Table 4.3 shows the results on this test set. Since individual characters are classified, there is no contextual information that can be used. Due to shape topography, some of the numerals and alphabets are classified into other classes.

Table 4.4: The classification results on English *Img* dataset, using *Hnd* for training. The number of classes is 62 and there are 55 training samples per class.

Feature vector	Recognition rate (%)
Global DCT	**46.4**
Block DCT	44.6
Shape Context [13]	31.1
Geometric Blur [13]	24.6
SIFT [13]	3.1
Patches [13]	1.7

Table 4.5: The cross validation results using different features on Kannada *Hnd* dataset, consisting of 657 classes and there are 25 samples per class.

Feature vector	Recognition rate (%)
Global DCT	**33.3**
Block DCT	33.1
Shape Context [13]	29.9
Geometric Blur[13]	17.7
SIFT [13]	7.6
Patches [13]	23.0

4.9.3 Results using *Hnd* dataset for training

English *Hnd* dataset consists of 55 samples per class. For this experiment, only this handwritten data is used for training the classifiers. Table 4.4 shows the classification results for English *Img* dataset. As the number of available training samples per class is less than that in English *Fnt* dataset, the classification result is poor.

4.9.4 Kannada *Hnd* dataset

Chars74k dataset has 25 samples for each of the 657 classes of Kannada handwritten symbols. These samples are split into 12 samples for training and 13 for testing. Figure 4.6 shows the plot of correctly classified samples in cross-validation. The classification accuracies of our experiments are tabulated in Table 4.5 and compared with the reported results.

Table 4.6: The classification results on Kannada *Img* data set, using Kannada *Hnd* dataset for training. The number of classes is 657 and there are 25 training samples per class.

Feature vector	Recognition rate (%)
Global DCT	**11.4**
Block DCT	11.1
Shape Context [13]	3.5
Geometric Blur [13]	2.8
SIFT [13]	0.3
Patches [13]	0.1

4.9.5 Kannada *Img* dataset

Kannada *Img* test dataset consists of 5135 test samples, and is classified using nearest neighbor method. The feature vectors are extracted from Kannada handwritten symbols for training the classifier. As already explained, de Campos et.al have provided only 657 classes in the Kannada handwritten dataset and set aside these as the training set. On the other hand, they have divided the Kannada *Img* dataset into 990 classes and have called it as the test set. Due to this unusually heavy mismatch in the number of classes between the training and test datasets, the classification results can be expected to be very low. Table 4.6 shows the classification results on this dataset.

4.9.6 Re-annotation of test data

The number of classes formed for training Kannada samples is 657, which is large, but the number of classes for testing Kannada samples is 990, which is even larger in Chars74k dataset. This difference in the number of classes yields less classification accuracy. The test class numbered from 658 to 990 were manually checked to know the different symbols that are added into the test set. Some test samples, which belong to the same class in the training set are named as separate classes in the test set (see Figure 4.3), more than once for several samples and same samples are present in several test classes. There is an error in annotating the test samples for the entire Kannada scene symbols in Chars74k dataset. To rectify these issues, all the 5135 Kannada scene symbols were re-annotated as Kannada Unicode words. The test samples of Kannada scene symbols are considered

Table 4.7: The classification accuracy(%) on the cleaned up Kannada test set from Chars74k data set using the training samples from Chars74k dataset and MILE Kannada OCR samples.

Features	Chars74k samples for training	MILE Kannada OCR samples for training
Block DCT	11.1	36.8
Global DCT	11.4	34.3

as cropped words and are recognized. The recognition rate of re-annotated samples are tabulated in Table 4.7. An improvement of 25% is observed with the change in training samples and re-annotation of test samples. The barrier due to erroneous annotation has been removed.

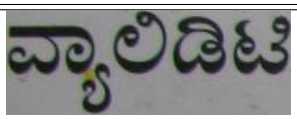
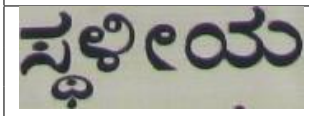
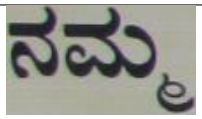
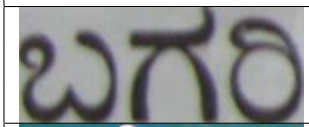
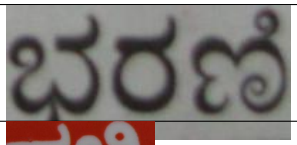
4.10 Word recognition results on the MRRC dataset

In the previous section, the results tabulated were for isolated characters extracted from a scene image. There is necessity for recognizing a word, which may contain a single or multiple characters/symbols. A binarized word image is segmented into respective components, each of which is classified and the recognized labels are used to generate the word. Here, Kannada words cropped from scene images are segmented and recognized. The training samples for Kannada characters/symbols are obtained from MILE laboratory Kannada OCR. As possible results for good word images, Table 4.8 shows the recognized words for some of the scene word images from the training set of MRRC.

4.10.1 Kannada word recognition

The test set used for Kannada word recognition is obtained from Multi-script robust reading competition conducted by us (Appendix B). This test set consists of 243 word images. One-third of the test set has skewed or curved text along with other pixel-level degradations. The Tesseract OCR engine that provides OCR for a few Indian languages also includes Kannada. The recognition rates are tabulated with edit distance in Table

Table 4.8: Recognition results on MRRC training samples using MAPS binarization, block DCT features and nearest neighbor classifier, using training samples of MILE Kannada OCR.

Word image	Recognized word (English transliteration)	Word image	Recognized word (English transliteration)
	೨೦೦೯ (2009)		ಸಂಗಮ (sangama)
	ವಿಜ್ಞಾನ (vijnaana)		ವ್ಯಾಲಿಡಿಟಿ (vyaaliditi)
	ಮೊಬೈಲ್ (mobile)		ಬ್ಯಾಂಕ್ (byaank)
	ಸ್ಥಳೀಯ (sthaLiya)		ನಮ್ಮ (namma)
	ಬಗರಿ (bagari)		ಭರಣಿ (bharani)
	ಅಶೋಕ (ashooka)		ಹಳ್ಳಿ (haLLi)
	ಮಹೋಗನಿ (mahoogani)		ಮಾರ್ಗ (maarga)

4.9. Methods such as benchmark, MAPS, NESP and PLT used for segmentation have been explained in Chapter 3. In all the cases, the recognition rate of block DCT features is higher than that of Tesseract OCR. The edit distance of block DCT is much less than Tesseract OCR in manually segmented word images, indicating that the mismatch between ground-truth and recognized words is less. The plot of the number of words versus edit distance in Figure 4.12 compares Tesseract OCR and block DCT based OCR.

The words recognized by Tesseract OCR and block DCT were individually analyzed for differences. Manually segmented word images are used in the process to understand which OCR is better. A few Kannada words with ottu that were recognized by Tesseract OCR were also recognized by block DCT. Figure 4.13 shows some sample words recognized

Table 4.9: Recognition rate of Tesseract OCR and block DCT on Kannada test samples. The number of words in the test set is 243.

Methods	Tesseract OCR		Block DCT	
	Word recognition (%)	Edit distance	Word recognition (%)	Edit distance
Benchmark	11.1	178.7	12.4	141.1
NESP	5.8	212.3	6.6	200.4
PLT	5.4	210.6	7.8	194.3
MAPS	4.9	209.1	7.4	182.4

Table 4.10: The word recognition rate of Kannada test samples using Tesseract OCR and block DCT, with the use of lexicon.

Methods	Tesseract OCR		Block DCT	
	Word recognition (%)	Edit distance	Word recognition (%)	Edit distance
Benchmark	32.1	155.4	43.2	111.0
NESP	14.8	201.6	25.1	164.9
PLT	15.6	197.6	28.0	158.6
MAPS	17.7	193.0	29.2	147.6

by both the OCR's. Other Kannada words with ottu recognized by block DCT but not by Tesseract OCR are shown in Figure 4.14. From Figure 4.14, it appears that Tesseract has issues in dealing with ottus.

The recognition rate for both Tesseract OCR and block DCT are less due to ottus and thus, the words with one edit distance are more in block DCT. So, a customized lexicon of 243 test words is used. The plot of correctly recognized words against edit distance as shown in Figure 4.12. The recognition rates using the lexicon are tabulated with the edit distance in Table 4.10. One important note is that mutually exclusive words recognized by Tesseract OCR and not by block DCT with lexicon do not contain any word image with ottus.

4.11 Conclusion and Future work

Since the Chars74k dataset had English characters and Kannada symbols, it was used in initial investigation. The feature vectors estimated by de Campos et.al were point-

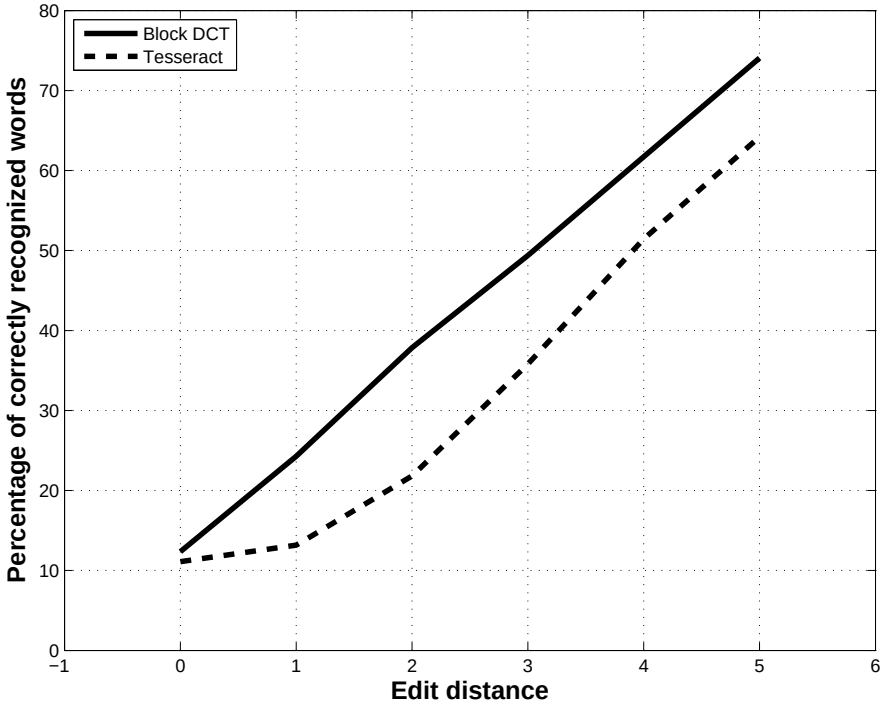


Figure 4.12: Plot of word recognition versus edit distance for Tesseract OCR and our classifier using block DCT OCR for manually segmented word images (Benchmark images). For values of the edit distance of one and more, the gap between the two recognizers is more than 10%.

or edge-based for the Chars74k dataset. In our work, the feature vectors extracted are based on orthogonal transforms. The truncated vectors inspired by JPEG compression are better in all the test cases. Thus, of the features tested, the DCT based descriptors are better than point- or edge-based descriptors for any character/symbol.

MPEG-7 consortium accepted ART as the 2-dimensional shape descriptor for image retrieval [26, 35, 61]. We compared DCT based features against ART features for recognizing English and Kannada isolated characters in [45]. ART performance was poorer than what is reported in the literature. Block DCT has much better performance than ART even while retrieving the low resolution image.

Scene image characters are more difficult to binarize than machine printed or handwritten characters. The classification accuracy obtained indicates that using DCT coefficients as features from binarized image results in higher recognition rates than the other methods. The scene word recognition rate for Kannada words is less with all our methods

Figure 4.13: Common Kannada words with ottu, recognized by both Tesseract OCR and block DCT from manually segmented word images.

Figure 4.14: Kannada words with ottu, recognized by block DCT but not by Tesseract OCR from manually segmented word images.

(MAPS, NESP and PLT), but with a lexicon, it improves by 20%. The classification accuracy is not sufficient to indicate that one among the three proposed segmentation methods (MAPS, NESP and PLT) as better due to the classifier used in recognition is not intelligent enough to capture the quality of segmentation and the number of images used in the evaluation is less (243 images).

A whole character/symbol is used to train the classifier which may not properly classify a CC segment from a degraded scene image. Training a classifier with partial characters may improve the algorithm efficiency in the presence of occlusion and certain degradations.

Chapter 5

Conclusion

5.1 Conclusion

Aids for the blind and unmanned navigational systems need to be able to analyze scene images for the presence of text. This thesis has addressed scene text localization and segmentation, as well as recognition of scene word and character images. This thesis has explored each sub-problem in camera-captured scene image analysis in breadth, but not in sufficient depth. Text localization and segmentation deal with scene or born-digital images, whereas character and word recognition deal with cropped images obtained from scene or born-digital images. Each problem addressed has significant scope for improvement either in dealing with complex non-text area for text localization and text segmentation or with increased number of classes for character and word recognition.

Unless, there is a standardized evaluation procedure, the efficiency of a method for a specific task cannot be ascertained. One of the basic methods is to estimate the computational complexity of a method. Here, the evaluation of a method focuses on its ability to match human score. Usually, the precision and recall measures are used to determine the capability of a method. OTCYMIST method proposed in this thesis was an entry for text segmentation and localization tasks in ICDAR 2011: Robust Reading Competition–Challenge 1, where it won the first place for performance on the text segmentation task. The other methods, namely, PLT, NESP and MAPS are state-of-the-art for five of

the seven datasets analyzed in this thesis for word recognition. All the developed methods were placed in the top three positions in ICDAR 2013: Robust Reading Competition for different challenges and tasks.

OTCYMIST is basically a segmentation method for text in born-digital images, but it is also used for text localization task by grouping segmented components. This thesis focused on born-digital images while describing the OTCYMIST method. However, it is an entry for text segmentation task for scene images challenge in ICDAR 2013: Robust Reading Competition. Minimum spanning tree used to select the segmented components is an intermediate step of Delaunay tessellation, which is a part of page layout analysis used on scanned document images. Since camera-captured scene or born-digital images cannot be assumed to have any specific layout, they are unsuited to build a Delaunay tessellation. OTCYMIST method is used as a baseline method in Multi-script Robust Reading Competition (MRRC), conducted by MILE Lab.

Computer vision inspired features such as histogram of oriented gradients (HOG) and scale-invariant feature transform (SIFT) have been used in the literature to recognize word from the cropped word images. The number of classes for recognition is reduced by combining each set of uppercase and lowercase Roman letters into a single class to lessen the complexity. This thesis proposed another approach to reduce the complexity by breaking the word recognition task into segmentation and recognition tasks.

Prominence is given only to segmentation of a word image in this thesis, since good OCR engines are available for Roman script for recognition. Different segmentation methods have been proposed to segment the cropped word images. PLT method uses non-linear enhancement on gray scale images. NESP method picks the right plane with a value of gamma to process a word image. MAPS method classifies the pixels based on the segmentation result of the middle line. Segmented word images are recognized by Omnipage OCR engine. However, some of the words cannot be extracted perfectly from word images due to the degradations encountered. Though our methods do not make use of any lexicon, their performance exceeds those of all the methods in the literature on four of the seven datasets, in spite of the fact that the latter methods employ lexicons customized for

each word image. Obviously, when similar lexicons are used. There is an improvement of approximately 10% in the results of everyone of our methods.

In the case of Indian scripts, a few OCR engines have been developed as research efforts. The performance of these engines is low compared to the OCR engines for Roman script. Nearest neighbor classifier is used to recognize Kannada characters segmented from a scene image. Vaguely defined class labels in Chars74k data set defied the classifier to achieve its performance. Properly defined class labels available in MILE laboratory were used in the classifier. The training samples are handwritten in Chars74k data set, whereas in MILE laboratory, the training samples were obtained from a machine-printed text. The number of classes defined for the training samples is different. There was an improvement in classification accuracy when MILE laboratory training samples are used rather than those of Chars74k. The experiment was carried out to determine which training samples are better.

For recognizing Kannada words also, nearest neighbor classifier is made use of. Ours is the first attempt on Kannada text segmentation and recognition from camera-captured scene images. Different features (DCT and block DCT) were explored. The classification accuracy for block DCT features is better than that with the other features. There is an improvement of 25% by applying a constrained lexicon.

5.1.1 Major contributions of the thesis

- A system to segment born-digital images with minimum spanning tree modules to determine words and non-words in the image.
- Non-linear enhancement of gray scale image to improve segmentation and hence, the recognition of the data set.
- Middle line analysis for self-training on image itself. Thus, the effectiveness of segmentation is dependent on the properties of the middle row of the image.
- Analysis on the number of class labels for the recognition of Kannada characters.

5.2 Scope for future work

Text extraction system comprises detection, localization and recognition of text. The thesis has explored text localization and text recognition at different depths. Nevertheless, improvements can create a better system. Some of the experiments, which can be carried out as future work, are listed below.

- Image segmentation is approached by splitting the colour channels of an image. If the pixels have values close to gray scale, then only one channel is sufficient for text segmentation.
- Images affected by slowly varying and/or strong illumination are not segmented properly. Illumination variation in the image needs to be identified and corrected while segmenting the image.
- Different word images have characters with different stroke widths. By ensuring uniformity in the stroke width of the segmented characters, the recognition rates may be further improved.
- Multiple rows can be used in MAPS method as input to the sub-image segmentation stage to avoid dependency only on the middle line. Also, the colour information may be made use of at the classification stage.
- If a classifier is built based on partial characters, then characters degraded by occlusion or strong illumination may be detected and processed for recognition.
- One-third of the English words in the test set of Multi-script Robust Reading Competition have skew, slant and curved words. These words in an image can be aligned horizontally and passed to OCR for better recognition.

Appendix A

Annotation of MASTER database

Summary

A semi-automated tool has been developed for annotating multi-script text from natural scene images by either manual seed selection followed by a region-growing process or using available segmentation techniques. If required, polygonal masks are used to add or delete a patch of pixels to or from a segmented word to convert it into valid characters/symbols. The text present in the image is tagged word-by-word. A virtual keyboard interface has also been designed for entering the ground-truth in anyone of ten Indic scripts, besides Roman. The keyboard interface can easily be generated for any new script, thereby expanding the scope of the tool. The ground-truth is represented by a pixel-level segmented image and a '.txt' file, which contains information about the number of words in the image, word bounding boxes, script and ground-truth Unicode. The tool was used to generate the ground-truth of MASTER database and also for the five standard word image datasets.

A.1 Introduction

In a multilingual country like India, it is common to find English words interspersed within sentences in Indic-script documents. Many documents, forms and signboards are generally bilingual or trilingual in nature. Two datasets containing multi-script text are available. First, Chars74k dataset [77] consists of English and Kannada characters segmented from

scene images, handwritten text and synthetic documents. Individual characters have been manually segmented and represented by rectangular bounding boxes or polygonal segments. This dataset is not a standard one (loosely annotated) and does not contain pixel-accurate ground-truth for individual characters. Secondly, KAIST scene text dataset [28] comprises 3000 images captured both outdoor and indoor using a digital or mobile camera under different lighting conditions. This database is divided into 3 categories namely Korean, English and a composite one (contains Korean as well as English). The ground-truth for each image is stored in an XML file that contains information about the location of single characters or single words (using bounding boxes) and their transcription along with global information about the image. In addition to the XML file, a bitmap image is provided, where the segmentation of the text is at pixel-level.

Saund et. al [73] presented a user-interface (UI) design for labeling elements in images of printed documents at the pixel level. It is targeted toward selection of collections of foreground pixels in a document image such as machine print text, machine print graphics, handwritten text, handwritten graphics, stamps and noise. After a user has selected a set of pixels with the help of the mouse, those pixels are assigned a particular color to indicate the selected label. The label descriptions and their colors are set by a user-editable XML configuration file. The above strategy cannot be used for scene text annotation due to the degradations that affect the image and presence of multi-script text. We have developed a software tool, named as 'MAST' for annotating multi-script textual information in scene images. The main features of this tool are listed below:

- The tool can be used to annotate multi-script documents. The virtual keyboard interface proposed can easily be extended for tagging text in any new script, not already supported.

- The ground-truth information generated consists of (i) the pixel-accurate segmented image; (ii) individual segmented word images; (iii) script information and (iv) the Unicode text that can be used for evaluating the performance of document analysis and recognition techniques.

(a) (b)

```
5
DSC09861_flyer_1.png 150 360 82 100 Kannada ಶ್ರೀ
DSC09861_flyer_2.png 313 226 134 369 Kannada ದೇಗುಲ
DSC09861_flyer_3.png 609 296 70 204 English 2794
DSC09861_flyer_4.png 793 212 106 363 Kannada ಕಮಲಮ್ಮ
DSC09861_flyer_5.png 945 182 115 415 Kannada ಪುಟ್ಟೇಗೌಡ
```

(c)

Figure A.1: (a) An example multi-script image from MASTER database. (b) Pixel-accurate segmented image obtained using MAST toolkit. (c) The corresponding '.txt' file describing the attributes of each annotated word in the image.

- The software is open source; users can modify it to suit their needs.

The input image is processed by the tool on a word-by-word basis. The segmented words, pixel-level segmented image and a '.txt' file containing the number of words, file-names of the segmented words saved, coordinates of the corresponding bounding boxes, script labels and the Unicode text are stored in the source directory of the image. The structure of the '.txt' file for an example multi-script image is shown in Figure A.1.

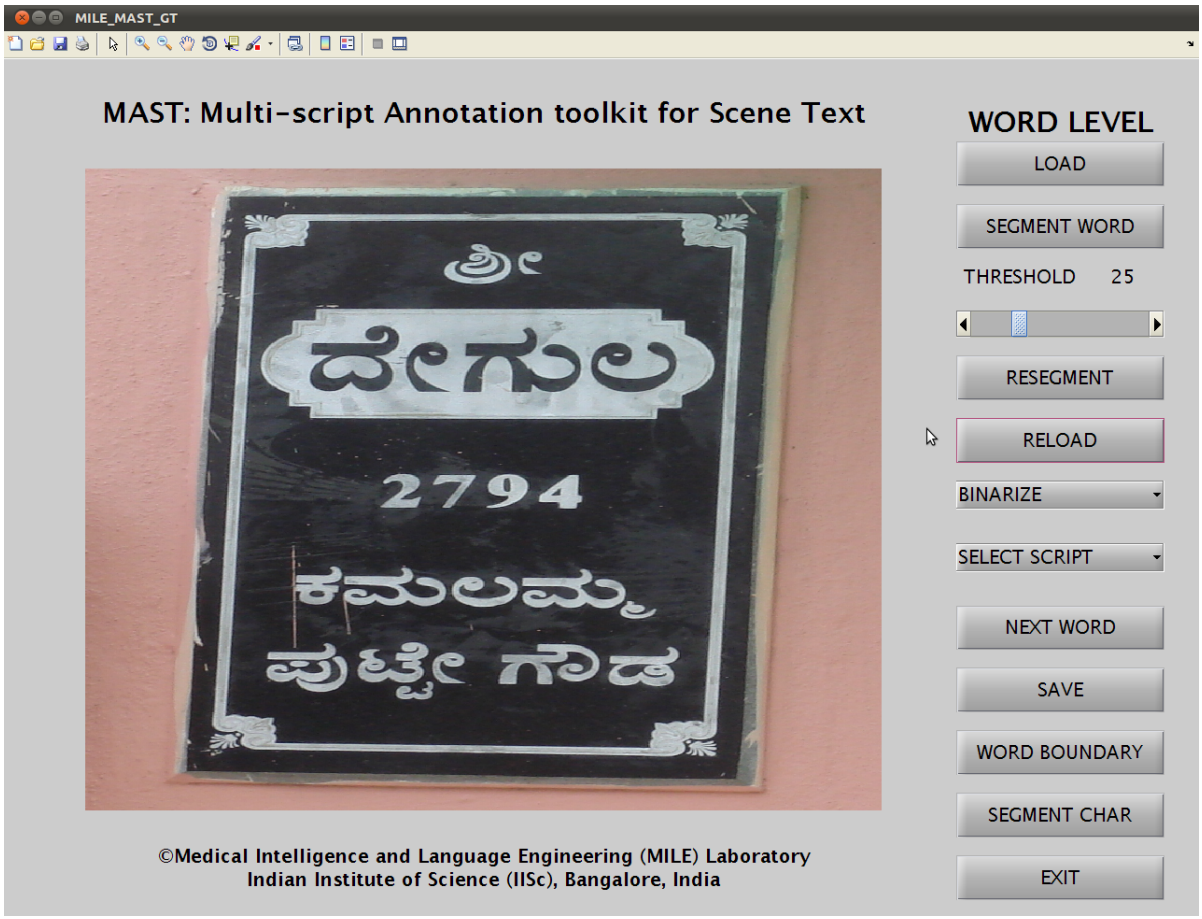


Figure A.2: Screenshot of the MAST user interface for word-level annotation.

A.2 Word-level annotation

A screen-shot of the main menu of the word-level annotation UI is shown in Figure A.2. After the image to be tagged is loaded, one can select a text region using the zoom option at the top of the UI tool bar. Actual segmentation is performed only on the zoomed-in region, accelerating the process. Once a word is zoomed in, the seed points are selected by the user within the character strokes and then the region-growing process is initiated. Whenever the input is through successive mouse clicks, the last selection should be made with the right click to complete the selection procedure. Region growing fails on low resolution characters even with repetitive manual seeding. Thus, to reduce the manual task and also to speed up segmentation time, known segmentation algorithms are used additionally.

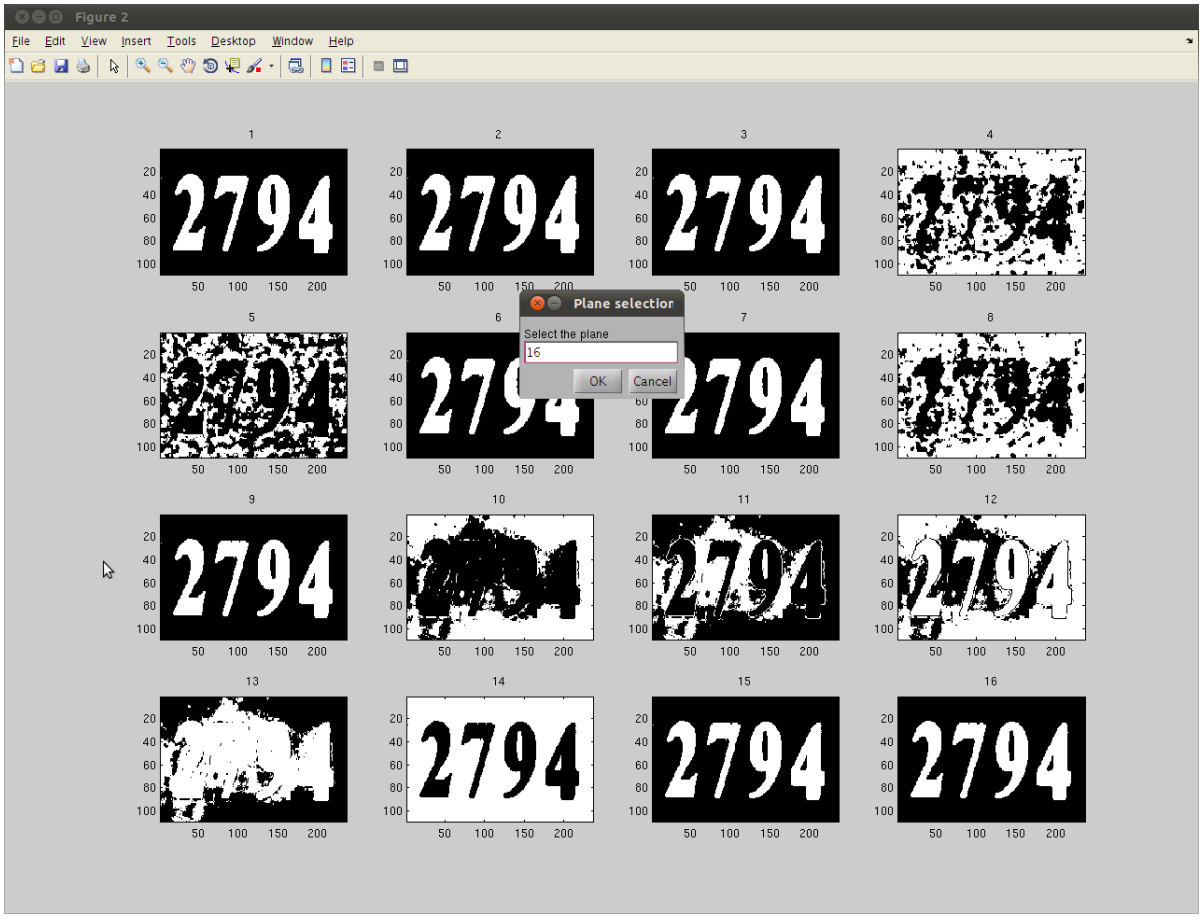


Figure A.3: Automated segmentation by MAST. Sixteen distinct segmentation results generated by MAST for a chosen scene word image shown in Figure A.1. The user selects the best result using a keyboard input (shown in the middle).

Sixteen different segmentation outputs are generated using multiple approaches. First, the original RGB image is converted to HSV and CIE L,a,b formats. Then, each of them are split into the three individual planes and Otsu's threshold [68] is applied individually on the resulting nine planes, which are essentially gray level images. In addition, three clusters are formed using the RGB information directly and six permutations of the formed clusters (each of the 3 clusters and the union of the other two clusters at a time) are obtained. The sixteenth output is obtained by binarizing the intensity image of the zoomed-in word using the robust automatic threshold of Kittler et. al [36].

All the segmented results are displayed in another window and a manual keyboard input is provided for the user to select the best of the results. Figure A.3 shows the

sixteen different segmentation results displayed to the user for a sample word image from MASTER dataset. The screen-shot also shows a text box with provision for the user entry through the keyboard to choose the best segmented output for either fully automated annotation or for subsequent manual correction. As seen in Figure A.3, the optimal choice is '16', since it has minimal requirement for manual editing. Once the user makes the choice, a mask is generated and overlaid on the original image. This semi-automated technique has improved the speed of segmentation and reduced the fatigue of the annotators. If the mask generated has distinct or well separated characters, then the user can select the script and annotate the word. If none of the segmentation results are satisfactory, the user can choose '0' and thus no mask will be generated. Canny edges [10] are also shown in a gray shade along with the segmented image for visual comparison of the segmented character boundaries and the edges so as to aid the user in deciding the quality of the segmented output.

Presently, the tool has provision for tagging the segmented word image in one of 10 Indic scripts, namely, Bangla, Devanagari, Gujarati, Gurmukhi, Kannada, Malayalam, Manipuri, Odiya, Tamil and Telugu besides Roman. The standard keyboard is used for tagging English text. The word segmentation process and tagging are repeated till all the words are considered.

A.3 Use of polygons to refine segmentation

A word image may not get segmented properly due to illumination changes, occlusion or low resolution of characters. For such cases, manual editing of the segmentation has been enabled by providing polygonal masks. These masks can be used to add parts of characters which have been merged to the background or delete parts of the background that have been added to a character. 'ADD PATCH' button provides the option for adding pixels in the polygonal format to the annotated mask. 'DELETE PATCH' button facilitates deletion of the background segmented as part of a character or splitting of merged characters. When add or delete option is selected, we can place a single polygon at a given time. Mask will be modified based on the operation performed and the shape

Figure A.4: Use of polygons to refine the best automated segmentation result. (a) Two sample word images from SVT dataset. (b) The segmentation results chosen by an user. (c) Segmented images after refinement by deletion and/or inclusion of appropriate regions defined using polygons.

and position of the polygon. The annotation tool then asks whether the same operation needs to be continued. If the user chooses 'yes', then the user can place another polygon to modify the annotated characters. If the choice is 'no', then the tool exits this edit loop. Figure A.4 illustrates the application of polygonal masks to refine the best segmentation result chosen by the user. The automated segmentation outputs and the outputs after manual editing using polygonal masks are shown for two sample images. In the case of the top word image in Figure A.4, a part of the background in the top right corner is marked by the tool as foreground. This patch of wrong binarization is selected using a polygonal mask and removed in a single step. In the bottom word image, multiple polygonal masks are used to remove and add patches to arrive at the proper segmentation result.

Figure A.5 shows some examples of pixel-level ground-truth obtained using MAST for typical scene images containing multiple scripts and various types of text layouts. For such images containing arbitrarily-oriented text, it is clear that bounding box-based ground-truth is not sufficient.

A.4 Creating keyboard interface for new scripts

In addition to segmenting and annotating the text present in generic scene images, MAST can also be used to generate a virtual keyboard interface for a new script. To do this, an image of the keyboard in the desired script is fed to the word-level annotation module as the input. The keyboard image may be obtained by creating a table in a html file, where

Figure A.5: Sample ground-truth images generated using MAST from scene images with multi-script content and arbitrary text orientations.

each cell of the table contains a specific character Unicode [81]. When this html file is viewed using a web browser, a table with the chosen script fonts will be displayed. A screen-shot of this table is used as the virtual keyboard image. Each key in the keyboard is segmented using the word-level segmentation module and mapped to its corresponding Unicode value using the standard English keyboard interface. These keymaps are stored in the resulting '.txt' file. The keyboard image and the corresponding keymaps are placed as required in the toolkit, which can be used to create ground-truth text in the chosen script.

For example, in Figure A.6, all the 83 keys of Kannada keyboard are segmented individually and mapped to their corresponding Unicodes. A part of the tagged file containing the Kannada Unicode values is also shown in the figure. The virtual keyboard image and the associated '.txt' file are then used to create an interface for ground-truthing texts in Kannada.

```
83
Kannada_unicode_1.png 8 7 61 55 English c85
Kannada_unicode_2.png 8 68 61 50 English c86
Kannada_unicode_3.png 8 124 61 43 English c87
Kannada_unicode_4.png 8 173 61 58 English c88
```

```
Kannada_unicode_17.png 75 7 61 55 English ccd
Kannada_unicode_18.png 75 68 61 50 English cbe
Kannada_unicode_19.png 75 124 61 43 English cbf
Kannada_unicode_20.png 75 173 61 58 English cc0
Kannada_unicode_21.png 75 236 61 49 English cc1
Kannada_unicode_22.png 75 290 61 61 English cc2
```

Figure A.6: Illustration of Kannada virtual keyboard interface. The keyboard image for tagging each key with the corresponding Unicode is overlayed on the .txt file generated using MAST, which contains the Kannada Unicode mapped for each key.

A.5 Conclusion

A versatile software tool has been developed for annotating the text present in multi-script scene images. It can be used to create ground-truth data for any generic scene image at the word level or character/symbol level depending on the user's requirement. The ground-truth text regions are represented at the pixel-level. Bounding boxes of adjacent CCs can have a high degree of overlap for skewed or curved text and hence, they do not represent accurate locations of the characters. The pixel-level ground-truth gives an accurate representation of the text regions for arbitrary text orientations.

Currently, MAST has provision for tagging in 10 Indic scripts and Roman. The Devanagari virtual keyboard interface can be used to tag Marathi, Hindi, Konkani and Sanskrit languages. Likewise, the Bangla virtual keyboard interface can be used to tag Bangla, Assamese and Manipuri texts. One useful contribution is the design of virtual

keyboard interface. It is easy to create interfaces for new scripts, thereby making the toolkit applicable to any script. The software is open source and is available online [54] along with a detailed description of the functionalities of each of the menu items. We hope that researchers worldwide find it useful in creating ground-truth for any generic document image.

There is a requirement of hundreds to thousands of images for organizing a competition on multi-script text segmentation and recognition. The members of MILE laboratory collected close to 4000 images in Bengaluru and their native places to cover all possible kinds of images that exist in Indian scenario. MAST was originally designed to generate ground-truth for the images collected by the members of MILE laboratory. The multi-script robust reading competition (MRRC) was organized as part of ICDAR 2013. The database released to the competition participants is termed as Multi-script And Scene TExt Reading (MASTER) database. The evaluation procedure, submitted methods, baseline methods and the tasks and images used in the competition are discussed in Appendix B.

Appendix B

Multi-script robust reading competition in ICDAR 2013

Summary

A competition was organized by Deepak Kumar et. al [42] as part of ICDAR 2013 to detect text from multi-script scene images. The motivation was to look for script-independent methods that detect the text and extract it from the scene images, which may be applied directly to an unknown script. The competition had four distinct tasks: (i) text localization and (ii) segmentation from scene images containing one or more of Kannada, Tamil, Hindi, Chinese and English words. (iii) English and (iv) Kannada word recognition task from scene word images. There were totally four submissions for the text localization and segmentation tasks. For the other two tasks, we have evaluated our own methods, namely NESP and MAPS to serve as reference results. Each algorithm is discussed and suggestions are provided to improve its performance. Graphical depiction of f-score of individual images in the form of benchmark values is proposed to show the strength of an algorithm.

B.1 Introduction

A decade has passed since the first robust reading competition (RRC) on camera-captured scene images with Roman text was organized by Lucas et. al in ICDAR 2003 [52]. Subsequent competitions as part of ICDAR 2005 and 2011 by Lucas et. al [53] and Shahab et. al [75], respectively. Born-digital images were introduced in RRC by Karatzas et. al [29] in ICDAR 2011, again containing Roman text only. However, no competition has been held on multi-script, camera-captured scene images; there are a few research contributions though [34, 48]. Applications which transcribe or translate words of unknown scripts in a scene are of great value to a foreigner visitor.

In the Indian scenario, we find hoardings and street name boards in multiple languages (scripts). A multi-script robust reading competition (MRRC) was organized [42], as part of ICDAR 2013 [24], to motivate the development of novel applications for identification and recognition of Indic scripts in camera-captured scene images. The images contain text in one of Roman, Kannada, Tamil, Devanagari and Chinese scripts. Figure B.1 shows some sample images from the training set of text localization and segmentation tasks. This MRRC gave a platform for researchers around the globe to address this issue, hitherto very less explored. The competition ran in open mode, where each participant downloaded the test set and uploaded the results of their algorithms.[1]. Thirty people registered to participate in MRRC and three of them submitted their results.

B.2 Datasets collected for MRRC

We collected nearly 4000 camera-captured images mainly from Bengaluru city roads, Karnataka, India. Four different tasks were organized in this competition, namely

1. Text localization: Obtain a bounding box around the text, irrespective of the script.

2. Text segmentation: Identify the text pixels, irrespective of the script.

[1] `http://mile.ee.iisc.ernet.in/mrrc`

Figure B.1: Sample multi-script images provided for training in the text localization and segmentation tasks of the competition.

3. Word recognition: Recognize the words from the given set of manually segmented word images, containing:

   (a) English words + Indo-Arabic numerals

   (b) Kannada words

In this competition, 167 camera-captured scene images each were provided for training and testing (1:1) for text localization and segmentation tasks. 495 and 645 word

(a) English word images

(a) Kannada word images

Figure B.2: Sample word images used for English and Kannada word recognition tasks in MRRC.

images were provided as the training and test set, respectively, for English word recognition task. Kannada word recognition task had 300 training and 243 test images. Figure B.2 shows some samples from the English and Kannada training sets. All the images had a background of two pixels all around the located boundary to provide proper background. Ground-truth for the data set was created using our multi-script annotation toolkit (MAST) [31, 54], a user interface (available for free download) to annotate scene images at the pixel level.

We name our data set as 'Multi-script and scene text reading' (MASTER) data set. An exhaustive variety of degradations and challenges are covered in this dataset, namely artistic fonts, curved, embossed, engraved, handwritten, multi-colored, multi-font, multi-script and slant text, non-uniform illumination, images with motion blur, occluded text, low-resolution text, text on glossy surfaces, text with shading and shear. An image captured during night with normal mode has been included in the dataset as a night-vision sample, for the first time.

B.3 Performance evaluation

In ICDAR 2003 competition [52], Lucas et. al proposed a procedure for evaluating algorithms. The same was used in ICDAR 2005 competition [53]. Since [52] heavily penalizes algorithms for detecting text lines, rather than words, Wolf and Jolion [86] proposed another method to evaluate the algorithms on text localization task, which we use. The area recall and precision thresholds are $t_r = 0.8$ and $t_p = 0.4$, respectively. For one-to-many matches, $f_{cs}(k) = 0.8$ is used for $Match_G$ calculation and for many-to-one matches, $f_{cs}(k) = 1$ is used for $Match_D$ calculation [86]. This value of $f_{cs}(k)$ indicates punishing over-segmented words and no punishment for under-segmented words, which in turn reduces penalization of the algorithms locating a text line, rather than a word.

Text segmentation performance is usually evaluated [12] using connected components (CCs). Ground-truth CCs of a test image are matched against the output of the algorithm to determine whether the components are well-segmented, merged, broken or lost. This evaluation does not account for non-text components output by an algorithm if well-segmented components are used in ranking [29]. Hence, we employ pixel-level information to evaluate the algorithms. Precision, recall and f-score are calculated for the participating algorithms on the text segmentation task.

Word recognition results are evaluated based on the number of correctly recognized words and Levenshtein distance computed from Unicode strings.

Precision, recall and f-score values of an algorithm on an entire dataset do not reveal any information about the algorithm performance on individual images. Hence, a novel

method is proposed to evaluate text localization and segmentation algorithms graphically.

Completion calls for presenting results for all the tasks of a competition. Hence, we have also obtained results on the test images using our own methods to be used as the baseline for the other tasks conducted in the competition.

B.4 Entries received for the competition

We had participation from three different countries: Spain, China and India. A brief description of all the methods submitted is provided in the following sub-sections.

B.4.1 Method1 by Yin et.al

Entries were submitted by Xuwang Yin[2], Xu-Cheng Yin[2] and Hong-Wei Hao[3] for the first two tasks.

Text localization algorithm: The character candidates are extracted by exploring the hierarchical structure of maximally stable extremal regions (MSERs) and adopting simple features. They are clustered into text candidates (TC) by a single-link algorithm, where distance weights and threshold for clustering are learned automatically by a novel, self-training, distance metric learning method. The posterior probabilities of TC corresponding to non-text are estimated with a character classifier using Bayes' rule. TC with high non-text probabilities are eliminated and others are identified using a text classifier.

Text segmentation algorithm: Text candidates are in fact MSERs. Text is segmented by setting pixels presented in text as white (text pixels) and others as black.

[2]Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology, Beijing.

[3]Institute of Automation, Chinese Academy of Sciences.

B.4.2 Method2 by Gómez and Karatzas

An algorithm for text segmentation was submitted by Lluís Gómez[4] and Dimosthenis Karatzas[4]. In the preprocessing stage, MSER algorithm is used to obtain a region decomposition of the input image. Then, two different clustering techniques are combined in a single parameter-free procedure to detect groups of regions organized as text. The maximally meaningful groups are first detected in several feature spaces, where each feature space is a combination of proximity information (x,y coordinates) and a similarity measure (in terms of intensity, color, size, gradient magnitude, etc.), thus providing a set of hypotheses of text groups. Evidence accumulation framework is used to combine all these hypotheses to get the final estimate. The resulting method is independent of the script, can deal with any kind of font types and sizes, and is not constrained to horizontally aligned text.

B.4.3 Method3 by Sethi and Bawa

Ganesh K. Sethi[5] and Rajesh K. Bawa[6] submitted a method for segmentation. In this method, text is segmented from natural scene images based on edge analysis and morphological operators. The images are converted to gray scale and Canny edges are detected. The edge image is morphologically dilated and analyzed to remove edges corresponding to non-text regions. Then, the image is binarized using the mean and standard deviation values of edge pixels. The resulting image is post-processed to fill the gaps and smoothen the text strokes.

B.5 Methods used as baseline for comparison

Our own methods, have been applied on the data to obtain reference level (baseline) performance for comparison:

[4]Computer Vision Center, Universitat Autònoma de Barcelona, Spain.
[5]M. M. Modi College, Patiala, Punjab.
[6]Punjabi University, Patiala, Punjab.

OTCYMIST method is used as the reference for text segmentation. PLT/NESP/MAPS methods are used for word recognition.

Benchmark recognition rate: The pixel-level segmented test word images are extracted using the BB information from the annotated (using MAST) text localization results. These clean images are used to obtain the upper bound of benchmark [39] word recognition rate on the MASTER data set.

The word images segmented by each of the above methods [39, 41, 43, 44] are padded with zeros around the boundary, with the minimum of the image height or width value. They are then fed to Omnipage [67] or Tesseract OCR [76], for English or Kannada word recognition, respectively.

B.6 Results and Discussion

The results of the participants' algorithms, as well as the reference methods, are discussed below, for each task:

B.6.1 Text localization task

A novel method is introduced to analyze text locating and segmenting algorithms. Let GT_i be the fraction of ground-truth text pixels in the i^{th} image. All the images in the dataset are sorted based on their GT_i values. Two metrics, named as benchmark (B_i) and algorithm result (AR_i) are computed for the individual images using inverse grading:

$$B_i = 1/GT_i \tag{B.1}$$

$$AR_{ji} = F_{ji}/GT_i \tag{B.2}$$

where, F_{ji} and AR_{ji} are the f-score and the result of the j^{th} algorithm on the i^{th} image, respectively. A high value of B_i indicates a high amount of non-text information in the image. An algorithm needs to eliminate all the non-text information to gain an AR value

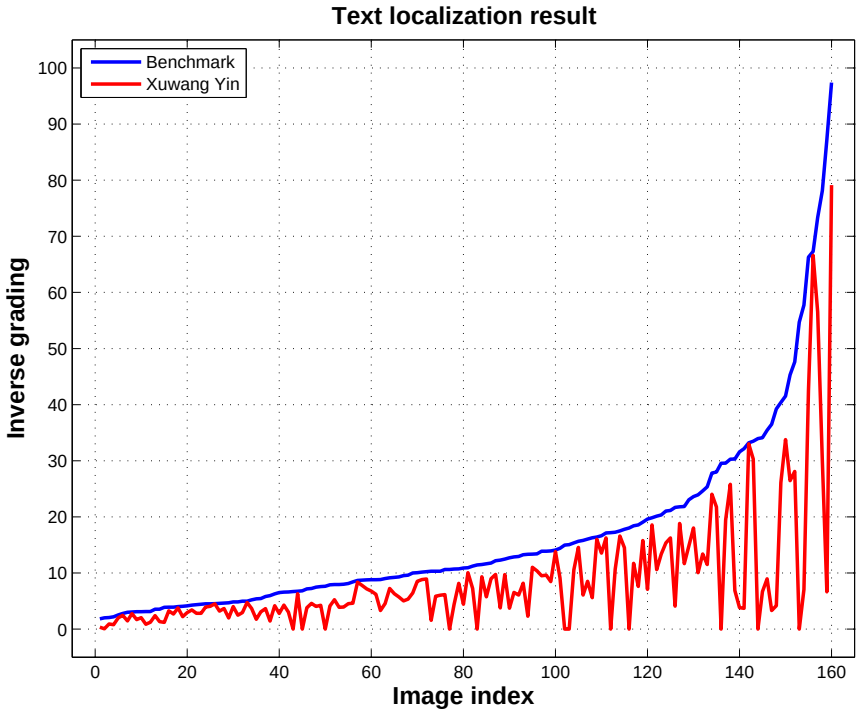


Figure B.3: A plot of benchmark values (B_i in blue colour) and algorithm result (AR_i in red colour) on individual images in the MASTER data set for the text localization task. AR_i follows the B_i values in the case of normal images and fluctuates between high and low values in the case of complex images.

equal to the benchmark.

The results of the single entry for this task from Yin et. al, evaluated using Wolf and Jolion method [86], are listed in Table B.1. Yin et. al group the horizontal character candidates. Since a set of images contain curved text, and Indic scripts need a unique way of grouping, this algorithm entails a low recall value for the text localization task.

Figure B.3 shows the plot of benchmark values and algorithm result on each image in the dataset. Images with B_i below 20 are considered 'normal'; the others, 'complex'. AR_i values are close to B_i values for images with a large fraction of text pixels. As the fraction of text pixels reduces, the performance of the algorithm is erratic. The algorithm strength (AS) is estimated as,

$$AS_j = \sum_i AR_{ji} / \sum_i B_i \tag{B.3}$$

Table B.1: Performance of text locating algorithm (evaluated using Wolf and Jolion method) on the MASTER dataset. AS: algorithm strength (for normal and complex images).

Participant	Precision	Recall	F-score	AS (Normal)	AS (Complex)
Yin, Yin and Hao	0.64	0.42	0.51	0.58	0.56

Table B.2: Performance evaluation of the algorithms submitted for the text segmentation task. Precision, recall and f-score values are calculated using the ground-truth. Algorithm strength (AS) values are shown separately for normal and complex images.

Method/ Participant Name	Precision	Recall	F-score	AS (Normal)	AS (Complex)
Yin, Yin and Hao	0.71	0.67	0.69	0.80	0.56
Gómez and Karatzas	0.64	0.58	0.61	0.69	0.35
Sethi and Bawa	0.33	0.72	0.45	0.65	0.18
OTCYMIST	0.50	0.29	0.37	0.46	0.21

where, i is the index of the normal or complex images. The AS values for normal and complex cases are tabulated separately in Table B.1.

B.6.2 Text segmentation task

We received three submissions for this task. Using OTCYMIST method as the baseline, the submissions are evaluated and the results are given in Table B.2.

Figure B.4 shows the benchmark values and the results of the best algorithm for the individual images in the data set. For normal images, the algorithm has good results. However, as shown by Table B.2, in the case of complex images, sometimes the result is either poor or bad, due to the degradations or the algorithm's post-processing threshold.

A moving average approach is used to includes the result of all the algorithms in a single plot. A window of 11 images is moved through the images ordered by their B_i values. Mean values of B_i and AR_i are computed. Figure B.5 shows the plot of these average values for the text segmentation task. The top two methods use MSER algorithm during the character segmentation stage, and differ only at the post-processing stage. So, their segmentation results are poor on images having non-uniform illumination on the

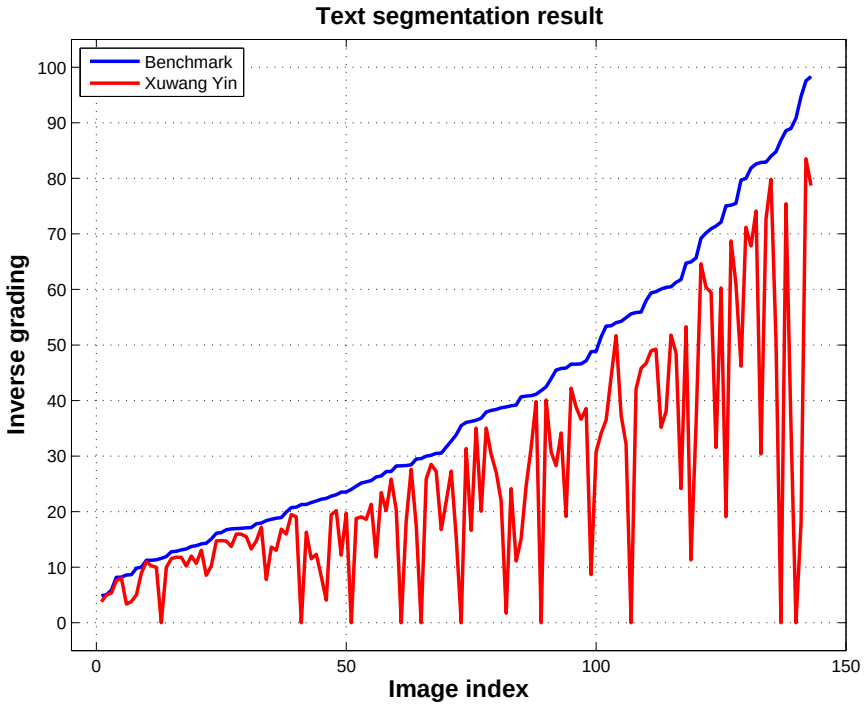


Figure B.4: A plot of benchmark values (B_i in blue colour) and algorithm result (AR_i in red colour) of Yin's method on individual images in the MASTER dataset, for the text segmentation task. AR_i follows the B_i values in the case of normal images, but fluctuates in the case of complex images.

text, text on glass or occluded text. The plots are close, revealing that the underlying methodology of the algorithms are similar. The results, where these two algorithms differ, are analyzed to figure out the reason. The top performing algorithm is more effective in removing non-text components, thereby increasing its precision. Neighboring components are removed while grouping and textured non-text components are not removed in Yin's method. If stroke width information of the characters is used, then all the non-text components can be filtered out properly in the segmented image.

B.6.3 English word recognition task

At least three characters exist in each English word image. We did not receive any submission for this task. Our own methods are evaluated on the dataset for the purpose of benchmarking. Omnipage OCR [67] is used for recognition. The edit distance (ED) between the ground-truth and the output of a method is calculated, giving equal weights

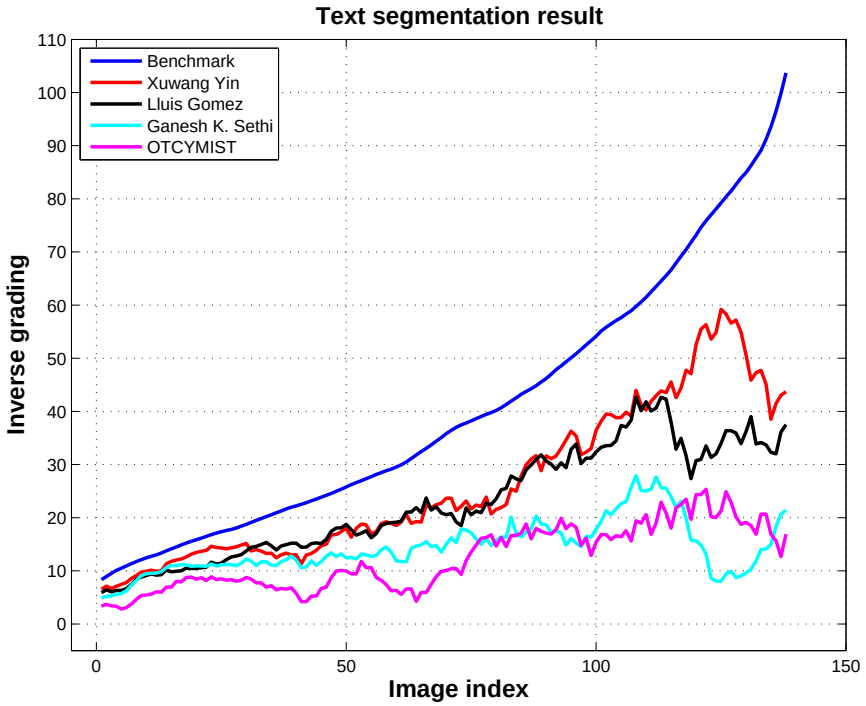


Figure B.5: A plot of average values of B_i and AR_i for the algorithms submitted. Top performing algorithm is the nearest to follow the average values calculated from the benchmark.

to additions, deletions and insertions. The ED of each word is normalized by the number of letters in the word and all the normalized EDs are accumulated to get the total edit distance (TED-E). The English word recognition (EWR) rate and the TED-E are tabulated in Table B.3 for the different methods. Compared to ICDAR 2003 or 2011 word image datasets, an additional complexity of curved text is included in this MASTER dataset. One-third of the word images contain shear, slant or curved text. Thus, 33% of the words in the dataset cannot be directly recognized by standard OCR engines, since they handle only horizontally oriented text. The number of words commonly recognized by all the three (PLT, NESP and MAPS) methods is 38.8% and the union of all the words recognized by these methods is 53.5%. These numbers indicate that a few words recognized by a method are not recognized by others. The union result does not even cross the benchmarked recognition rate.

Table B.3: Comparison of word recognition rate and total edit distance measures for English (EWR, TED-E) and Kannada (KWR, TED-K) for different methods, namely, Benchmark, PLT, MAPS, NESP and raw image.

Method	EWR	TED-E	KWR	TED-K
Benchmark [39]	57.7	215.1	11.1	178.7
PLT [44]	46.9	299.5	5.3	210.6
MAPS [41]	46.9	305.9	4.9	209.1
NESP [43]	45.1	305.4	5.8	212.3
PLT ∪ MAPS ∪ NESP	53.5	—	7	—
PLT ∩ MAPS ∩ NESP	38.8	—	4.1	—
Baseline (raw image)	37.5	369.4	2.5	218.5

B.6.4 Kannada word recognition task

At least two Unicodes exist in each Kannada word image. We did not receive any submission for this task, and the dataset is benchmarked with our methods. Tesseract OCR engine [76] is used for recognizing Kannada word images. The ED is calculated at Unicode string level before normalization. The Kannada word recognition (KWR) rate and the total edit distance for Kannada words (TED-K) are given in Table B.3 for the different methods. The words commonly recognized and the union of words recognized by all the three methods are 4.1% and 7%, respectively.

The recognition rate (RR) for manually segmented word images is low (both in English and Kannada) due to the additional complexity of curved or slanted text. If those are aligned before feeding them to OCR, then the benchmark RR may reach 90% for English words. The union of the recognized words is below the benchmark RR, indicating the need to improve word segmentation. In the case of Kannada words, part of the consonants appear in the descender region of the word. Care should be taken while segmenting these components for recognition, generating the Unicode and also while aligning the curved words.

B.7 Conclusion

A robust reading competition is organized on multi-script scene images. Each character in a script is confined within broadly defined three sections, namely ascender, x-height and descender. A set of script-related rules needs to be defined to group characters and locate a word. Hence, the text localization is a more complex task than text segmentation. Yin et. al perform horizontal grouping, which works for Roman and Chinese, but not for Kannada and Devanagari.

A novel method is proposed to sort the images by inverse grading. The number of normal and complex images can be used as a measure to benchmark a dataset. The several object detection datasets available can be ranked based on the sum of benchmark values. Only the count of text or BB pixels is used to benchmark an image in this competition, which in turn benchmarks a dataset. Additionally taking into account the pixels affected by degradation can further improve the benchmarking measure on the datasets.

Publications based on this Thesis

Conferences:

1. Thotreingam Kasar, Deepak Kumar, M. N. Anil Prasad, D. Girish and A. G. Ramakrishnan. MAST: Multi-Script Annotation Toolkit for Scenic Text. In *Proc. Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data (J-MOCR-AND)*, Sept. 17, 2011, Beijing, China. doi: 10.1145/2034617.2034633

2. Deepak Kumar and A. G. Ramakrishnan. OTCYMIST: Otsu–Canny Minimal Spanning Tree for Born-Digital Images. In *Proc. 10th IAPR Intl. Workshop on Document Analysis Systems (DAS 2012)*, Queensland, Australia, March 27-29, 2012. doi: 10.1109/DAS.2012.65

3. Deepak Kumar and A. G. Ramakrishnan. Power-law Transformation for Enhanced Recognition of Born-Digital Word Images. In *Proc. Intl. Conf. on Signal Processing and Communications (SPCOM)*, July 22-25, 2012, Bangalore. doi: 10.1109/SPCOM.2012.6290009

4. Deepak Kumar, M. N. Anil Prasad and A. G. Ramakrishnan. MAPS: Midline analysis and propagation of segmentation. In *Proc. 8th Indian Conf. on Vision, Graphics and Image Processing (ICVGIP)*, December 16-19, 2012. doi: 10.1145/2425333.2425348

5. Deepak Kumar and A. G. Ramakrishnan. Recognition of Kannada characters extracted from scene images. In *Proc. Workshop on Document Analysis and Recognition (DAR 2012)*, December 16, 2012. doi: 10.1145/2432553.2432557

6. Deepak Kumar, M. N. Anil Prasad and A. G. Ramakrishnan. Benchmarking recognition results on camera captured word image data sets. In *Proc. Workshop on DAR 2012*, December 16, 2012. doi: 10.1145/2432553.2432572

7. Deepak Kumar, M. N. Anil Prasad and A. G. Ramakrishnan. NESP: Nonlinear enhancement and selection of plane for optimal segmentation and recognition of scene word images. In *Proc. Document Recognition and Retrieval (DRR) XX*, San Francisco, CA, USA, February 5-7, 2013. doi: 10.1117/12.2008519

8. Deepak Kumar, M. N. Anil Prasad and A. G. Ramakrishnan. Multi-script robust reading competition in ICDAR 2013. In *Proc. MOCR 2013*, Washington DC, USA, August 24, 2013. doi: 10.1145/2505377.2505390

Journals:

1. Deepak Kumar, M. N. Anil Prasad and A. G. Ramakrishnan. Elegant techniques for robust segmentation of scene and born-digital word images. *manuscript under preparation to be communicated to IEEE Transactions on Image Processing.*

2. Deepak Kumar and A. G. Ramakrishnan. Kannada word recognition from scene images. *manuscript under preparation to be communicated to ACM Transactions on Asian Language Information Processing.*

References

[1] Abbyy Fine Reader. `http://www.abbyy.com/`.

[2] Adobe Reader. `http://www.adobe.com/products/acrobatpro/scanning-ocr-to-pdf.html`.

[3] H. J. Alitto and W. M. Usrey. Corticothalamic feedback and sensory processing. *Current Opinion in Neurobiology*, 13:440–445, 2003.

[4] J. H. AlKhateeb, J. Ren, J. Jiang, and S. S. Ipson. Word-based handwritten Arabic script recognition using DCT features and neural network classifier. In *Proc. IEEE International Multi-Conference on SSD*, 2008.

[5] A. Antonacopoulos, B. Gatos, and D. Karatzas. ICDAR 2003 Page Segmentation Competition. In *Proc. 7th ICDAR*, pages 688–692, 2003.

[6] K. G. Aparna and A. G. Ramakrishnan. A complete Tamil Optical Character Recognition System. In *Proc. 5th DAS*, pages 53–57, 2002.

[7] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. PAMI*, 24:509–522, 2002.

[8] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *Proc. CVPR*, pages 26–33, 2005.

[9] Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Trans. PAMI*, 26(9):1124–1137, September 2004.

[10] J. Canny. A Computational Approach to Edge Detection. *IEEE Trans. PAMI*, 8(6):679–698, November 1986.

[11] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, and B. Girod. Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In *Proc. ICIP*, pages 2609–2612, 2011.

[12] A. Clavelli, D. Karatzas, and J. Llados. A Framework for the Assessment of Text extraction Algorithms on Complex Colour Images. In *Proc. 9th DAS*, pages 19–28, 2010.

[13] T. E. de Campos, B. R. Babu, and M. Varma. Character recognition in natural images. In *Proc. VISAPP*, February 2009.

[14] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, 2 edition, 2006.

[15] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *Proc. 23rd CVPR*, pages 2963–2970, 2010.

[16] B. Gatos, K. Ntirogiannis, and I. Pratikakis. ICDAR 2009 Document Image Binarization Contest (DIBCO 2009). In *Proc. 10th ICDAR*, pages 1375–1382, 2009.

[17] GNU Image Manipulation Program (GIMP). `http://www.gimp.org/`.

[18] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, 2 edition, 2002.

[19] Jr. H. C. Thode. *Testing for Normality*. Marcel Dekker, New York, 2002.

[20] T. K. Ho, J. J. Hull, and S. N. Srihari. A word shape analysis approach to lexicon based word recognition. *Pattern Recognition Letters*, 13(11):821–826, 1992.

[21] IAPR TC11 Reading Systems-Datasets List. `http://www.iapr-tc11/mediawiki/index.php/Datasets`.

[22] ICDAR 2005 Competitions.
http://http://algoval.essex.ac.uk:8080/icdar2005/index.jsp.

[23] ICDAR 2011 Robust Reading Competition - Challenge 1: Reading Text in Born-Digital Images (Web and Email).
http://www.cv.uab.es/icdar2011competition/.

[24] ICDAR 2013. www.icdar2013.org.

[25] Inzisoft. http://www.inzisoft.com/english/.

[26] S. Jeannin. MPEG-7 Visual part of experimentation Model Version 9.0. (ISO/IEC JTC1/SC29/WG11/N3914), January 2001. 55th MPEG meeting, Pisa, Italia.

[27] Joint Photographic Experts Group (JPEG). http://www.jpeg.org/.

[28] KAIST AIPR Scene Text Database. http://ai.kaist.ac.kr/home/.

[29] D. Karatzas, S. R. Mestre, J. Mas, F. Nourbakhsh, and P. P. Roy. ICDAR 2011 Robust Reading Competition - Challenge 1: Reading Text in Born-Digital Images (Web and Email). In *Proc. 11th ICDAR*, pages 1485–1490, 2011.

[30] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i. Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. d. l. Heras. ICDAR 2013 Robust Reading Competition. In *Proc. 12th ICDAR*, pages 1115–1124, 2013.
http://dag.cvc.uab.es/icdar2013competition/.

[31] T. Kasar, D. Kumar, M. N. Anil Prasad, D. Girish, and A. G. Ramakrishnan. MAST: Multi-script Annotation Toolkit for Scenic Text. In *Proc. Joint workshop on Multilingual OCR and Analytics for Noisy and Unstructured Text Data*, pages 1–8, September 2011. http://mile.ee.iisc.ernet.in/mast/.

[32] T. Kasar, J. Kumar, and A. G. Ramakrishnan. Font and background color independent text binarization. In *Proc. 2nd CBDAR*, pages 3–9, 2007.

[33] T. Kasar and A. G. Ramakrishnan. COCOCLUST: Contour-based color clustering for robust binarization of colored text. In *Proc. 3rd CBDAR*, pages 11–17, 2009.

[34] T. Kasar and A. G. Ramakrishnan. Multi-script and multi-oriented text localization from scene images. In *Springer LNCS 7139*, pages 1–14, 2012.

[35] W. Y. Kim and Y. S. Kim. A new region-based shape descriptor. (TR 15-01), December 1999. Pisa.

[36] J. Kittler, J. Illingworth, and J. Foglein. Threshold selection based on a simple image statistic. *Computer Vision, Graphics, and Image Processing*, 30(2):125–147, 1985.

[37] B. Vijay Kumar and A. G. Ramakrishnan. Machine Recognition of Printed Kannada Text. In *Proc. 5th DAS*, pages 37–48, 2002.

[38] B. Vijay Kumar and A. G. Ramakrishnan. Radial basis function and subspace approach for printed Kannada text recognition. In *Proc. ICASSP-04*, pages 321–324, 2004.

[39] D. Kumar, M. N. Anil Prasad, and A. G. Ramakrishnan. Benchmarking recognition results on camera captured word image data sets. In *Proc. Workshop on Document Analysis and Recognition (DAR 2012)*, 2012.

[40] D. Kumar, M. N. Anil Prasad, and A. G. Ramakrishnan. Benchmarking recognition results on word image datasets. *CoRR*, abs/1208.6137, 2012. `http://arxiv.org/abs/1208.6137`.

[41] D. Kumar, M. N. Anil Prasad, and A. G. Ramakrishnan. MAPS: Midline analysis and propagation of segmentation. In *Proc. 8th ICVGIP*, 2012.

[42] D. Kumar, M. N. Anil Prasad, and A. G. Ramakrishnan. Multi-script robust reading competition in ICDAR 2013. In *Proc. MOCR*, 2013.

[43] D. Kumar, M. N. Anil Prasad, and A. G. Ramakrishnan. NESP: Nonlinear enhancement and selection of plane for optimal segmentation and recognition of scene word images. In *Proc. 20th DRR*, 2013.

[44] D. Kumar and A. G. Ramakrishnan. Power-law transformation for enhanced recognition of born-digital word images. In *Proc. 9th SPCOM*, 2012.

[45] D. Kumar and A. G. Ramakrishnan. Recognition of kannada characters extracted from scene images. In *Proc. Workshop on Document Analysis and Recognition (DAR 2012)*, 2012.

[46] D. Kumar and A. G. Ramkrishnan. OTCYMIST: Otsu–Canny Minimal Spanning Tree for Born-Digital Images. In *Proc. 10th DAS*, 2012.

[47] V. Lavrenko, T. Rath, and R. Manmatha. Holistic Word Recognition for Handwritten Historical Documents. In *Proc. Document Image Analysis for Libraries (DIAL)*, pages 278–287, 2004.

[48] S. Lee, M. S. Cho, K. Jung, and J. H. Kim. Scene Text Extraction with Edge Constraint and Text Collinearity Link. In *Proc. 20th ICPR*, August 2010.

[49] LevenshteinVI. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707–10, 1966.

[50] H. Liu and X. Ding. Handwritten Character Recognition Using Gradient Feature and Quadratic Classifier with Multiple Discrimination Schemes. In *Proc. 8th ICDAR*, pages 19–25, 2005.

[51] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. 7th ICCV*, pages 1150–1157, 1999.

[52] S. M. Lucas. ICDAR 2003 Robust Reading Competitions: Entries, Results, and Future Directions. *IJDAR*, 7(2):105–122, June 2005.

[53] S. M. Lucas. Text Locating Competition Results. In *Proc. 8th ICDAR*, pages 80–85, 2005.

[54] MAST: Multi-script Annotation toolkit for Scene Text. Open source code available at `http://mile.ee.iisc.ernet.in/mast/`.

[55] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. BMVC*, pages 384–393, 2002.

[56] MATLAB. `http://www.mathworks.com/`.

[57] Microsoft paint. `http://www.microsoft.com/`.

[58] A. Mishra, K. Alahari, and C. V. Jawahar. An MRF Model for Binarization of Natural Scene Text. In *Proc. 11th ICDAR*, pages 11–16, 2011.

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

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

[61] Moving Picture Experts Group (MPEG). `http://mpeg.chiariglione.org/`.

[62] L. Neumann and J. Matas. A Method for Text Localization and Recognition in Real-World Images. In *Proc. 10th ACCV*, pages 770–783, 2010.

[63] W. Niblack. *An Introduction to Digital Image Processing.* Englewood Cliffs, N.J. Prentice Hall, 1986.

[64] F. Nourbakhsh, P. B. Pati, and A. G. Ramakrishnan. Document Page Layout Analysis using Harris-corner Points. In *Proc. 4th ICISIP*, pages 149–152, 2006.

[65] T. Novikova, O. Barinova, P. Kohli, and V. S. Lempitsky. Large-Lexicon Attribute-Consistent Text Recognition in Natural Images. In *Proc. ECCV*, pages 752–765, 2012.

[66] K. Ntirogiannis, B. Gatos, and I. Pratikakis. An Objective Evaluation Methodology for Document Image Binarization Techniques. In *Proc. 8th DAS*, pages 217–224, 2008.

[67] Nuance Omnipage Reader. `http://www.nuance.com/`.

[68] N. Otsu. A Thresholding Selection Method from Gray-level Histogram. *IEEE Trans. SMC*, 9:62–66, March 1979.

[69] P. B. Pati and A. G. Ramakrishnan. Word Level Multi-script Identification. *Pattern Recognition Letters*, 29:1218–1229, 2008.

[70] R. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 38:1389–1401, 1957.

[71] T. Rath, R. Manmatha, and V. Lavrenko. A Search Engine for Historical Manuscript Images. In *Proc. SIGIR'04*, pages 369–376, 2004.

[72] S. V. Rice, F. R. Jenkins, and T. A. Nartker. The Fifth Annual Test of OCR Accuracy. *Information Science Research Institute*, (TR-96-01), 1996.

[73] E. Saund, J. Lin, and P. Sarkar. PixLabeler: User Interface for Pixel-Level Labeling of Elements in Document Images. In *Proc. 10th ICDAR*, pages 646–650, 2009.

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

[75] A. Shahab, F. Shafait, and A. Dengel. ICDAR 2011 Robust Reading Competition - Challenge 2: Reading Text in Scene Images. In *Proc. 11th ICDAR*, pages 1491–1496, 2011.

[76] Tesseract OCR Engine. `http://code.google.com/p/tesseract-ocr/`.

[77] The Chars74K dataset. `http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/`.

[78] The Type-Reading Optophone. *Scentific American Monthly*, pages 109–110, October 1920.

[79] C. M. Thillou and B. Gosselin. Color text extraction from camera-captured images: the impact of the choice of the clustering distance. In *Proc. 8th ICDAR*, pages 312–316, 2005.

[80] C. M. Thillou and B. Gosselin. Color text extraction with selective metric-based clustering. *Computer, Vision and Image Understanding*, 107(2):97–107, 2007.

[81] Unicode 6.0 Character Code Charts. `http://unicode.org/charts/`.

[82] K. Wang, B. Babenko, and S. Belongie. End-to-End Scene Text Recognition. In *Proc. 13th ICCV*, pages 1457–1464, 2011.

[83] K. Wang and S. Belongie. Word spotting in the wild. In *Proc. 11th ECCV*, pages 591–604, 2010. `http://vision.ucsd.edu/~kai/svt/`.

[84] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks. In *Proc. ICPR*, pages 3304–3308, 2012.

[85] J. J. Weinmann, E. Learned-Miller, and A. R. Hanson. Scene text recognition using similarity and a lexicon with sparse belief propagation. *IEEE Trans. PAMI*, 31(10):1733–1746, 2009.

[86] C. Wolf and J. M. Jolin. Object count/area graphs for the evaluation of object detection and segmentation algorithms. *IJDAR*, 8(4):280–296, 2006.

[87] Xerox. `www.xerox.com`.

[88] F. Yin and C. L. Liu. Handwritten Chinese text line segmentation by clustering with distance metric learning. *Pattern Recognition*, 42:3146–3157, 2009.

[89] C. Zeng, W. Jia, and X. He. An Algorithm for Colour-based Natural Scene Text Segmentation. In *Proc. 4th CBDAR*, pages 67–72, 2011.