UNIVERSITÉ FRANÇOIS - RABELAIS DE TOURS

École Doctorale MIPTIS

Laboratoire d'Informatique de l'Université de Tours

THÈSE présentée par :

Arundhati Tarafdar

soutenue le : **12 Juillet 2017**

pour obtenir le grade de : **Docteur de l'université François - Rabelais de Tours**

Discipline/ Spécialité : Informatique

Wordspotting from Multilingual and Stylistic Documents

JURY :

RAMEL Jean-Yves	Professeur, Université François Rabelais de Tours (Directeur)
PAL Umapada	Professeur, Indian Statistical Institute, Kolkata, India (Co-directeur)
RAGOT Nicolas	Maître de conférences, Université François Rabelais de Tours (encadrant)
EGLIN Véronique	Professeur, INSA de Lyon (rapporteur)
JURIE Jean-Christophe	Professeur, Université de La Rochelle (rapporteur)
VINCENT Nicole	Professeur, Université Paris Descartes

To, my loving Family

Acknowledgments

First and foremost, I would like to present my sincere gratitude to my thesis director, Prof. Jean-Yves Ramel and my co-supervisors, Prof. Umapada Pal and Dr. Nicolas Ragot for being there for me and guiding me in this journey. I thank them for the trust and confidence they have put in me. Without their encouragement, guidance and support, this thesis would not have been possible. I thank Prof. Bidyut Baran Chaudhuri for providing me with his expert advice whenever I needed. I am thankful to Véronique EGLIN and Jean-Christophe BURIE for accepting to be the jury members of my thesis. Also thanks to Nicole Vincent for accepting to be the president of my defense. I am specially thankful to Dr. Partha Pratim Roy for being a mentor to show the path of the work, when I was confused and clueless about the path that I was about to start. I am grateful to Indo-French Centre for Promotion and Advanced Research (IFCPAR) for providing me the scholarship to pursue my PhD.

I admire Nilangshu Sir and Yusufda for their support and advice, which gave me courage and motivation to focus on my thesis when I became distracted by my problems. I feel fortunate to have Ranjuda and Prabir as my friend, philosopher and associate during the times. A special thanks to Tanmoy for his intense technical support. I extend my gratitude to Dr. Anjan Dutta and Abhijit for providing me their opinion. I am highly thankful to Kaustuv for his genuine effort to improve my skill regarding the writing of my thesis, without that effort, it was impossible for me to start organizing the thesis. I feel blessed for the cooperation that I have received from Sounak. I would like to thank all my friends, colleagues and the staff members at Université François –Rabelais de Tours for making my experience at the Lab so pleasant. A special thank to all my friends, colleagues and the staff members at Computer Vision and Pattern Recognition Unit of Indian Statistical Institute, Kolkata. You are all just like a family to me, thank you for all the smiles and light moments that I have shared with you.

My friends, thank you for being there always. I owe my deepest gratitude to my family for bringing out the best in me. Ma and Baba, I have no words to show my reverence for guiding me to be what I am today.

Last but not the least, I want to convey a big heartfelt thank you to Google for their search engine, email service and scholar product. I was never alone in the journey for the availability and efficiency of those products.

Résumé

Les outils et méthodes d'analyse d'images de documents (DIA) donnent aujourd'hui la possibilité de faire des recherches par mot-clés dans des basesd'images de documentsalors même qu'aucune transcription n'est disponible. Dans ce contexte, beaucoup de travaux ont déjà été réalisés sur les OCR ainsi que sur des systèmes de repérage de mots (*spotting*) dédiés à des documents textuels avec une mise en page simple. En revanche, très peu d'approches ont été étudiées pour faire de la recherche dans des documents contenant du texte multi-orienté et multi-échelle, comme par exemples dans les documents graphiques. Par exemple, les images de cartes géographiques peuvent contenir des symboles, des graphiques et du texte avecdes orientations et des tailles différentes. Dans ces documents, les caractères peuvent aussi être connectés entre eux ou bien à des éléments graphiques. Par conséquent, le repérage de mots dans ces documents se révèle être une tâche difficile.

Dans cette thèse nous proposons un ensemble d'outils dédiésaux images de documents géographiques pourle repérage de mots (*keyword spotting*). L'approche proposéerepose sur plusieurs originalités. Premièrement, lors des prétraitements, nous proposons de générer une représentation structurelle de bas niveau du contenu des documents, séparant les éléments textuels des éléments graphiques. L'originalité ici vient du fait que l'information est produite à la fois au niveau pixel (par des méthodes de filtrage) et à un niveau structurel élémentaire (analyse et classification de composantes connexes). Par ailleurs, chaque niveau d'information aboutit à la création de cartes de probabilités au lieu de fournir pour chaque région de l'image une décision stricte (texte ou graphique). Ces cartes de probabilité peuvent être utilisées séparément ou agrégées pour extraire différents types d'information (identification de leurcontenu, repéragede contenu textuel, etc.). Ces différentes approches et niveaux d'information ont été utilisés pour évaluer leur qualité pour une tâche de séparation entre la couche texte et la couche graphique. Ellesont été comparées avec différentes autres méthodesde la littérature, tant dans le domaine fréquence que dans le domaine spatial. Une fois cette description structurelle élémentaire obtenue (séparation texte-graphique), un niveau de description lexical est rajouté au document en séparant les éléments textuels connectés entre eux pour obtenir des caractères et les identifier par des classificateurs. Ici, des descripteurs invariants à l'échelle et à la rotation sont utilisés.

Partant de là, la méthode de *spotting* permettant la recherche d'un mot-clé procède en plusieurs étapes. L'initialisation s'effectue en recherchant au niveau lexical les éléments (caractères) correspondants à la requête et ayant été reconnus avec un bon tauxconfiance par le classificateur. En effet, à cause de la complexité inhérente aux documents (dégradations, liaisons inter-caractères ou liaisons texte-graphique), certains caractères de la requête peuvent ne pas être identifiés clairement ou induire des ambigüités dans la recherche, ce que nous préférons évité en premier abord. En partant de ces éléments textuels stables bien identifiés, et en prenant en compte leur position, taille et orientation, nous estimons

les régions candidates correspondant aux parties manquantes de la requête. Afin d'identifier ces éléments manquants, nous utilisons une méthode à base de points d'intérêts pour confirmer leur présence dans les régions candidates.

Nous avons effectué des expérimentations à la fois sur des cartes numérisées en anglais et enbengali. Les résultats expérimentaux démontrent que la méthode est efficace pour repérer les mots ainsi que les emplacements dans des documents graphiques étiquetés par texte. Le jeu de données et la vérité terrain correspondante ont été rendus publics afin que d'autres chercheurs puissent se comparer et faire de nouvelles propositions.

Mots clefs : Analyse d'images de documents, repérage de mots (*word spotting*), documents graphiques, recherche d'information, séparation texte-graphique, filtrage, vectorisation, cartes de probabilité, points d'intérêts (SIFT), Bengla.

Abstract

In the light of increased usage of electronic media, Document Image Analysis (DIA) provides the capacity to quickly retrieve document images containing specific text content even when no transcription is available. As a part of this document retrieval technique, lots of works have been done already on OCR or even on word spotting systems, which are dedicated to highly structured textual documents (with Manhattan like layout). Whereas still very few solutions exist for the retrieval of multi-oriented and multi-scaled text from non-structured digitized documents. From application perspective, there are numerous documents which are unstructured or not so well structured. For example, geographical document images may contain symbols, graphics, and texts in any orientations and sizes. Furthermore, presence of multi-oriented characters, connections between characters, intersections of texts and symbols with graphical elements, etc., are common in such documents. Consequently, word spotting in these documents proves to be a challenging task. We have focused towards this area and worked for the proposition of a robust word spotting system dedicated to multi-oriented and multi-scaled text that can occur in geographical document images.

The originality of the proposed framework comes from the following ideas : Firstly, with some pre-processing, we propose to generate a low level structural representation of the content of the documents. Secondly, pixel level and structural level information for the next steps of the analysis is used. Instead of using a classical text/graphics separation method, a probabilistic approach is used to provide a final decision for each region inside the image. Here, we propose set of classification methods that could provide soft decisions in terms of probability maps. The content of each of the two levels and associated probability maps can be characterized and processed individually or jointly according to the specific content type : recognition of the characters and of the words, looking for missing characters. Finally, information from all these layers can be combined together according to the content of the query images to spot the desired content (text) in an image dataset. The proposed approaches for the text layer separation are compared with the different available methods of text-graphics separation both in frequency domain and in spatial domain. Due to mis-recognition in classifier, some of the query parts might not be spotted during initial spotting. With the help of probability maps, we can choose the candidate elements having good recognition score through classifier for initial spotting of some queries.

Incidentally, mis-recognition occur due to the structural complexity of graphical documents as well as, over connectivity between components. In such scenario, position, size and orientation of the partially spotted query are noted. Thereafter, using those information we estimate regions(candidate regions) of missing query parts. Key point detection algorithm is used to confirm about the presence of the missing parts in the candidate regions for possible spotting. We have performed experimentations both on English and Bangla scanned

maps. The experimental results demonstrate that the method is effective to spot words as well as, locations in text labeled graphical documents. We have built a dataset along with corresponding annotations for experiment of our proposed methods. Also, the dataset will be publicly available for other researchers to allow comparison and future propositions.

Keywords : Document Image Analysis, Word Spotting, Graphical documents, Text-graphics separation, Vectorization, graphical Documents, Bangla, Probability map.

Table des matières

Liste des tableaux

Table des figures

Chapitre 1

Introduction

"Begin at the beginning," the King said, gravely," and go on till you come to the end; then stop"
- Lewis Caroll, Alice in Wonderland

Contents

Abstract

In this chapter, we present a brief introduction on about how the necessity of digital image processing and pattern recognition system are growing up. We also formally define the word spotting problem in graphical document images followed by the motivation and the contributions of this thesis. Finally, we present the organization of this thesis at the end of this chapter.

1.1 Context

With the progress of human civilization, we have discovered countless facts and have invented numerous technologies. Thus, our immense inquisitiveness is continuously increasing the sphere of human knowledge. To keep track of the evolutionary growth of knowledge, documentation is necessary. To serve this purpose, cave painting began roughly 40,000 years ago. Later, writing was invented for documentation at around 3000 B.C. Nearly 100 A.D. onwards, human civilization started preservation of information with inscription on papers. A library is a typical example of an information storage system, where information is kept using papers.

Conventional paper based information storage systems suffer from several drawbacks. Some of the major problems are that papers are prone to deteriorate due to aging ; it is difficult to find a specific piece of information from such elements and these materials suffer from lack of portability. Though a piece of paper is cheap, these systems are indeed costly as information density on papers is low. Especially, recent advances in communication technology, information technology, and other computer related technologies have increased the process of information generation by a large extent. For example, as on September, 2015, there are 4,967,181 articles in the English Wikipedia. If we try to print all of them on papers, it might be practically impossible. It takes a large amount of time, cost, and man-power to back up a paper based information storage system to another similar type information storage system. Paper also allows extremely limited collaboration. For example, if n number of information seekers needs the same paper-based documents at the same time, typically we need to have n number of copies of the same paper-based documents to be available. After documentation in a paper based storage, it is hard to modify the information. Thus, to change a piece of information, we need to publish a newer version of the book. Note that, recent advances have increased the speed of information updating ; remote access is not possible to access any paper based document. A piece of fact that was relevant and true yesterday, may not be relevant or true today. Consequently, storing these types of data in a paper based system would be simply meaningless. In an ecologic point of view, we can also mention that generation of papers originates deforestation and other pollutions.

Computer based digital information storage and retrieval systems resolve most of the problems of conventional paper based information storage systems. However, there are large amount of information already stored in paper based systems. Many of them are already old and in poor condition. For example, as estimated by Google, on August, 2010, there were approximately 129,864,880 paper based books in the world, that were printed on papers. Digitization of these documents, by scanning them, is probably the best possible way to save the information stored in them. Only scanning is not enough. We need to retrieve the texts stored in them and tag the texts with the corresponding scanned digitized documents or files. This would enhance the searching ability in the scanned documents. To attain this, the domain, called digital image processing, came up in 1960s [Wikimedia, 2016]. It uses computational intelligence based techniques to scan, store, retrieve, and manage digital images. The archival activity through digitization and image processing promises durability, re-usability and redistribution of the information stored in them. It also provides permanent or temporary storage of information according to the specific requirement. As a result, we can easily manage the modification and disposal of any document. Some of

the best beauties of digitization are that it provides portability, easy sharing, controlled access, remote access, etc. Moreover, we can backup multiple copies of the same data in distantly located servers to avoid any unnatural destruction. Due to dense information storage capability, a digitized system is comparatively cheaper and environment friendly.

In the last century, we have gone through notable discoveries, inventions, and population growth. They have caused us to store huge amount of data in paper based documents. Due to aging, a major part of these documents, primarily stored in libraries, individual households, offices and archives, are already in degraded condition. These documents can be categorized into several prominent classes. Geographical documents, i.e., maps are definitely a prominent category among them. Note that, political, environmental, and other changes generates continuous alteration of geographical maps. Hence, they are essential to keep track of meaningful changes. Therefore, digitization of degraded geographical documents is an urgent and important task. Governments of several countries and other organizations are consequently investing in the research projects that deal with digitization of degraded printed documents. Some of them are specially targeted for digitization of degraded geographical documents.

1.2 From CBIR to Content-spotting in Graphical Documents

Digitization of documents is nowadays happening through mobile devices and spreading through internet. The broad access to digital documents through Internet and new mobile devices has led to the generation of large volumes of data in digital form. If we want an effective usage of this huge amount of data, we need automatic tools to allow the retrieval of relevant information. Document Image is a particular type of information that requires specific techniques of description, recognition and indexing. The "Document Image Analysis (DIA)" research community is working since many years on different challenges, linked to the automatic recognition of text (handwriting recognition, OCR, layout analysis etc.) and linked to the recognition of pre-definite symbols into line drawing documents. However, less amount of research works has been done in the direction of processing documents with multi-oriented and multi-scaled text mixed with graphics ; e.g. geographical maps. Considering some aspects, these type of documents looks much more like natural images than like documents. For the analysis of such kind of textual document images can be achieved by some conventional techniques of computer vision domain.

Since few years, the researchers in computer vision field has concentrated on different approaches for retrieving images based on the existing contents in the image, known as "Content-Based Image Retrieval (CBIR)". Instead of using text-based descriptions, a system of CBIR deals on properties that are inherent in the images themselves [Smeulders et al., 2000]. Hence, the feature-based description provides a universal expression in contrast with the numerous scripts (Latin, Greek, Bangla etc.) used all over the world in documents. In addition, such methods seem to be more robust to noise and degradation as they are not trying to transcribe or understand all the content of a document but just to decide if a specific content corresponding to a query is present or not.

The user is in charge of formulating the queries and using these queries, the process of retrieval is performed. Different types of queries can be used when defining a CBIR system

and the role of the user is a key point in the development of a CBIR application. The user can provide a sample image to represent the prototype of what the person is looking for. The visual content of the query should reproduce similar visual features as a sub-part of the target images.

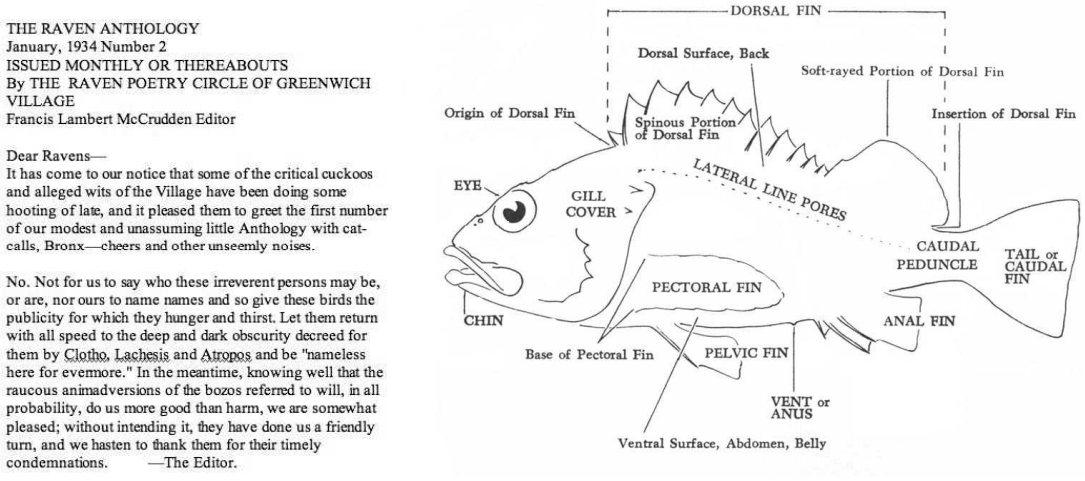
The adaptation of these approaches in the context of document image indexing gives rise to what has been called **keyword, or word or even symbol spotting systems** [Manmatha et al., 1996] [Rusiñol and Lladós, 2010]. The goal of such systems is to provide an alternative to full OCR or full retro-conversion systems that try to interpret all the content of a document image in order to generate a complete editable version of the initial document. Performing OCR for degraded or non-structured documents is highly error prone and less accurate.

Although there are some algorithms for **word-spotting** available, to be discussed later in section 2.4, to process structured document written in main languages [Doermann, 1998], few efforts are done towards graphical documents with multi-oriented and multi-scaled text (like geographical maps). Other than these kind of documents, some documents contain mainly graphical elements with noisy background and broken foreground and text written in non-conventional languages e.g. various Indian scripts ; such as Bangla, Hindi, Tamil, Telegu, Oriya etc.

For retrieval of relevant documents from such huge database and for their categorization, multi-oriented, multi-scaled and multi-scripted word-spotting technique could be very useful. It is in this context that a joint project between two research teams from **Indian Statistical Institute, Kolkata, India** and **Université François Rabelais de Tours, France** has been funded by the **Indo-French Centre for the Promotion of Advanced Research (IFCPAR)**. The project is to take benefits of the past experiences of the two groups towards the development of multi-lingual and multi-script word spotting method that will be able to process graphical documents as well as textual documents. In this project, the main Indian languages like Bangla, Devnagari and Gurumukhi, along with English and French languages has to be considered as the research aspects between these two countries.

1.3 Content Spotting into Graphical Documents : The Difficulties

Based on their content, document images are usually classified as : i) Textual document images, ii) Graphical document images. In Figure 1.1, we have shown examples of both textual and graphical document images. Unlike textual document images that contain structured textual components, graphical document images contain unstructured textual and line drawing elements. The graphical documents that have been studied and processed in the DIA community are mainly geographical documents, engineering drawings, musical scores, electronic diagrams, and architectural plans. The scope of our work is also to process graphical documents but in a different manner, compared to the previous works. Our goal is not to recognize and understand all the parts or elements inside the images [Smith, 2007] but to compute both **visual and lexical indexes** that will allow a fast retrieval of documents, containing a specific content, specified by the user through a query. It is



THE RAVEN ANTHOLOGY
January, 1934 Number 2
ISSUED MONTHLY OR THEREABOUTS
By THE RAVEN POETRY CIRCLE OF GREENWICH VILLAGE
Francis Lambert McCrudden Editor

Dear Ravens—
It has come to our notice that some of the critical cuckoos and alleged wits of the Village have been doing some hooting of late, and it pleased them to greet the first number of our modest and unassuming little Anthology with catcalls, Bronx—cheers and other unseemly noises.

No. Not for us to say who these irreverent persons may be, or are, nor ours to name names and so give these birds the publicity for which they hunger and thirst. Let them return with all speed to the deep and dark obscurity decreed for them by Clotho, Lachesis and Atropos and be "nameless here for evermore." In the meantime, knowing well that the raucous animadversions of the bozos referred to will, in all probability, do us more good than harm, we are somewhat pleased; without intending it, they have done us a friendly turn, and we hasten to thank them for their timely condemnations. —The Editor.

(a) (b)

FIGURE 1.1: (a) An example of textual document image. It contains only textual components, (b) An example of graphical document image. It contains both textual components and graphical components (line drawing).

noticeable that it could be interesting to define a new spotting system dedicated to graphical documents that combines the advantages of the both possible approaches : **Query by Content and Query by String** when researched element is a text part (word). The off-line indexing steps have then to be revisited to extract (as mentioned before) as well **visual signatures as structural and lexical ones**.

Between all the categories of graphical document, we will focus on **grey level geographical maps** as it seems to be rich type of graphical documents, in terms of heterogeneity of content. This special type of graphical document images may contain the texts related to graphic parts, specific textures and symbols. Again, in these documents, the texts are usually multi-oriented and multi-scaled. The languages and the fonts of these texts may differ. To reduce the complexity in terms of source of information that can be used to extract content, we have decided to not work on color information. That is why, we decide to use only binary or grey level maps. We can see examples of geographical document images with multiple scripts in Figure 1.2. The image features of the characters vary with respect to the writing style of a particular script. In this research work, we have proposed a generalized, as well as specialized, approach of spotting words in multi-script geographical documents. Note that, geographical documents are a special category of graphical document, hence have some special properties, compared to other kinds of graphical documents e.g. building plans, mechanical instrument designs etc. Most of graphical documents are structured (not as text but with structure : electronic, architect, scores etc.). Unlike those documents, maps are less structured but with least conventions. If we focus specifically on Word Spotting, the task seems to be much easier for textual documents compared to graphical documents. Some of the prominent difficulties that we face while spotting contents into graphical documents are as follows :

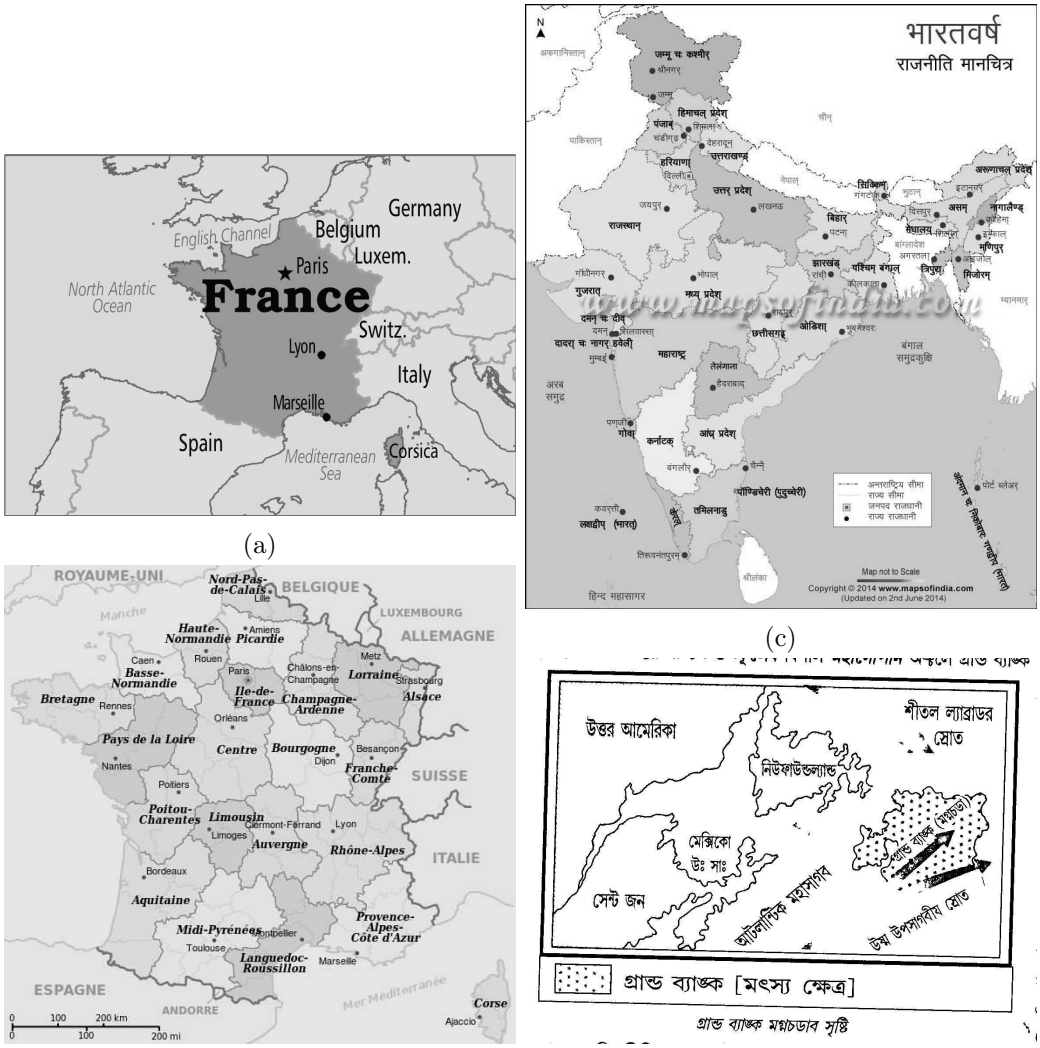


FIGURE 1.2: (a) Image showing France with neighboring countries. It contains texts in English, (b) Image showing the details map of France. It contains texts written in French, (c) Image showing the map of India. It contains texts in Devanagari, (d) Image showing the part of North America, contain texts in Bangla script.

- ✍ **Heterogeneity and variety of the contents :** Text, graphic, lines, curves, hatched, filled, colored etc. these are the basic elements which are present in graphical documents.
- ✍ **Relation and proximity between (heterogeneous) elements :** For example, a name of the river means a proximity between a line and a text to link them. So, the graphical components may intersect and intertwine with other graphical components, as well as with the textual components. The primary reason behind this is as follows. In maps, texts related to the names of different places and symbols related to different geographical components that may co-occur along with territorial boundaries.

This problem is illustrated Figure 1.4a. It can be noted that the word "সন্দেশখালি" (in Bangla) have intersected and intertwined with several graphical components. The structures of the graphical components of a document image may be different. Consequently, it becomes hard to find attributes that would help to identify the graphical components. An instance of the problem is illustrated in Figure 1.4b.

- **Multiple orientations :** Textual components can have multiple orientations, such as, horizontal, vertical, curvy-linear, etc. For example, in Figure 1.4c, the word "ইউক্রেন" (in Bangla) is horizontally placed and the word "আটলান্টিক মহাসাগর" (in Bangla) is placed in a curvy-linear fashion.
- **Unstructured documents :** The textual components do not lie in a specific location or portion of a document. Instead, they are usually found in a scattered way covering the entire document.
- **Unpredictable fonts, spaces and scales :** A geographical document may even contain text characters printed in different fonts, sizes, styles, etc. For example, in Figure 1.4c words "ইউক্রেন" (in Bangla), "আটলান্টিক মহাসাগর" (in Bangla), "বেরিং" (in Bangla) and "ওশেনিয়া–রাজনৈতিক" (in Bangla) have different font-faces, font-sizes, and font-styles. If the texts are written in a script that supports multiple cases, a map may contain letters with both the cases. Even a single word may contain multiple cases. For example, in Figure 1.4d, the word "Godavari R." and the word "Halley" contain both upper and lowercase letters. Undesirable presence of any component and noise may cause joining of multiple text components into a single one. It may also cause fragmentation of a single textual component into multiple ones. We demonstrate this issue using Figure 1.4e. In that Figure, the characters of the word "EQUATOR" are broken due to noise, whereas the characters of the word "MOZAMBIQUE" are merged due to noise and other undesirable components. Texts, in these documents, may have different inter-character spaces depending on the annotation style. To exemplify this, we use Figure 1.4f. It shows inter-character spaces present in the word "JHARKHAND". Here, we would like to mention that there may be a complete word or a part of word in the inter-character spaces of another word.
- **Script properties :**Textual components belong to different documents are comprised of different scripts. We have taken English and Bengali scripts into consideration in our work. Every script has its own properties or limitations to deal with, that we face when we try to separate them as textual components from graphical component or try to segment or recognize them for retrieval purpose. Properties of English script is very much familiar and well known that need no elaboration or description.

  Bengali, a standout amongst the most prominent dialects in India and national dialect of Bangladesh, depends on the Bangla script. Handling of Bengali is not quite the same as that of the Roman (English) script because of various reasons. The quantity of vowels and consonants is vast in Bengali despite the fact that there is no upper and lower case separation. More than 200 compound characters (called yuktakhars) are framed by joining at least two consonants. The state of a yuktakhar might be totally unique in relation to that of its individual segment characters. Vowels and a couple of consonants go about as modifiers when put beside another consonant. At the point when characters frame a word, unlike English script , here generally the individual

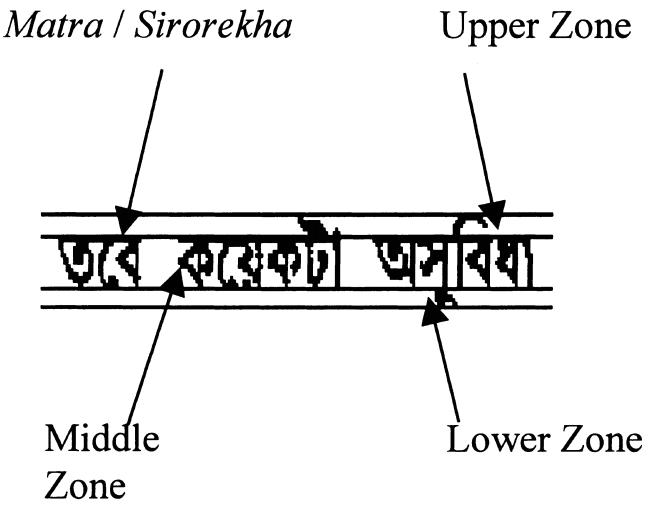


FIGURE 1.3: Different text zones in a line of scanned Bengali script(Figure credit : [Sural and Das, 1999])

letters get associated by a head line called matra or sirorekha as shown in Figure 1.3. Subsequently, development of consolidated characters is a standard instead of a deviation in this script. Singular characters show up in the upper zone, center zone and in addition in the lower zone of a content line. Encourage, a similar character may show up in one zone in its ordinary shape and in an alternate zone while going about as a modifier.

1.4 Objectives and Contributions of the Thesis

Based on above discussions, content-spotting in graphical documents becomes a challenging task. In this thesis, we have proposed some methodologies for **content (mainly text) spotting of graphical document images**. To provide an effective solution, we work for the proposition of a spotting system dedicated to multi-oriented and multi-scaled text.

The contributions of the thesis are as follows :

- We propose to generate and use at the same time **pixel, lexical and structural level information** in order to better index the **heterogeneous content** of a graphical documents. All these different types of signatures, which are pixel, lexical and structural level information have to be merged in a unified representation of the images content.
- Another strong objective is to **construct soft or probabilistic indexes** during the off-line indexation step. The use of **machine learning approaches during the generation of the signatures** describing the regions of interest (ROI) detected inside the images allows to **incorporate these soft and probabilistic information** (that we will call "probability maps") in addition to the region's signatures. Then, instead of storing binary decisions concerning the appurtenance of a pixel to a

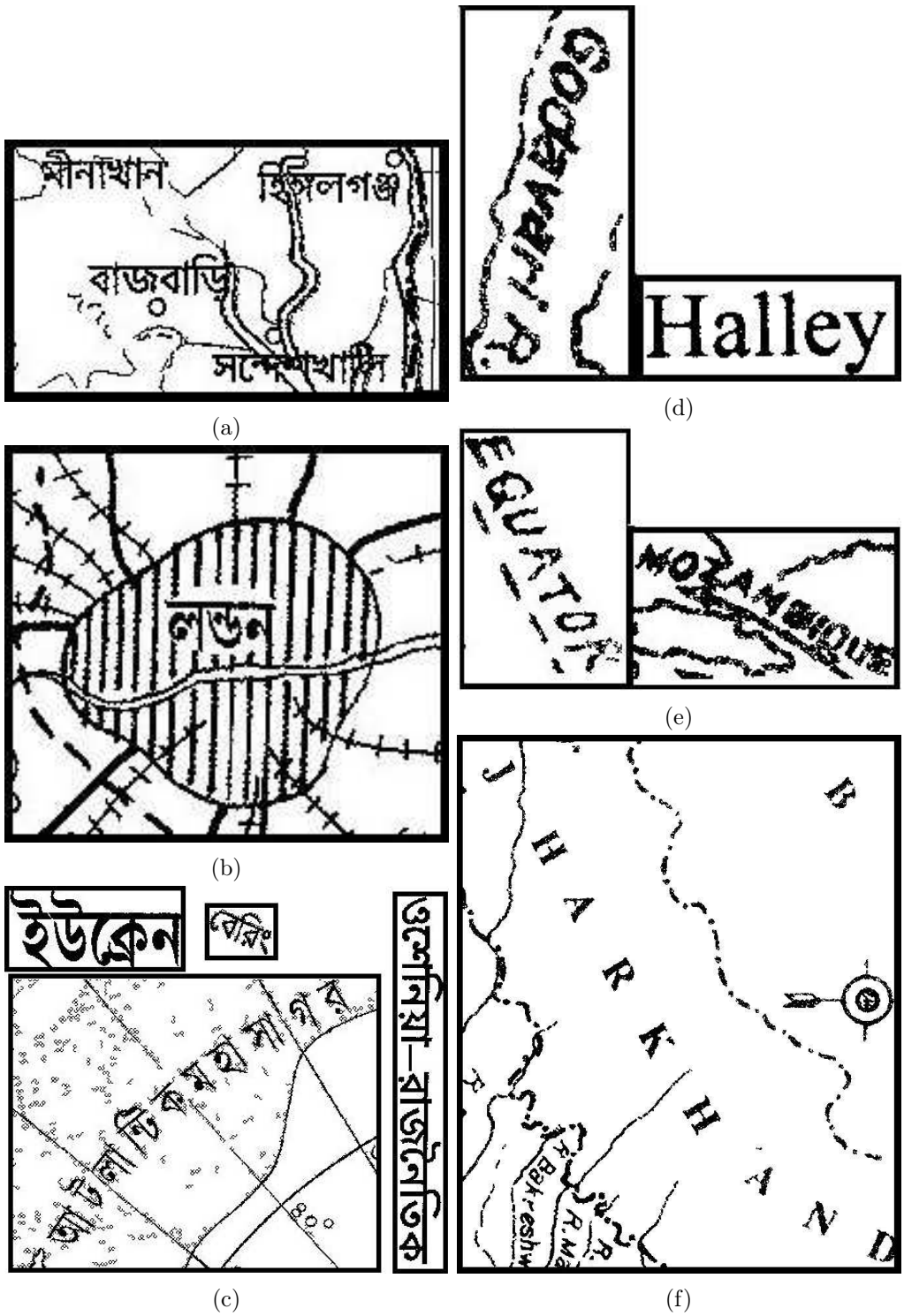


FIGURE 1.4: (a) Overlapped textual contents and graphical contents, (b) A number of structure of graphical components, (c) Textual components having multiple orientations, fonts, sizes and styles, (d) Words are in multiple cases (both upper and lower cases), (e) Broken and touching text components, (f) Words with inter-character spaces.

specific ROI or information layer (text, graphical symbol, hatched area, ...), a set of probability maps can be computed individually or conjointly and can then constitute one part of the indexes that we will use during the querying step (on-line).

- ✍ Using previous works done on graphical documents (vectorization, retro-conversion of engineering drawing into CADS formats etc.) and the concept of probability maps, we propose to extract and compute a variety of signatures and indexes with more semantic information than classical low level image features (like key points, shape descriptor etc.). In the proposed system, **lexical information** coming from an OCR module based on **water reservoir** technique [Pal et al., 2003], **structural primitives** coming from a vectorization [Ramel et al., 1998], and **local key point** detection and matching technique like SIFT [Lowe, 1999] are used for the computation of the indexes. This **multi-level characterization** constitutes a new way to compare the content of an image with the content of a query (Image or String). The concept of probability maps will be helpful to combine the information coming from different types of signatures according to the content of the query, to spot the desired content (text) in an image dataset.
- ✍ With the help of probability maps, an initial spotting of some queries is sometime possible by selecting regions having high probability scores for a specific type of signature (lexical ones for example). These ROI could then be selected as seeds for a deeper analysis using complementary features (SIFT key-points for example). Incidentally, mis-recognition can occur due to the structural complexity of graphical documents as well as due to the presence of multiple connected components. In such scenario, the **intrinsic characteristics** (position, size and orientation etc.) of the **partially spotted regions** matched with one part of the query could be a **very useful source of knowledge** to realize the final and entire spotting of the query (or for the rejection of a false alarm). Thereafter, **incremental spotting** process using probability maps is proposed to cope with variability and heterogeneity present in graphical documents as well as noise and degradations. The probability values and the complementary features are used to better estimate the final regions containing the missing parts or just to confirm about the pertinence of the candidate regions for possible spotting.
- ✍ As we take benefit of methods initially used for graphical document understanding rather than spotting, and even if they differ from several aspects, the proposed approaches for signatures and probability maps generation are compared, when it is possible, with the main methods of the literature like for example text-graphics separation methods.
- ✍ Concerning the retrieval parts, to overcome the lack of ground truthed data, the dataset, we have built and used have been made publicly available online [1] : for other researchers to allow comparison between our system and future propositions in several languages. This would not only help us in this project, but also help the future researchers, who would work on word spotting in geographical document images. Additionally, for ease of access and for future use, we have developed two software solutions : one tool is developed, to annotate the data which tag a unique label, mi-

1. `https://github.com/arundhati87/fluffy-pancake`

nimum enclosing bounding box and another one tool is to separate text and graphics from grayscale graphical documents by the implementation of one of our proposed text-graphics separation approaches. We have performed experimentation on the datasets, both on English and Bangla scripted scanned maps. The experimental results demonstrate that the method is effective to spot words in such graphical documents.

1.5 Organization of the Thesis

The organization of the chapters in this exposition is as follows :

The Chapter 2, of the thesis discusses about the literature survey related to the line of work we have done regarding this content retrieval system. Here, we have tried to draw a conceptual understanding of the state-of-the-art and its connection to our problem, related to text-graphics separation, touching character segmentation, multi-oriented character recognition and finally word spotting/content retrieval.

The Chapter 3, of the thesis discusses about the data sets with annotations that we have generated for our experimental benefit. In this regard, we also discuss about the corresponding annotation tool, implemented to generate those annotations.

In Chapter 4, we propose different approaches through pixel and structural level indexing towards low level structural representation of the content of the documents i.e. text and graphics representation from our document images. After this low level representation, this chapter progressed towards the subsequent text processing techniques which are (i) touching character segmentation, and (ii) multi-oriented character recognition through lexical level indexing.

In Chapter 5, we present our proposed approaches related to online retrieval process, which are (i) word localization, (ii) noisy/missing character estimation and (iii) word spotting. Here, we propose the adaptation of SIFT-based approach in the context of text character localization, i.e., word spotting in graphical documents.

All the experimental details regarding proposed works are given in the Chapter 6. Results i.e. Precision, Recall and F-measure from every stage of our proposed work which are related to, text-graphics separation, character recognition and word spotting are reported here in tabulated form.

Conclusions and future scopes of the work are given in the last Chapter 7. As an supplementary information, the summary of few existing theories used in our proposed works are given in the Appendix A Section.

Chapitre 2

Literature Survey

"To read well, ..., is a noble exercise"
- Henry David Thoreau

Contents

Abstract

In this chapter, we present an overview of the state of the art about graphical document analysis techniques that are relevant for every stages of our framework of word spotting in graphical document images. The stages included are text-graphics separation, touching character segmentation, multi-oriented and multi scaled character recognition and finally content retrieval. We have also reported the performances and overall classification of these related works along with their main drawbacks in synthetic tables at the end of each section.

2.1 Introduction

In this chapter, we have done a literature survey to get the ideas of the state-of-the-art for each possible step that could occur in a content spotting system dedicated to graphical documents i.e. text-graphics separation, character recognition and touching character segmentation.Related works about word spotting and content retrieval systems as well as few other related works that have been done specifically on graphical documents are also reported in the last sections of this chapter. As a starting point of this state of the art we can mention, a complete survey of digital map processing techniques from 2014, available in Chiang et al. [Chiang et al., 2014]. Researchers in the area of text-graphics separation have investigated a considerable number of approaches until the 2000 years. The comprehensive surveys of these methods is summarized and presented by Jung et al. [Jung et al., 2004]. A survey of strategies and approaches for character recognition is done by Impedovo et al. [Impedovo et al., 1991], Trier et al. [Due Trier et al., 1996] and for character recognition in Indian script has been considered by and Pal and Chaudhuri [Pal and Chaudhuri, 2004]. Besides, advancements and open issues in touching character segmentation has been presented in the studies by Saba et al. [Saba et al., 2010] and Kumar et al. [Kumar et al., 2013]. Finally, spotting in different kind of document images is broadly discussed in the survey by Doermann [Doermann, 1998] and more recently in 2016 by Alaei et al. [Alaei et al., 2016].

2.2 Performance Evaluation Metrics for Content Spotting Tasks

As a first part of the literature analysis, we would like to introduce how content spotting solutions could be compared and evaluated in terms of performances. As this work is focused in text spotting inside graphical document, we will focus of metrics proposed for text extraction but most of these metrics can be extended to evaluate more general spotting tasks.

The robustness of a localization technique can be estimated at different levels. In [Kumar et al., 2007] precision and recall are calculated in pixel level. The precision and recall accuracy of this pixel level measurement is computed as :

$$\text{Recall rate} = \frac{\text{Number of searched elements correctly identified}}{\text{Number of searched elements in ground truth}} \tag{2.1}$$

$$\text{Precision rate} = \frac{\text{Number of searched elements correctly identified}}{\text{Number of searched elements detected}} \tag{2.2}$$

$$\text{Harmonic mean} = \frac{2 * \text{Recall rate} * \text{Precision rate}}{\text{Recall rate} + \text{Precision rate}} \tag{2.3}$$

Pixel level calculation of precision and recall as mentioned above is a simpler and deeper measurement. The problem here is to that we need to have pixel level ground-truth for the corresponding performance measurement. Pixel level ground-truthing is a very tedious tasks almost impossible to achieve except on synthetic images.

Furthermore, pixel level analysis is not always necessary for end user purpose. Also, it provides an incomplete information about retrieval of portions of components which could constitute a sufficient seed for the detection of a specific researched element.

Consequently, from the concept of component wise checking, text regions could be detected as a complete component or part of a component. In addition, text could be detected word wise or character wise or partially which cannot be assumed as any known text but a part of it. Also, in text detection measurement, we cannot conclude with binary decision that whether a text object is completely detected or not.

Hence, few other measurement modifications are proposed by [Lucas et al., 2003] in ICDAR 2003 reading competition to adopt a flexible notion of a match. For a text character component locating ground truth preparation, it is next to impossible to agree exactly with the bounding rectangle for a text ground truthed by a human tagger. For a given query (in this case, find all the word-region rectangles in an image), we have a ground-truth set of targets, T and the set returned by the system that to be tested, which we call estimates, E. They have defined the match m_p between two rectangles as the area of intersection divided by the maximum area of the bounding box which contain both rectangles. The value 1 would be for identical rectangles and zero for rectangles that have no intersection. For each rectangle in the set of estimated the closest match is found in the set of targets, and vice versa.

Hence, the best match $m(r, R)$ [Lucas et al., 2003] for a rectangle r in a set of Rectangles R is defined as :

$$m(r, R) = max(m_p(r, r'))|r' \in R \tag{2.4}$$

Then, they have proposed new more forgiving definitions of *precision* and *recall* :

$$p' = \frac{\sum_{r_e \in E} m(r_e, T)}{|E|} \tag{2.5}$$

$$r' = \frac{\sum_{r_t \in T} m(r_t, E)}{|T|} \tag{2.6}$$

The standard *f-measure* to combine the precision and recall figures into a single measure of quality is proposed as mentioned below in equation 2.7. The relative weights of these are controlled by α, which we set to 0.5 to give equal weight to precision and recall :

$$f = \frac{1}{\frac{\alpha}{p'} + \frac{1-\alpha}{r'}} \tag{2.7}$$

Later on, in 2005, Wolf and Jolion [Wolf and Jolion, 2005] proposed a new equation for calculating precision and recall in an optimized way. This proposal is accepted in further ICDAR robust reading competitions for scene images till ICDAR 2013 [Karatzas et al., 2013]. Wolf and Jolion proposed a combined performance measurement, where precision and recall are calculated in matrices. Matrices are composed by taking all set of ground truth bounding boxes and resultant bounding boxes of text components. For all detected rectangles, the amount of overlapping is calculated with all ground truth rectangles. There could be one-to-one, one-to-many or many-to-one overlapping. With these overlapping taken into consideration, new precision and recall measurement is given in equations [Wolf and Jolion, 2005] below.

Recall and Precision matrices for single image :

$$Recall(G, D, t_r, t_p) = \sum_i \frac{Match_G(G_i, D, t_r, t_p)}{|G|} \tag{2.8}$$

$$Precision(G, D, t_r, t_p) = \sum_j \frac{Match_D(D_j, G, t_r, t_p)}{|D|} \tag{2.9}$$

Where, G_i is ground truth rectangle, D_j is detected rectangle, $i = 1$,..number of ground truth rectangle, $j = 1$,..number of detected rectangle. The two rectangles are considered as matched one only if their overlapping satisfies a threshold. where $tr \in [0, 1]$ is the constraint on area recall and $tp \in [0, 1]$ is the constraint on area precision.

$$Match_G(G_i, D, t_r, t_p) = \begin{cases} 1 & \text{if } G_i\text{matches against a single detected rectangle} \\ 0 & \text{if } G_i\text{does not match against any detected rectangle} \\ f_{sc}(k) & \text{if } G_i\text{matches against several } (\rightarrow k) \text{ detected rectangle} \end{cases}$$

$$Match_D(D_j, G, t_r, t_p) = \begin{cases} 1 & \text{if } D_j\text{matches against a single detected rectangle} \\ 0 & \text{if } D_j\text{does not match against any detected rectangle} \\ f_{sc}(k) & \text{if } D_j\text{matches against several } (\rightarrow k) \text{ detected rectangle} \end{cases}$$

$f_{sc}(k)$ is a parameter function of the evaluation scheme which controls the amount of punishment which is inflicted in case of scattering, i.e. splits or merges for bounding boxes between detected bounding boxes and ground truth bounding boxes. If it evaluates to 1, then no punishment is given, lower values punish more. In our experiments we set it to a constant value of 0.8. Recall and Precision matrices for multiple images :

$$Recall(\overline{G,D}, t_r, t_p) = \frac{\sum_k \sum_i Match_G(G_i^k, D^k, t_r, t_p)}{\sum_k |G^k|} \tag{2.10}$$

$$Precision(\overline{G,D}, t_r, t_p) = \frac{\sum_k \sum_j Match_D(D_j^k, G^k, t_r, t_p)}{\sum_k |D^k|} \tag{2.11}$$

where, $k = 1..N$, N is total number of images. In the case of N images, we compare several lists, $G_k \in \overline{G}, k = 1..N$ of ground truth rectangles with several lists, $D_k \in \overline{D}, k = 1..N$ of detected rectangles. Combined single value for Recall and Precision and performance of the system :

$$Recall = \frac{1}{2T}\sum_{i=1}^{T} Recall(\overline{G,D}, \frac{i}{T}, t_p) + \frac{1}{2T}\sum_{i=1}^{T} Recall(\overline{G,D}, t_r, \frac{i}{T}) \tag{2.12}$$

$$Precision = \frac{1}{2T}\sum_{i=1}^{T} Precision(\overline{G,D}, \frac{i}{T}, t_p) + \frac{1}{2T}\sum_{i=1}^{T} Precision(\overline{G,D}, t_r, \frac{i}{T}) \tag{2.13}$$

$$Performance = 2 \cdot \frac{Precision \times Recall}{Precision + Recall} \tag{2.14}$$

The parameter T is a granularity parameter which controls the trade-off between the computational complexity of the evaluation algorithm and the precision of the integration approximation. However, it is not likely that the object related measures change sharply after changing the quality constraints in very small steps. Consequently, they have assumed $T = 20$.

During our work of text-graphics separation, for performance measurement we have decided to report the accuracies both on pixel based [Kumar et al., 2007] calculation and bounding box based [Wolf and Jolion, 2005] calculation.

2.3 Text-Graphics Separation

This section provides a brief overview of all the different works related to text-graphics separation techniques in binary and grayscale graphical document images to gain an understanding of intuition behind each method. These document images contain both a graphical and a textual layer. The texts are multi-oriented, multi-spaced and multi-positioned to label the corresponding graphical objects or areas in the images. The performances reported in the relevant papers are also highlighted in the final table.

The Text/Graphic Separation step can be considered as a source of inspiration to implement a ROI detection step that should be present in all the segmentation-free spotting systems. According to the relative alignment and orientation of the texts, corresponding grayscale graphical document images can be separated into three types :

- ✍ The images, where texts are positioned closely and angled horizontally or vertically. Such images e.g. comics books, official forms etc. contain texts of similar size. A sample image is shown in Figure 2.1a.
- ✍ The images, in which texts are aligned in various positions with horizontal or vertical orientation, as shown in Figure 2.1c. Engineering drawings and floor architecture maps belong to this category.
- ✍ Images, where texts are not only located in multiple positions, but also they are aligned in multiple orientations. These texts could lie together or far apart from each other in a linearly or curvilinear fashion. The font size of these texts also varies. Examples of these classified documents are shown in Figure 2.1b and Figure 2.1d. Geographical maps are the kind of graphical document images that belongs to this category. As per the classifications mentioned, complexities increase chronologically for text-graphics separation problem.

Existing text-graphics separation methods can be broadly classified into the following different categories.

- ✍ The first group of methods is based on connected component analysis (CCA). These methods propose different heuristics based on the black blobs coming from the binarization of the respective document images. Practically, these methods work most effectively for type I of documents as mentioned in previous section.
- ✍ On the other hand, second group of methods process the images by splitting them into sub-parts. To deal with the definition of the size of these sub-parts (grid or sliding windows) multi-level analysis have been proposed, where equivalent approach is applied on every level to extract text. These methods perform good at both type I and type II documents.
- ✍ A third group of methods uses elementary primitives like strokes or edges to detect the text regions.
- ✍ Texture based methods have also been studied for text detection. This group of methods is completely independent from CCA. Such methods applied many other processes like machine learning approach. texture based approach etc. These third and fourth category of methodologies can handle range of type III documents as well.
- ✍ Finally, some proposed methods are dedicated specifically to multi-oriented and multi scale text detection in graphical document. The main methods will be described at the end of this section.

With all the approaches described below, it is hard to reproduce and compare all with a single dataset. Also, their way of performance measurement are not always the same. Hence, for each category, we are only able to report the results of all of these works in synthesis tables to estimate them in a very rough manner.

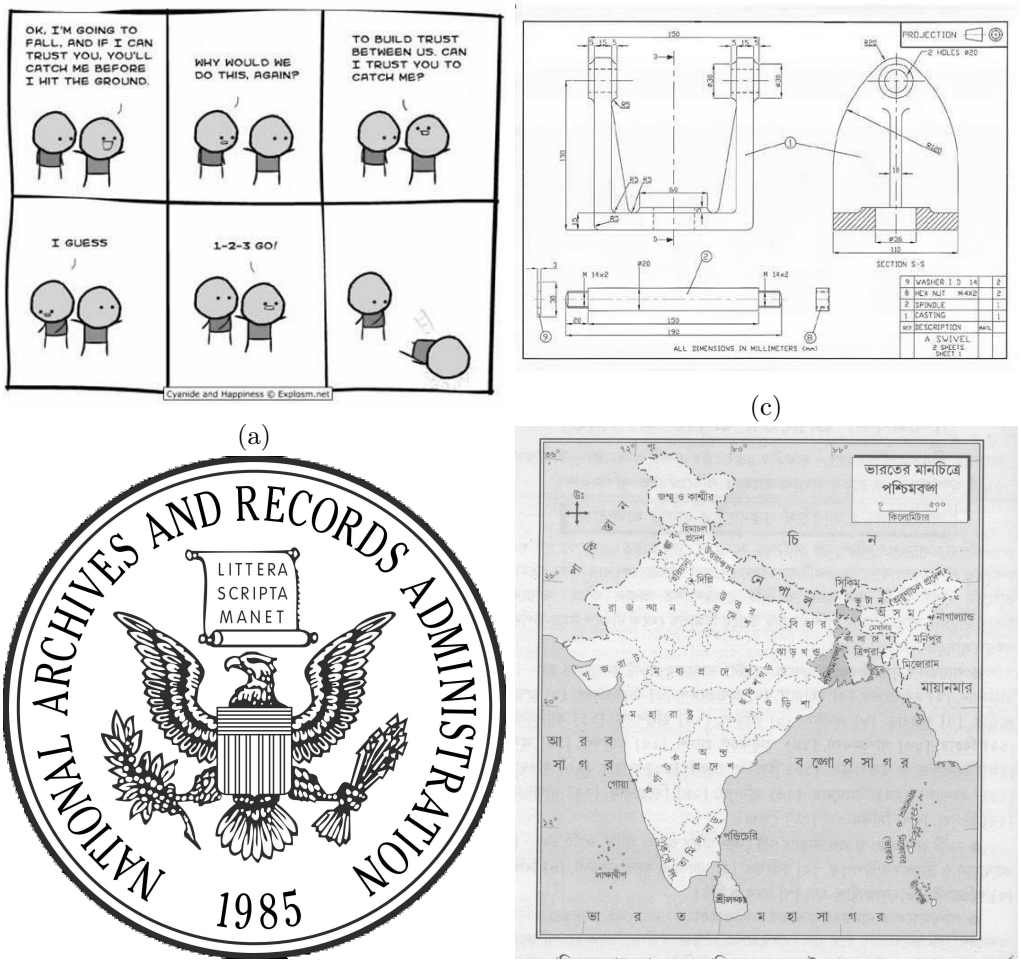


FIGURE 2.1: Examples of different graphical documents (a) Cartoon document image, (b) Logo document image, (c) Engineering drawing image, (d) Geographical document image.

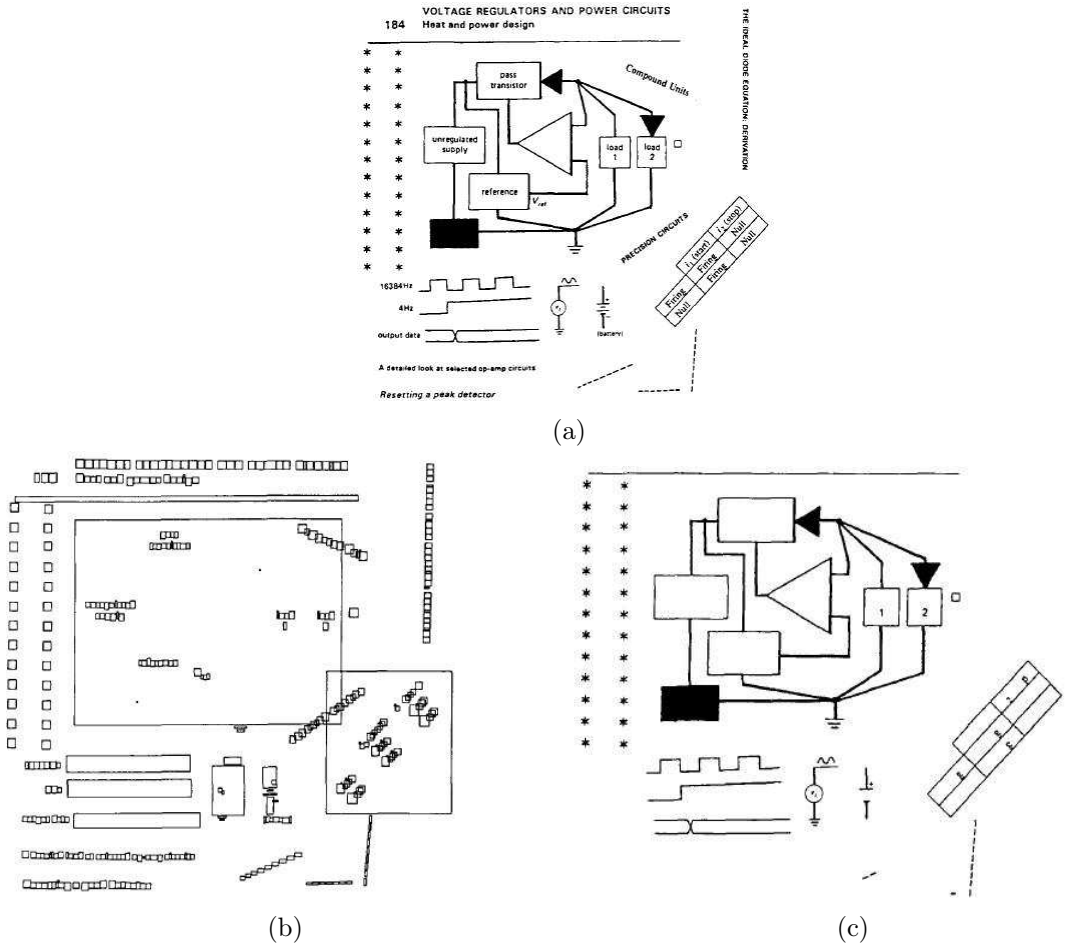
2.3.1 Connected Component Analysis (CCA) based techniques

Since late 1980s, researchers have faced with the problem of text-graphics separation in different kinds of graphical documents. For these kind of problems, connected component analysis is safely regarded as the primary solution.

In 1988, Fletcher and Kasturi [Fletcher and Kasturi, 1988], have proposed CCA for text and graphics separation. The connected components (CCs) obtained as a result of CCA are filtered by their area to differentiate them, component wise as text or non-text. In order to differentiate according to area, at first, most populated area and average area of CCs are computed. An example of using this approach is shown in below in Figure 2.2. The components that are identified as text, are grouped linearly with their neighborhood using Hough transformation. Hough transformation is used to compare the centroid of the selected text component with the centroids of its neighborhood components. If positions

of all the centroids are found to be linear, then those components are considered as group of text or string.

Similarly, Gloger [Gloger, 1992] and Su and Cai [Su and Cai, 2009] proposed CCA and grouping of them as string using Hough transformation to detect text from official forms and engineering drawing respectively. The works [Pierrot et al., 1995], [Hase et al., 1997], [Pouderoux et al., 2006], [Ahmed et al., 2011] and [Zhang et al., 2012] also suggest the wide usage of CCA to separate text and graphics from graphical document images. Besides, Strouthopoulos & Nikolaidis proposed [Strouthopoulos and Nikolaidis, 2008] CCA classification based on Self Organizing Feature Map (SOFM) and fuzzy classification for text and non-text separation. Furthermore, Rigaud et al. [Rigaud et al., 2013] used CCA and applied topological filtering and character grouping for text detection in comic books.



(a) (b) (c)

FIGURE 2.2: (a) Test Image, (b) Rectangles enclosing the connected components of test image, and (c) Graphical portion after text string separation.(Figure credit : [Fletcher and Kasturi, 1988])

In some part of the literature, along with CCA additional post-processing based on geometric and topological information coming from binary blobs have been proposed.

In [Lu, 1998], different features are used to detect text components from Engineering

drawings. CCA is adopted to remove the long linear graphical components. The features, size of the CC, and the gap between the selected CC and its neighborhood along with the local stroke density are used to discriminate text CCs from graphical CCs. These features are useful because they showed different values for text and graphical components, e.g., unlike graphical components, text components generally follow an average size, inter character difference in size are much smaller than inter graphical objects size differences and local stroke densities are much higher for text than graphics. After, morphological operation like erosion and dilation operation are performed on text components. These operations group the text and make a string as a single component. Again another CCA is done to choose those components and decide about the final text regions.

A work [Tan and Ng, 1998] was proposed with CCA and pyramidal analysis. First they have completed CCA and filtered CCs of particular sizes as text components. Next, from input image, pyramids are obtained by repeatedly and uniformly averaging pixel intensities in a non-overlapping 2×2 block of pixels. The pyramid is obtained by regular averaging of pixel intensities. As each succeeding copy is generated, the image becomes smaller by half in both the horizontal and vertical axes. This causes the connected components of the image to naturally move towards each other and merge at a particular stage of the construction. A pyramid allows attention to image for detailed or, oppositely, outlined examination. Hence, filters are applied at many level of pyramids to detect text component of various dimensions. The limitation of this work lies in choosing the specific levels of pyramid for analysis.

Further, Cao and Tan [Cao and Tan, 2001] introduced integrated steps to come up with effective text and graphics separation. Next, morphological method is used to remove solid graphical components. In these kind of documents different symbolic notations or textures occur as part of graphical objects. Dashed lines are one of those common objects. If any group of neighboring small straight line segments exist towards a particular direction having similar size distributions, then those are considered as dashed lines. In this proposed work, these dashed lines are detected using total linear regression on each CC and removed from the existing text components as part of graphics which are derived after first level of CCA. From remaining CCs, large components are denoted as graphics and small components as text. Besides, the graphical components may hold touching texts. Consequently, for touching text separation, those components are decomposed into strokes for further analysis. Stroke size of text is much smaller than graphics. Hence, presence of multiple small size strokes are considered as part of textual components and separated from corresponding graphical components. After the separation, damaged textual components are restored using thickening operation. Finally, character clustering is processed using dilation operation to confirm about presence of text string or group of text.

Another efficient integrated method was proposed by Tombre et al. [Tombre et al., 2002] using CCA. In that paper, they have calculated a histogram of the sizes of the bounding boxes of each of the CC. Consequently, from the histogram obtained, they have considered the densely polled area as the guidline for choosing the text components. The consideration is based upon the fact that unlike graphical components, texual components follow regular and repetitive font sizes. Thus, CCs belonging to the densely polled area are thresholded and filtered as textual components. After this priliminary filtering, for each of the text component, minimal enclosing rectangle is computed. Again, in a similar way

the elongation and density thresholds are calculated from the minimal enclosing rectangles of textual components. Using these thresholds a second stage of filtering is performed on the remaining components. Next, Hough transformation is computed towards horizontal, vertical and diagonal direction using the centroids of the texual CCs. From the corresponding transformation towards each direction, the highest voted direction is determined as the alignment of text string. Mostly texts are present as string. It is assumed that in case of touching text with any graphical component, part of the string is touching, but remaining must be already detected as individual text component during CCA. Depending on this fact, using the text alignment touching texts are estimated by Hough Transformation. Moreover, intersection points or junction points, where three or more strokes intersect each other inside a component and length of linear parts of a components are also used to detect touching texts.

Synthesis about CCA methods :

However, to separate text from graphical documents entirely CCA based analysis has some limitations. The prime limitation is, if any text component is touching with a graphical object, CCA outcomes that object as graphical one along with touching text component, as a single component. As a result, CCA cannot separate that touching text from the graphical object. There are very rare proposal exist to handle touching text based on stroke based analysis integrated with CCA, but still the multiple integrations is an overhead for the system and chances of error increases in every step of integration. Also, some graphical objects look alike textual objects from the structural perspective. Consequently, to handle such scenarios further improvement was required to achieve enhanced outcome.

Table 2.1 reports main methodology of CCA based techniques along with dataset used, performance accuracy with the publication details.

2.3.2 Sub-windows and grid based methods

In this category, a first work was proposed by [Wu et al., 2005], [Chen and Wu, 2009] for text extraction. Firstly, input image is segmented into equal sized non-overlapping blocks and different thresholds are calculated for automatic text, graphics and background segmentation. According to gray level intensity grouping, here first the image is decomposed into distinct object planes. Consequently, first an initial plane is selected for analysis, which plane consists of set of non-overlapping blocks using their homogeneous features. Consequently, texts are clustered using single link matching and centroid link matching. Single link matching is the degree of local connectedness between pair of neighboring blocks where one block is already determined as text and another is taken for the decision to be text or not through the connectivity between those blocks. Whereas, centroid link matching is the degree of global connectedness between an unclassified block with text class of the image. At the end of possible classifications of initial plane, remaining unclassified object planes are classified to its closest distance class (text, graphics or background). Finally, CCA is done and it is followed by spatial clustering for grouping characters and confirming about the presence of text strings.

Some researchers propose to use specific method to determine if a sub-parts of the images is a text part or not. As an example, a work [Maguluri et al., 2013] based on

TABLE 2.1: Methodologies and corresponding performances along with their experimental dataset on CCA based techniques

Publication	**Methodology**	**Dataset**	**Performance**
[Fletcher and Kasturi, 1988]	CCA and grouping	–	–
[Gloger, 1992]	CCA and Hough Transform	Official forms	–
[Pierrot et al., 1995]	CCA and string construction	set of IGN topographic maps	–
[Su and Cai, 2009]	CCA and Hough Transform	50 practical engineering drawings	–
[Ahmed et al., 2011]	CCA	90 floor plan images	97% Precision, 99% Recall
[Hase et al., 1997]	CCA and adaptive processing	book cover pages	–
[Pouderoux et al., 2006]	CCA and morphological filtering	set of IGN topographic maps	–
[Zhang et al., 2012]	CCA and local similarity value measurement	scene images	92.4% Accuracy
[Strouthopoulos and Nikolaidis, 2008]	Self Organizing Feature Map (SOFM) and fuzzy classification	mixed-type binary documents e.g. official forms	–
[Rigaud et al., 2013]	Minimal Connected Component Thresholding (MCCT)	comics pages of 1700 lines, publicly available `http://ebdtheque.univ-lr.fr`	76.15% Precision, 75.82% Recall
[Lu, 1998]	CCA, feature analysis, morphological operation	Set of Engineering Drawings	–
[Tan and Ng, 1998]	CCA and pyramidal analysis	Road maps	–
[Cao and Tan, 2001]	CCA, large component removal, character restoration, morphological dilation for grouping	24 large maps	87.1% Precision, 99.4% Recall
[Tombre et al., 2002]	CCA, best enclosing rectangle formation, Hough transform and linear component detection	document images	–

window based HOG feature extraction and a SVM classifier is used for text and non-text classification, [Zhu and Zanibbi, 2013] based on unsupervised feature learning method using Adaboost classification are proposed for text separation.

Jung et al. [Jung et al., 2002] proposed a method first to divide the image into several frames depends on the texture and then used some rough estimation of the frames to decide their content as text or non-text. Finally, they have used texture based two layers classification of the frames into text or non-text using Multi Layer Perceptron (MLP) method.

Some of the works [Pan et al., 2007], [Hoang and Tabbone, 2010] and [Do et al., 2012] proposed work based on dictionary based learning and classification of image patches using sparse representation for text and non-text. The feature set used in the learning are undecimated wavelet and curvelet transformation or K-SVD algorithm or Morphological features.

Pezeshk and Tutwiler [Pezeshk and Tutwiler, 2010a] proposed defect model that is applied on unit square grid of pixels of the image for text identification in input images in order to simulate the type of defects that are specific to characters extracted from map images. They have identified parameters to create artificial training sets that closely model

variety of noise and defects caused by the imaging of the documents. Their system is used HMM for classification of those grid of pixels as text or non-text.

Another approach is proposed in 2011 [Biswas and Das, 2011] on scanned map documents, where the image is divided into a large number of small blocks of fixed size and to decide about a block content, a centralized block along with its neighborhood blocks is processed through radon transformation method to distinguish the centralized block as text if it contain a strong peak after radon transformation.

Synthesis about sub-windows and grid based methods

One of the main problem with sub-window or grid based methods comes from the difficulty to decide the good size of the sub-parts (grid, sub-windows) to consider. Multi-level approaches have been developed to bring solutions to this difficulty, where same procedure is applied in every level to distinguish between text and graphic objects. Besides, sometimes a unit grid or, sub-window does contain a mix of text and non-text. Resultantly, for those grids or sub-windows, it cannot extract the extract the exact boundaries of text and non-text.

Like in previous section, we have also gathered the methods described in this section in below Table 2.2.

2.3.3 Elementary primitives/stroke based analysis

As Connected Components are sometimes not precise enough to detect text regions because of broken and touching characters, some researchers propose to use some more elementary primitives extracted from grey level or binary images.

A proposal [Li et al., 2000] was based on vector i.e. small unit segment of a component without having a connecting junction. First CCA is made and then components are grouped till orientation and spacing between components do not vary to a large extent. After this grouping, vector analysis is processed and short vectors are taken as part of text components. Finally, segmentation free prototypes are matched from detected text components to confirm the presence of text characters.

In 2003, Gllavata et al. [Gllavata et al., 2003] proposed a work, where the gray level image is first converted into edge image. Edge based histogram analysis is done to distinguish text and non-text edges.

In 2006, Liu et al. [Liu et al., 2006] proposed text segmentation using initially stroke based filtering and local region growing approach using neighborhood check.

A work [Zhang et al., 2015a] was proposed canny edge detection method and then CCA to detect text. For CCs, strokes are composed using stroke width transformation and that are grouped using morphological dilation with dynamic structuring element. Then, text strings are searched using k-nearest neighbors from already detected text components. Finally, falsely detected text components are removed using template matching.

Furthermore, multiple works were proposed methods based on stroke analysis. Along with CCA, in [Jain and Yu, 1998] proposed block adjacency graph, [He and Abe, 1996] used maximum likelihood estimation (MLE), [Garcia and Apostolidis, 2000] used variance

TABLE 2.2: Methodologies falls under sub-windows and grid based methods and corresponding performances along with their experimental dataset

Publication	**Methodology**	**Dataset**	**Performance**
[Wu et al., 2005]	spatial variance based analysis	ICDAR'03 competition datasets	–
[Chen and Wu, 2009]	Multi-level thresholding, connectedness measurement and CCA	28 English document images, and 37 Chinese and mixed - Chinese/English document images	99.6% Precision, 99.4% Recall and 99.4% Precision, 99.2% Recall
[Maguluri et al., 2013]	window based HOG feature extraction	floor maps of 30 libraries, publicly available at `http://www.public.asu.edu/~bli24/icassp2013.html`	85.8% Precision, 57.9% Recall
[Zhu and Zanibbi, 2013]	unsupervised feature learning using Adaboost classification	USPTO Images	83.72% precision, 83.72% recall
[Jung et al., 2002]	Multi Layer Perceptron (MLP) method	Several kind of scene images	97% Precision, 99% Recall
[Pan et al., 2007]	dictionary based learning by morphological analysis using sparse representation	set of IGN topographic maps, 1373 characters, 266 strings	Characters - 96% Precision, 92% Recall, Strings - 91% Precision, 85% Recall
[Hoang and Tabbone, 2010]	dictionary based matching by wavelet and curvelet analysis using sparse representation	5 images	Better than [Tombre et al., 2002]
[Do et al., 2012]	dictionary based matching by K-SVD learning using sparse representation	5 images	Better than [Tombre et al., 2002] and [Hoang and Tabbone, 2010]
[Pezeshk and Tutwiler, 2010a]	defect model training	125 images of street labels	96.4% Accuracy
[Biswas and Das, 2011]	block wise radon transformation	Map images	Approx. 90%

of edges and their orientations, [Kim et al., 2002] used clustering and template matching, [Biswas and Das, 2012] used morphological operation and first order approximation differentiation and [Chen et al., 2002] used MRF to extract text.

A method in [Goto and Aso, 2000] proposed block analysis. Weighted histogram and edge analysis is calculated for every block to confirm about a text presence. Finally region growing method is applied by CCA.

Another work [Fu et al., 2006] was proposed by neighborhood analysis of CCs using their size, distance and pixel density. Feature vectors of three neighbors is assumed to follow distribution of Gaussian mixture. Successively, CCA is done and morphological closing operation is followed. After that each component is labeled as text or non-text using Gaussian Mixture Model (GMM) of neighbor characters. Neighbor characters are determined by Voronoi region and Delauney triangulation processing.

Biswas and Das [Biswas and Das, 2013] proposed work based on structural analysis by fuzzy graph matching for classification of text from graphical document images. Pixels with exactly one or more than two neighbors are considered as nodes ; most of these types of pixels appear as end point or junction point of a line of map. Here, a cost matrix is calculated depends on fuzzy similarity between two nodes. That fuzzy similarity value is

calculated based on the Euclidian distance between two nodes and the amount of neighborhood components, they are having. Cost matrix values of text nodes are much lower than non-text node. Hence, using these values text nodes and non-text nodes are distinguished.

Another work [Biswas et al., 2014] proposed Delauney triangulation. Successively, big triangles and triangles, which are having one angle less than 15 degrees are removed. Next, CCA is processed and triangles whose vertices are not really part of single CC are removed. Then, Cyclometric numbers are calculated and the component with number zero is removed. Finally, morphological reconstruction is used for region growing.

Bourbakis [Bourbakis, 2001] has processed character recognition without going for any prior processing. The image is scanned through temporal windows to detect edges. Consequently, from those edges chain codes are derived based on their shapes. Next, those chain codes are classified to characters using fuzzy graph matching. If a character is not isolated from neighboring characters, the temporal window moves left, center and right. Accordingly, for each direction movement chain codes are classified to its nearest character class and finally the movement with highest voted recognition is taken into consideration. In such a way, chain codes are connected to frame characters. Similarly, connection of word frame to generate text line frame also processed.

Furthermore, Chen et al. [Chen et al., 2001] proposed text line extraction method by edge detection and morphological dilation. Morphological dilation is used to extract connected, small oriented edges as shown in Figure 2.3. Text positions are identified by normalization of text line using bilinear interpolation. Consecutively, text recognition is done using sliding window trained with distance map feature. The distance map feature is calculated by the distances between strong edge points within the window. But limitation of this work is that it looks for almost horizontal text.

In 2010, a work [Pezeshk and Tutwiler, 2010b] based on Multi Angled Parallelism by directional morphological operations is proposed. Linear features are extracted from the images by deconstruction of the whole image, followed by reconstruction of lines. Using this method, it can extract text from intersecting linear features.

In 2001, Ye et al. [Ye et al., 2001] proposed method for character extraction based on topological modeling of character strokes using morphological operations. They have used hybrid approaches to extract text from various background. They define different arrangement of strokes to classify edges into character strokes. Also, Lebourgeois [Lebourgeois, 1997] proposed work based on character texture filtering, stroke localization and edge based analysis, [Zhang et al., 2012] based on edge enhancement and local similarity value measurement method.

Similarly, Li et al. [Li et al., 2008] proposed multi-polarity image analysis by graph theory. After graph partitioning based on grayscale intensities, CCA is applied on single pole images to detect textual components. Single pole image means which covers a specific gray range. All single-pole images are generated using graph cut theory. Next, using a SVM classifier and pixel distribution features, texts are selected from single pole images and the multi-polarity text segmentation problem is turned into several single-polarity text segmentation problems.

In 2010, Epshtein et al. [Epshtein et al., 2010] proposed stroke width transformation method, where width of stroke related to each pixel is calculated. Here, stroke is defined

as a band of a almost constant value denoting contiguous or connected part of an image. They have claimed that, searched elements of an image contain certain amount of similar stroke widths. Thus, through filtering they are differentiating text from non-text.

Synthesis based methods based on elementary primitives or stroke based methods

Difficulty arises with elementary primitives or stroke based methods in the decision about which elementary primitive would sufficient enough to distinguish between text and non-text. Both way over or under segmentation of the primitives can lead to ambiguous discrimination between text and non-text. Moreover, due to size variation these methods might lack of scalability and such deeper level analysis can cause to a performance bottleneck in terms of time complexity.

Here also, we have collected the methods described in this section in below Table 2.3.

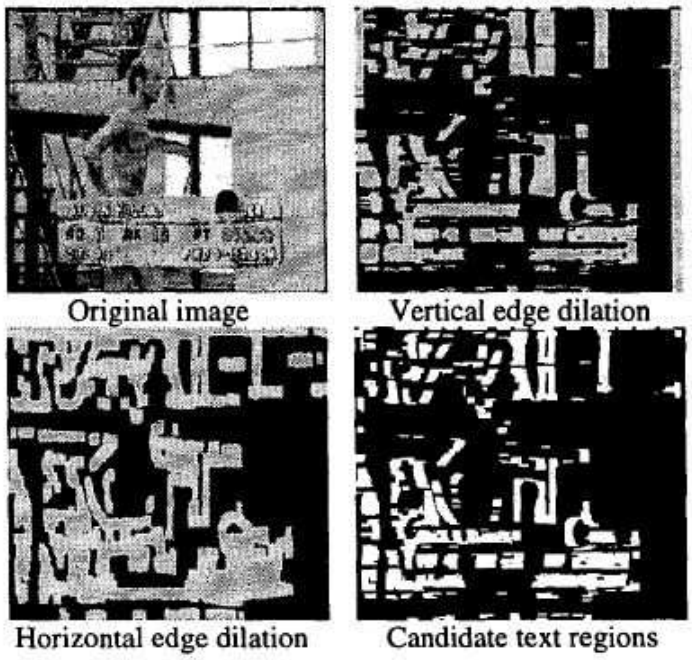


FIGURE 2.3: Text region extraction based on edges.(Figure credit : [Chen et al., 2001])

TABLE 2.3: Methodologies related to elementary primitives or, stroke analysis and corresponding performances along with their experimental dataset

Publication	**Methodology**	**Dataset**	**Performance**
[Li et al., 2000]	CCA, vector analysis, segmentation free prototype matching	Street Level maps	–
[Gllavata et al., 2003]	Edge based histogram analysis	326 video frame images	83.9% Precision, 88.7% Recall
[Liu et al., 2006]	Stroke filtering and local region growing	435 news images from about 4-hours videos of South Korea TV channels	–
[Zhang et al., 2015a]	Edge detection, CCA and stroke analysis	ICDAR 2013 dataset, 400 indoor and outdoor scene images	77% Precision, 66% Recall, 73% Precision, 69% Recall
[Jain and Yu, 1998]	Block adjacency graph and CCA	Advertisement images, Web images, Scanned color images, Video frames	–
[He and Abe, 1996]	Maximum likelihood estimation (MLE) and CCA	–	–
[Garcia and Apostolidis, 2000]	Edge variance and CCA	200 jpeg video frame images	93% Precision
[Kim et al., 2002]	clustering and template matching and CCA	100 magazine images, 200 web images and 1000 video frames	97.6 % Accuracy
[Biswas and Das, 2012]	CCA, morphological operation and first order approximation differentiation	Scanned land map images	Around 90%
[Chen et al., 2002]	MRF and CCA	Images from BBC sports	93.8% Precision
[Goto and Aso, 2000]	weighted histogram and edge analysis for blocks and CCA	Magazine cover, Magazine, Tech papers	Around 90%
[Fu et al., 2006]	Neighborhood analysis by Voronoi region and Delaunay triangulation, CCA, GMM	20 scene images	92.66% Precision, 80.54% Recall
[Biswas and Das, 2013]	Fuzzy graph matching	Scanned map images	89.09% Accuracy
[Biswas et al., 2014]	Delaunay triangulation, CCA, Cyclometric number calculation, morphological reconstruction	446 maps (318 map in Indic texts and 128 map in English)	89.52% f-measure
[Bourbakis, 2001]	Fuzzy graph matching	document image	–
[Chen et al., 2001]	Edge analysis and morphological operation	18000 video frames	98.7% Recall
[Pezeshk and Tutwiler, 2010b]	Multi Angled Parallelism by directional morphological operations	USGS topographic maps	–
[Ye et al., 2001]	Topological modeling of character strokes	A French cheque, and a tourism magazine	–
[Lebourgeois, 1997]	Texture and stroke localization	scene images	–
[Zhang et al., 2012]	Edge enhancement and local similarity value measurement	Real image dataset collected from web images and TV news video frames	92.4% Accuracy
[Li et al., 2008]	Graph theory, partitioning and CCA	2400 multi-polarity text images	98.99% Precision, 81.11% Recall
[Epshtein et al., 2010]	Stroke Width Transformation and Filtering into text or non-text	ICDAR 2003, 2005 dataset	73% Precision, 60% Recall, 66% f-measure

2.3.4 Texture based methods

Coming from computer vision field, and more precisely from works on text detection into natural images and videos, some techniques propose to use texture analysis to localize text regions inside an image.

In [Wu et al., 1997] an end to end automatic text extraction system was proposed. Generally, texts follow certain frequency and orientation. Subsequently, texture segmentation is proposed here for text extraction. Then a set of heuristics is used to find text string within the segmented region. Strokes are used for applying those set of heuristics and Gaussian derivative is used for processing the texture segmentation. In addition, pyramidal approach is employed for handling very large and very small texts. Also, local thresholding is proposed for background noise removal.

In [Kumar et al., 2007] a method based on wavelet is proposed. Globally matched wavelets (GMW) are created from document image. For widespread training, GMW based method detect text regions using Fisher classifier. Finally, post processing is completed using MRF to correct misclassification of detected text.

Furthermore, in 2001, Matti & Okun [Matti and Okun, 2001] proposed method based on edge detection and then block classification of edge-based textures into text or non-text, Liu et al. [Liu et al., 2008] proposed based on pixel density clustering and then texture based Haar wavelet analysis is performed to identify text or non-text.

In 2008, Journet et al. [Journet et al., 2008] proposed a method based on an analysis of the textures within the images, without looking for structure of the pages. The texture features are calculated on a local level at different resolutions, with the help of a sliding window. These features are used to group pixels according to their orientations and frequencies. Finally the homogeneous areas are clustered into text, graphics and background.

In 2015, a work [Mehri et al., 2015] is proposed based on texture based analysis of the document using Gabor filter. After filtering, the pixels are clustered into text and non-text according to their similar textual properties.

Synthesis about Texture based methods

One of the main problem with texture based methods is that we do not have any clue about the grouping count that should be used to get text areas into a single group. Like previous division here also, a deeper level analysis is required which can cause a performance bottleneck in terms of time complexity. Moreover, simple binarized document will not help in such methods to conclude anything about the text locations. In case of clear discriminative situation, easy structural conclusion is not used here.

The methods described in this section, are jotted down in below Table 2.4.

2.3.5 Methods dedicated specifically to multi-oriented and multi-scaled text

Roy et al. [Roy et al., 2008c] proposed method to detect multi-oriented text line from graphical document images. First CCA is approached. Next, very big CCs are removed as graphical components. Succeedingly, text lines are extracted by clustering other CCs

TABLE 2.4: Methodologies related to texture analysis and corresponding performances along with their experimental dataset

Publication	Methodology	Dataset	Performance
[Wu et al., 1997]	Texture segmentation and Pyramidal approach	48 document images	95% Recall
[Kumar et al., 2007]	Wavelet matching and Markov Random Field	33 scene images	76.8% Precision, 73.7% Recall
[Matti and Okun, 2001]	Block based texture segmentation	25 document images	74.32% Recall
[Liu et al., 2008]	Pixel density and texture based analysis	100 cover pages, logo images	75% recall
[Journet et al., 2008]	Texture based orientation and frequency grouping	400 pages of old documents	92% searched elements extracted
[Mehri et al., 2015]	Gabor Filtering and clustering	Six centuries French History document images	Average 90% Accuracy

using their size, linearity and stroke width information. After clustering extremely positioned character components of the cluster are chosen for boundary growing. The boundary growing is applied by external boundary growing of contour points of those extreme characters. In this regard, the direction of boundary growing is decided using orientation of those extremely positioned characters. Correspondingly, orientation of those characters is computed using orientation of water reservoir of inter-character background information of those character pairs of the cluster. In Figure 2.4, one example is given of such text line extraction from map documents.

In [Roy et al., 2009b], it deals with touching characters that generally present in graphical documents. For proper recognition of those characters, it need to be segmented first.

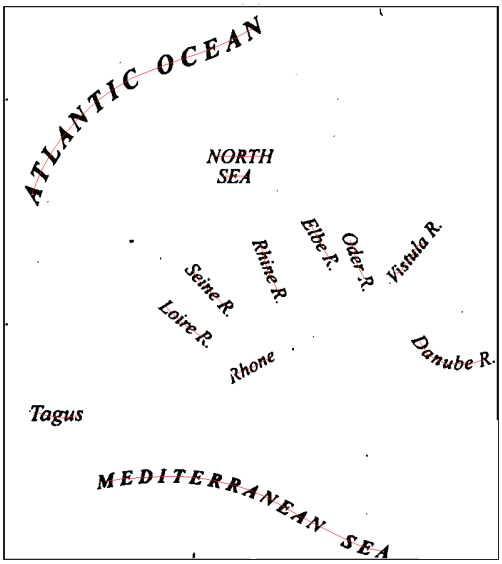


FIGURE 2.4: Text line extraction from map documents.(Figure credit : [Roy et al., 2008c])

Here, initial character segmentation points are found by polygonal approximation of the edges of that particular connected component, that need to be segmented. Consequently, the vertices of the polygon are taken as initial segmentation points, which are merged in a alternative manner using dynamic programming. During merging with different combinations, classifier is used for recognition for every combination of segmentations. The segmentation points with optimized recognition result is considered as final segmentation points.

Also in [Ahmed et al., 2012], method for extracting text touching with graphical components was proposed. First non-touching text components are extracted. These text components are taken as templates. If very few templates are found, then reference dataset is used as templates. Next, from both text and non-text templates SURF key point features are extracted. Ambiguous templates are removed from text and non-text templates. Key points are extracted from unrecognized image portions as well. Finally, these key points and key points from templates, similarity with text and non-text is calculated using nearest distance method to detect touching text portions.

Synthesis about methods dedicated to multi-oriented and multi-scaled text

There are very few works existed specific to multi-oriented, multi-scaled and multi-spaced text present in graphical documents. Also, there is lack of publicly available dataset to compare the existing approaches related to these kind of graphical documents.

Like in previous section, we have pointed out the brief of the methods described in this section in below Table 2.5.

TABLE 2.5: Methodologies of all other various approaches of text separation and corresponding performances along with their experimental dataset

Publication	**Methodology**	**Dataset**	**Performance**
[Roy et al., 2008c]	CCA and boundary growing based on Water reservoir method	maps, magazines, newspapers, advertisements, computer printouts	Approx. 97% of text lines are extracted
[Roy et al., 2009b]	Polygonal approximation and recognition based dynamic programming	maps, newspapers and magazines. Synthetic dataset at url : http ://mathieu.delalandre.free.--fr/projects/sesyd/charseg.html	91.44% accuracy in touching character segmentation
[Ahmed et al., 2012]	SURF and nearest distance method	90 floor plan images	95% accuracy in touching character detection

2.4 Recognition based Spotting in Graphical Documents

The original intention of our work is to spot a specific query or location from set of geographical maps. Consequently, to get the idea of the relevant field we have concentrated on the literature study of word spotting from graphical documents and also from a very few reputed work from word spotting in textual documents.

2.4.1 Frameworks for word spotting

As per our understanding from the earlier survey in word spotting depicted by Doermann [Doermann, 1998], the concept of word spotting achieved in the scanned document images through several specific logical understanding of the ways.

The most fundamental way is the use of OCR to interpret all the text contents [Lopresti and Zhou, 1996] of a document and use a searching technique to spot a query.

Another possibility is based on dictionary based retrieval of important words (nouns, objects etc. - not stop words) [Salton, 1989] via image matching or OCR and finally spot those document using those specific queries made of those important words. Another similar way used, is to prepare an abstract of the document content [Chen and Bloomberg, 1997] and search or spot those documents by searching only through their abstract.

A further approach [Takasu et al., 1994] is used, which first segmented the document according to its physical/ structural layout and then it followed by logical segmentation, which might include text-graphics separation and then it does the word spotting through specific features.

There are many passive applications existing that are related to word spotting, which are logo detection/ recognition [Doermann et al., 1996], image compression [Witten et al., 1994], image matching [Doermann, 1998], image interpretation through their caption [Srihari, 1995] etc. In logo detection, particular word of a particular entity is recognized with its spatial information. In image compression, stock of similar words are indexed in hash table and thus reduce the storage space as a whole. In image matching, a word is used as query and its dictionary based similar words or same words in degraded format are spotted or reconstructed.

The Figure 2.5, shows the classical architecture of a word spotting system dedicated to textual document including a line segmentation step. The Figure 2.6 proposes a more generic framework for content spotting in textual or even graphical documents. No segmentation step is included and an optional learning step is included to define the most adapted

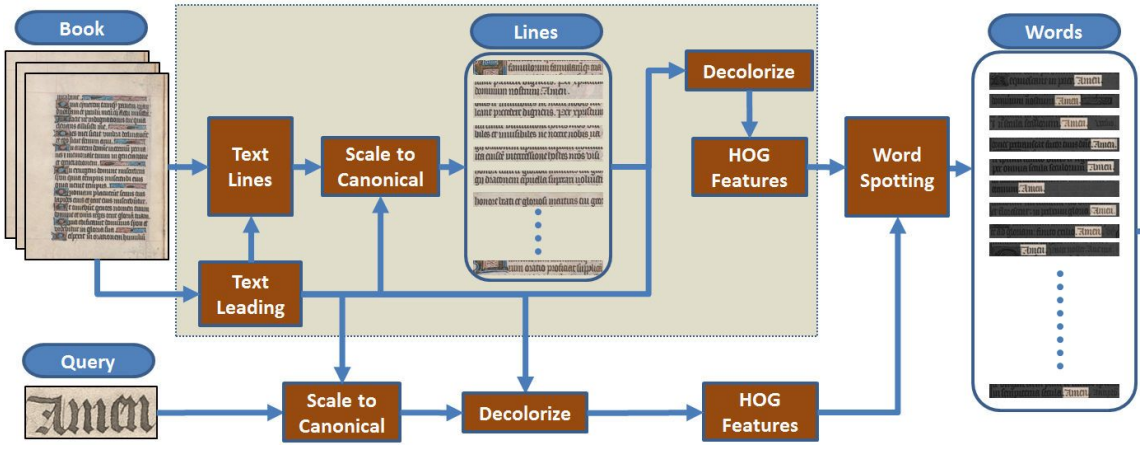


FIGURE 2.5: Classical architecture for a word spotting system dedicated to textual documents (Figure credit : [Pintus et al., 2016])

feature to use according to the types of images to process [Almazan et al., 2014]. It is noticeable that this learning step can be assimilated to new approaches that propose to use convolution neural network [Wang et al., 2012] to **automatically select or construct higher level features** that should be used to index ROI according to the content we want to spot inside the images.

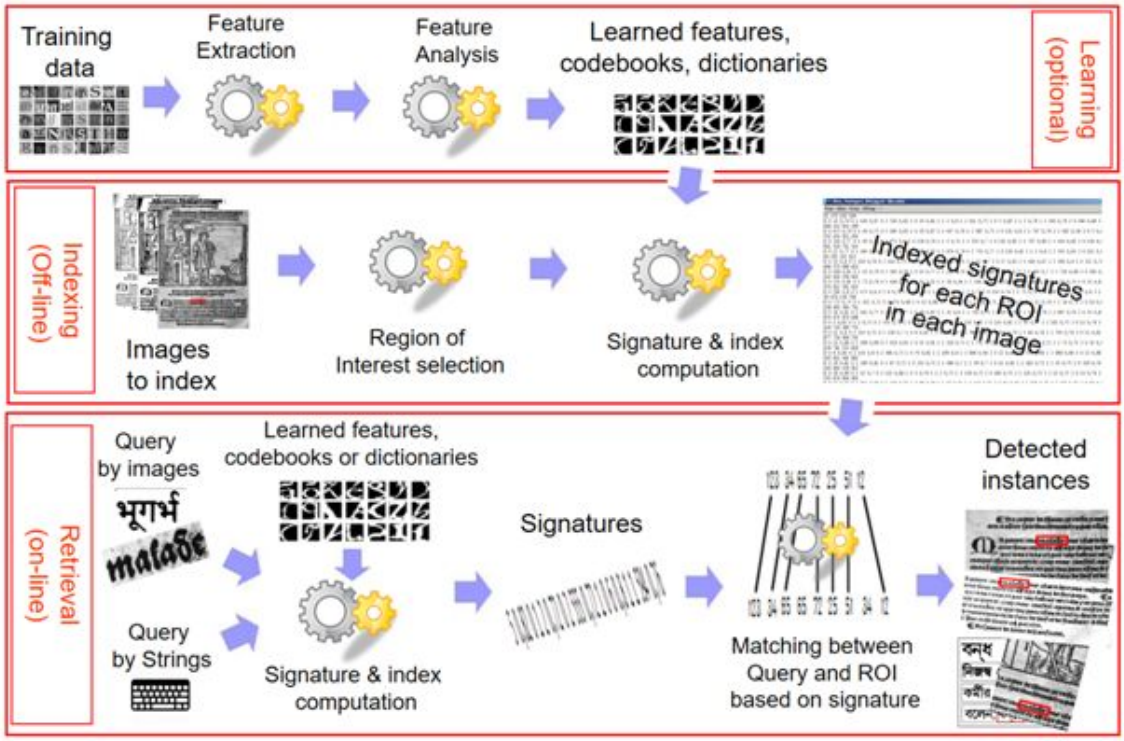


FIGURE 2.6: General Framework of content spotting system

2.4.2 Comparison between word spotting systems dedicated to textual and graphical documents

The word spotting techniques are generally classified based on various criteria : First of all, depending on whether segmentation is needed or not. The research community speaks about segmentation-free or segmentation-based systems. Second, we can dissociate systems that includes learning capabilities from the others. Finally, when spotting systems deal with textual queries, the user can decide to use an image of a word as query ; we then speak of Query-By-Example (QBE) or user can input the researched word using a keyboard and we then speak about Query-By-String (QBS). The query-by-string approach usually requires a model for every character and these methods are often achieved by learning-based approaches, while query-by-example is often achieved by learning free, image matching based approaches. A well-known drawback of learning based approaches is the requirement of a set of transcribed text word or line images for training and it may be costly to obtain. Moreover, it should be done for all fonts, considering variability of writing/font

styles that seems impossible in case of graphical documents. If neither the language nor the alphabet of a document are known or even if writing style or font is different from the learned patterns, learning based word spotting approaches performs poorly. In this kind of situations, **learning free or on-line learning word spotting approaches** might be the only option available. Furthermore, a fair comparison is difficult to perform between learning based and learning free approaches as each of them have their own set of advantages and disadvantages. In the case of learning based approaches, ground truth (GT) is a mandatory requirement for training the system whereas in the case of learning free approaches, GT is not required and it could be more independent of data, language and scripts. As a counterpart, learning based approaches have most often high accuracy than learning free approaches and query-by-string based word spotting can be adapted in this category of techniques whereas learning free approaches is generally achieved by query-by-example based techniques.

According to our objectives, in the following, we propose a different way to categorize word spotting works according their abilities to deal with textual and/or graphical documents.

Spotting methods dedicated textual documents

In the segmentation-based approach, there is a tremendous effort towards solving the word segmentation problem [Gatos and Pratikakis, 2009] [Ghosh and Valveny, 2015]. One of the main challenges of keyword spotting methods, either learning-free or learning-based, is that they usually need to segment the document images into words [Rath and Manmatha, 2003] [Liang et al., 2012] or, text lines [Frinken et al., 2012] using a layout analysis step. In critical scenarios, dealing with highly degraded documents [Louloudis et al., 2009] segmentation is highly crucial. Any segmentation errors have a cumulative effect on subsequent word representations and matching steps. The work of Rusiñol et.al. [Rusiñol et al., 2011] avoids segmentation by representing regions with a fixed-length descriptor based on the well-known bag of visual words (BoW) framework [Csurka et al., 2004].

In 1991, Ho et. al. [Ho et al., 1991] proposed a method of word recognition that consists of a set of serial filters and parallel classifiers, and the decisions for word recognition were the combined consensus ranking based on the input lexicon. Actually this system has combined three different scenarios to handle the recognition in a complete manner. First one is based on isolated character recognition, where characters are well separated. Second one is dependent on context based recognition, where some of the isolated characters belong to a word, are well recognized and the third one is completely word shape based recognition. The choice of the particular method is depended on the quality of character resolution and noise around. The combination of the classifiers decision based on border count and the estimated logistic regression function has drawback of its own.

In 2008, Mesheshaet. al. [Meshesha and Jawahar, 2008] proposed a novel partial matching algorithm which is designed for morphological matching of word form variants in a language. They formulate feature extraction scheme that extracts local features by scanning vertical strips of the word image and combining them automatically based on their discriminatory potential. The success of this work greatly depends on the performance of the page segmentation algorithm and efficiency of the indexing scheme which was taken

as given. The matching scheme also needs to be more general so that it also addresses synonymous word variants that are seen in real-life documents.

In [Gatos and Pratikakis, 2009], Gatos and Pratikakis perform a fast and very coarse segmentation of the page to detect salient text regions. The represented queries are in the form of a descriptor based on the density of the image patches. Then, a sliding-window search is performed only over the salient regions of the documents using an expensive template-based matching.

The recent work of Rodriguez et.al. [Rodríguez-Serrano and Perronnin, 2012] proposes methods of word spotting that relax the segmentation problem by requiring only segmentation at the text line level and it does model-based measure of similarity between vector sequences using Semi-Continues HMM [Rabiner, 1989].

Synthesis about Word spotting methods on textual documents

The state of the art of word spotting in textual documents work feasibly in such document images. However, if a document consist of words that contain highly degraded series of characters or a word consist of a rare combination of character pairs, then it limits the performance of the word spotting system. On the other hand, there are different factors which do not have direct impact to the system but still need to be addressed regarding word spotting problem. These factors are complexities of spotting algorithm in terms of time and space, dealing with image quality of the documents and balance between recall and precision of the spotting system.

Like in previous section, we have pointed out the brief of the methods described in this section in below Table 2.6.

TABLE 2.6: Methodologies of word spotting approaches in textual documents and corresponding performances along with their experimental dataset

Publication	Methodology	Dataset	Performance
[Ho et al., 1991]	Combined approach of isolated, partial or holistic recognition	machine-printed postal words obtained from live mail	98.6% Accuracy
[Rath and Manmatha, 2003]	Shape based feature extraction and matching through DTW	15 images from George Washington manuscript	72.56% Accuracy
[Csurka et al., 2004]	Patch based SIFT descriptor incorporated with SVM classifier	1776 images in seven classes : faces, buildings, trees, cars, phones, bikes and books	–
[Meshesha and Jawahar, 2008]	Partial matching using DTW	4,000 basic word images from English, Hindi and Amharic languages	89.58% recall, 90.81% precision and 90.19% F-score
[Gatos and Pratikakis, 2009]	Block based feature extraction and matching	100 pages of a historical book, 5 keywords	Precision 75.1%, Recall 93.2%
[Rusiñol et al., 2011]	Patch based SIFT descriptor and the later refinement of the descriptors by using the latent semantic indexing technique	George Washington (GW) dataset, Lord Byron (LB) dataset, Persian (PE) dataset	For the GW dataset the mean average precision - 30.42%, recall - 71.1%, For the LB dataset average precision - 42.83% and the recall - 85.86%
[Liang et al., 2012]	Incremental learning , Grapheme representation, hypothetical evaluation between query and word set	a total of 163 unique words (267 instances) have been segmented from the first three pages of Bargrave's Diary	–
[Frinken et al., 2012]	Sliding window based recognition using BLSTM neural network	IAM Dataset, GW Dataset, Parzival Dataset	Approx. 60% accuracy
[Rodríguez-Serrano and Perronnin, 2012]	Query By Example, Semi Continuous - HMM based measure of similarity between vector sequences	630 scanned handwritten French letters, GW Dataset, IFN/ INIT Arabic Dataset	–
[Ghosh and Valveny, 2015]	pyramidal histogram of characters labels (PHOC) based word attributes	The George Washington (GW) dataset, The Lord Byron (LB) dataset, IAM Offline Dataset	GW - 67.7%, LB - 90.45%, IAM - 42.08% Accuracy

Spotting methods dedicated to graphical documents

Rusinol et.al. [Rusiñol and Lladós, 2008] in 2008, proposed a spotting architecture able to index both words and symbols, inspired in off-the-shelf object recognition architectures. Key points are extracted from a document image and a local descriptor is computed at each of these points of interest. The spatial organization of these descriptors validate the hypothesis to find an object (text or symbol) in a certain location and under a certain pose. This work just presents a spotting architecture which is able to tackle with both words and graphical symbols spatially combining state of the art classical features.

Roy et. al. [Roy et al., 2010] in 2010, presented a word indexing architecture in graphical documents to facilitate searching of query word from documents like maps, engineering drawing, floor plan images, electrical diagrams etc. This system is used query-by-string approach where, given a query text word (ASCII/Unicode format), the initial character pairs present in it are searched in the document using nearest neighbor method [Roussopoulos et al., 1995] and they are validated using their size similarity. Next, the retrieved character

pairs are linked sequentially through nearest neighbor method to form character string using the initial pairs' in between distance and angle. Dynamic programming is applied to find different instances of query words. A string edit distance is used to match the query word as the objective function. Recognition of multi-scale and multi-oriented character component is done using the features related to the minimum enclosing circle and convex hull of the character component incorporated with the Support Vector Machine classifier.

A method of camera-based document image retrieval from heterogeneous-content documents using different types of features from different layers of information was proposed by Dang et. al [Dang et al., 2015] in 2014-2015. They used two kinds of features in their work (Locally Likely Arrangement Hashing - LLAH - and SIFT reduced dimensions using PCA). Then, a single hash table method is used for indexing these multiple kinds of feature vectors. In addition, they employ a technique for reducing the memory required for indexing the key points in hash table.

The hierarchical method for text extraction proposed by Gomez et al. in 2016 [Gomez-Bigorda and Karatzas, 2016] where text detection is posed as a search within a hierarchy produced by an agglomerative similarity clustering process over intensity, color, gradient magnitude, stroke width etc. based regions. The grouping process starts with a set of regions Rc extracted with the MSER algorithm [Matas et al., 2004]. This algorithm is used for blob detection, which extract connected components from its corresponding grey level sets of the image. Initially, each region $r \in R_c$ starts in its own cluster of a distinct intensity, color, stroke width, gradient magnitude etc. Then, the closest pair of clusters (A, B) is merged iteratively, using the single linkage criterion $(mind(r_a, r_b)r_a \in A, r_b \in B)$, until all regions are clustered together (CR_c).

The distance between two regions $d(r_a, r_b)$ is defined as follows :

$$d(a,b) = \sum_{i=1}^{D}(w_i * (a_i b_i))^2 + (x_a - x_b)^2 + (y_a - y_b)^2 \tag{2.15}$$

w_i is the similarity feature used for grouping analysis as mentioned above.

Synthesis about word spotting methods on graphical documents

There are very few works existed specific to multi-oriented, multi-scaled and multi-spaced text spotting or retrieval present in graphical documents. Also, there is no common dataset available to compare such existing approaches related to these kind of graphical documents.

Like in previous section, we have pointed out the brief of the methods described in this section in below Table 2.7.

2.4.3 Multi-oriented word decomposition into individual characters

It may sound a little strange to include a study of word decomposition into isolated characters when dealing with word spotting systems but, as we deal with graphical documents, we decided to include this part in the indexation step because of our decision to

TABLE 2.7: Methodologies of word spotting approaches in graphical documents and corresponding performances along with their experimental dataset

Publication	**Methodology**	**Dataset**	**Performance**
[Rusiñol and Lladós, 2008]	Key point extraction and spatial organization of local descriptors	Ten scanned wiring diagrams	Approx. 50% Precision - Recall
[Roy et al., 2010]	Neighborhood pairing and joining through dynamic programming	synthetic dataset of maps	89% precision and 81% recall
[Dang et al., 2015]	Locally Likely Arrangement Hashing - LLAH - and SIFT reduced dimensions using PCA	12 images of linguistic map of France	Average accuracy approx. 85%
[Gomez-Bigorda and Karatzas, 2016]	Agglomerative similarity clustering process over intensity, color, gradient magnitude, stroke width using MSER	ICDAR Robust Reading Competitions datasets	Approx. 80% accuracy

include also lexical information in addition to the visual signatures of the selected ROI.

A lot of works have been proposed to decompose handwritten text into characters or N-grams in order to recognize each individual character with a OCR system. The problem becomes much more complex when we deal with graphical documents that could contain multi-oriented, multi-scale and multi-script text zones. In Figure 2.7, we could see examples of many words belong to a geographical map. Here, as per the Figure 2.7, in most of the cases the characters belong to a particular word, are touching with each other by a single headline. Resultantly, we cannot individually extract those characters through CCA. Except using holistic approaches, in order to spot or recognize these kinds of words, we need to go through individual character recognition, those are belong to a word. To achieve that recognition we cannot solely rely on CCA based techniques in these kinds of scenarios. One possibility to deal with these is to go for touching characters segmentation before individual character recognition. In this regard, we have crawled through the literature study related to character segmentation.

According to the works in the literature, we can classify this problem into various ways. The most of the work [Lu, 1993] [Lee et al., 1996] [Bansal and Sinha, 2002] [Muralikrishna and Koti Reddy, 2011] has been done to segment the set of characters vertically and linearly, where the characters are horizontally aligned parallel to axes. In few other cases, [Roy et al., 2009b], [Roy et al., 2012], [Mancas-Thillou and Gosselin, 2006], [Mancas-Thillou et al., 2005], [Roy et al., 2008b] the method for multi-oriented connected character segmentation are proposed. Also, from methodology perspective many methods [Ramana Murthy et al., 2013], [Roy et al., 2008b] are applied on the binarized version of the image whereas few methods [Lee et al., 1996], [Mancas-Thillou and Gosselin, 2006] are applied on the grayscale image.

Moreover from another point of view, works has been done by segmentation of characters using a linear cut [Garain and Chaudhuri, 2002] or using a non-linear cut [Lee et al., 1996]. Additionally, there are works [Mancas-Thillou and Gosselin, 2006] exists that could be applied in case of multi-oriented component segmentation, even though they are proposed for horizontally aligned component segmentation.

FIGURE 2.7: Different alignments of Bengali words belong to a map

Another point to note is that, most of the work has been done on English scripted documents. Very few works [Ramana Murthy et al., 2013] [Bansal and Sinha, 2002] are exist in Indian regional scripts.

2.4.3.1 Segmentation of English (Roman) script into characters

In 1993, Yi Lu [Lu, 1993] proposed a method to segment the characters of fixed pitch or of proportional fonts. The interval between characters is a constant in case of fixed pitch characters. Here, the interval distribution is obtained from the vertical projection of a text. After obtaining the interval distribution, a conditional check is performed for each components, if any intend values smaller than the fixed interval, that indicate broken characters and larger values indicate touching characters. Once the size of the pitch-box is determined, the touching character segmentation process is initiated through that conditional checking. In case of proportional fonts, peak-to-valley function is used to find break points through which component can be broken into characters.

$$pv(x) = V(l_p) - 2 * V(x) + 2V(r_p)V(x) + 1 \tag{2.16}$$

where $V(x)$ is the vertical projection function, x is the current position, l_p is the peak location on the left side of x, and r_p is the peak location on the right side of x. Sharp minima in the vertical projection are represented as maxima in the pv function. These maxima are then considered as potential break points between the touching characters. This work is proposed using the binarized version of the image.

In 1993, another work is proposed by Rocha and Pavlidis [Rocha and Pavlidis, 1993] based on the graph analysis. Here, a character shape is described by a graph, where edges of the graph are features (strokes or arcs) of the character, and nodes of the graph are

junctions among strokes and arcs. Consequently, sub graph recognition is used to extract already defined prototypes of the characters. Sub graph recognition can better handle both the touching or broken characters as it has not to handle a connected component as a whole but it handles in an exhaustive manner.

In 1996, Lee et al. [Lee et al., 1996] proposed a method based on the grayscale intensities of the image. On grayscale images, specific topographic features such as peak, ridge, hillside, saddle, ravine, flat, and pit of pixel intensities and the variation of intensities is observed in the character boundaries. In the proposed method, the character segmentation regions are determined by using projection profiles and topographic features extracted from the grayscale images. Then, a nonlinear character segmentation path in each character segmentation region is found by using multi-stage graph search [Horowitz and Sahni, 1989] algorithm. Next, on a trial basis recognized portion of the graph is estimated and based on that segmentation is performed through a nonlinear character segmentation paths.

In 2006, Thillou and Gosselin [Mancas-Thillou and Gosselin, 2006] proposed a character segmentation approach that could be applied on grayscale images. It has been noticed that the words are sometimes separated in corresponding grayscale image but due to information loss after binarization they become connected. The approach is based on Log-Gabor filters which exploit simultaneously gray-level variation of the image and spatial location. Slanted or misaligned characters are largely supported by this method because the filter can be applied irrespective of rotation of the image.

In 2016, Farulla et al. [Farulla et al., 2016] proposed method that combines three existing methodologies of touching characters segmentation, which methodologies are individually applied in the literature so far. The strategy is based on a 3-inputs/1-output fuzzy inference system with fuzzy rules specifically optimized for segmenting touching characters. The strength of this fuzzy strategy relies on the possibility to adjust its parameters in such a way that they can fit the characteristics of the data set.

2.4.3.2 Segmentation of Indian scripts into characters

In 2002, Garain and Chaudhuri [Garain and Chaudhuri, 2002] proposed a technique based on fuzzy multi-factorial analysis in Indian scripts. A predictive algorithm is developed for effectively selecting possible cut columns for segmenting the touching characters. Few factors are taken into consideration to find a connected component, which should be further segmented to extract individual character. Those factors are the measure of dissimilarity of a particular component from other components and aspect ratio of the particular component. The dissimilarity is measured between the component of consideration and stored prototypes of the characters. After selecting the component, to select the cut columns for separating the touching characters another multi-factorial analysis is proposed. The factors considered to choose cut columns are the vertical crossing count, the number of black pixels those are encountered in single column scan of the component, the distance of the middle most black pixel from uppermost and lowermost black pixel encountered in that single column scan and continuous trial matching of the sub-part of the component with pre-defined patterns or shape of a possible touching portion.

In 2009, Roy et al. [Roy et al., 2009b] proposed a method which can segment touching

components even if they are oriented in some arbitrary direction. This work depends on the fact that when two or more characters touch each other, they generate a big cavity region at the background portion. Using Convex Hull information of the component, these background information is used to find some initial points to segment a touching string into possible primitive segments (a primitive segment consists of a single character or a part of a character). Next, these primitive segments are merged to get optimum segmentation and dynamic programming is applied using total likelihood of characters as the objective function. SVM classifier is used to find the likelihood of a character. To consider multi-oriented touching strings, the features used in the SVM are invariant to character orientation. Circular ring and convex hull ring based approach have been used along with angular information of the contour pixels of the character to make the feature rotation invariant.

Synthesis about word segmentation methods on graphical documents

There are very few works existed specific to multi-oriented, multi-scaled and multi-spaced touching or merged text segmentation specially related to graphical documents. Also, there is no common dataset available to compare such existing.

Like in previous section, we have pointed out the brief of the methods described in this section in below Table 2.8.

2.4.4 Multi-oriented and multi-scale character recognition

Still associated to our objectives to include lexical information inside the ROI signatures constructed during the indexing step, we also study the previous works done on multi-oriented and multi-scaled character recognition. Let us notice that these kinds of methods are quite different from the ones used in OCR systems.

In Graphical documents, with the increase of structural complexity of the document layout, the problem of character recognition becomes more challenging. This problem has in fact many facets. For example, characters could belong to multiple fonts, multiple scales, multiple orientations/skewing, noisy/degraded environment, touching with other components that might be text or non-text object. Consequently, we have studied the literature regarding this category of works done on character recognition.

In 2000, Adam et al. [Adam et al., 2000] proposed a rotation invariant approach to recognize oriented characters. Here, the Fourier Mellin Transform, is integrated in a global strategy, named as Analytic Fourier Mellin Transform with which multi-oriented and multi-scaled shapes are recognized within the same formalism and without any a priori knowledge. The strategy is divided into two stages. The first one construct a geometric invariant-feature vector from each shape of the character layer. The second one recognize search shape with a selected classifier.

In 2006, Pal et al. [Pal et al., 2006] presented a scheme towards the recognition of multi-oriented and multi-sized English characters. The features used mainly based on the angular information of the external and internal border points of the characters, the features are used invariantly to character orientation. For recognition, modified quadratic discriminant function (MQDF) is used.

TABLE 2.8: Methodologies of text segmentation and corresponding performances along with their experimental dataset

Publication	Methodology	Dataset	Performance
[Lu, 1993]	Segment touching characters of proportional font using peak-to-valley function	–	–
[Rocha and Pavlidis, 1993]	Recognition based processing of images, converted into graph forms	5,282 words (over 24,000 numerals) belonging to a USPS database of real printed addresses	96.5% accuracy
[Lee et al., 1996]	Topographic feature analysis over grayscale intensities and using that multi-stage graph search algorithm to find non-linear cut for segmentation	scanned from the photocopy of five kinds of the real-life documents, such as technical journals, magazines, and some printed materials	Average accuracy approx. 97%
[Garain and Chaudhuri, 2002]	Fuzzy Multifactorial Analysis	11577 Devnagari touching characters, 16714 Bangla touching characters	98.87% (for Devnagari) and 98.63% (for Bangla) accuracy
[Bansal and Sinha, 2002]	Segmentation using multiple pixel based analysis	18 document pages of Devnagari script from two different magazines	85% accuracy
[Mancas-Thillou and Gosselin, 2006]	Log-Gabor filtration of gray level intensities	ICDAR'03 database	–
[Roy et al., 2009b]	Water reservoir based segmentation and modification based on recognition and dynamic programming	real data from map, newspaper, magazines including multi-oriented data	94.44% Accuracy
[Muralikrishna and Koti Reddy, 2011]	Non-linear technique that utilizes data from grayscale pixel values to determine a reliable segmentation path	555 words with nearly 3,000 smeared characters spanning from a corpus of 19 document images	81.98% Accuracy
[Ramana Murthy et al., 2013]	slant correction using headline detection and segmentation using vertical projection profile	130 samples from scene images	55.77% Accuracy
[Farulla et al., 2016]	Fuzzy approach by combined three existing strategies of segmentation	Latin printed and handwritten dataset	–

In another work, Roy et al. [Roy et al., 2008a] proposed method for multi-oriented character recognition lies in these kind of input document images using convex hull and minimum enclosing circle based information of the character components.

In 2009, Chen et al. [Chen et al., 2009] proposed an invariant descriptor using the radon transform, the dual-tree complex wavelet transform [Kingsbury, 1998] and the Fourier transform. The radon transform can capture the directional features of the pattern image by projecting the pattern onto different orientation slices. The dual-tree complex wavelet transform can select shift-invariant features in a multi-resolution way. The Fourier transform can extract features that are invariant to rotation of the patterns. Standard normalization techniques are used to normalize the input pattern image so that it is translation and scale invariant.

In 2010, Pezeshk and Tutwiler [Pezeshk and Tutwiler, 2010a] proposed a method from a different perspective. Their problem domain is restricted to topographic maps, which

contain a small amount of text compared to other forms of printed documents. Graphical components typically intersect with text thus making the extraction of text difficult. The proposed method creates training sets including the types of defects represented by Baird's [Baird, 1993] document image degradation model in order to create pseudo randomly generated training sets that closely mimic the various artifacts and defects encountered in characters extracted from maps. Consequently, two Hidden Markov Models are then trained and used to recognize the text.

In 2014, Yao et al. [Yao et al., 2014] proposed unified framework for text detection and recognition in natural images. The proposed work can detect text and recognize text concurrently using exactly the same features and classification scheme using Random Forest. A new dictionary search method is proposed, to correct the recognition errors usually caused by confusions among similar yet different characters. After character recognition, by keeping false recognition error in mind, they have proposed a new dictionary based search method to correct recognition errors.

Synthesis about character recognition methods, specially on graphical documents

Several works are done specific to character recognition, but very few works exist regarding multi-oriented and multi-scaled character recognition and also very few works exist for non-English or Indian scripted character recognition. However, even though many approaches exist related to character recognition, scanned multi-oriented and multi-scaled character recognition from geographical documents is still a myth.

Like in previous section, we have pointed out the brief of the methods described in this section in below Table 2.9.

2.4.5 Other problems coming with graphical documents

In addition to the previous problems, few other difficulties arise when word spotting is concerned with graphical documents. In this regard, we have surveyed the literature to note those problems and their existing solutions. The two main problems are dynamic grouping characters of collinear or curvilinear alignments to form a full word and automatic interpretation of complete map into different semantic layers.

2.4.5.1 Dynamic grouping of curvilinear characters

In geographical documents texts are aligned in various manners to label graphical objects or places. Mostly, character grouping of a particular word align collinear or in a curvilinear manner in such kind of graphical document images. From below Figure 2.8, we could variety of possible alignments. Among them, in geographical maps, text characters belong to a particular word, align into either collinear manner, or, curvilinear manner or, in rare cases, align-along-road manner. To group and extract such words from graphical documents, few works are proposed in literature in the year of 2013 [Zhang et al., 2013] and 2016 [Xu et al., 2016b], [Xu et al., 2016a].

In [Zhang et al., 2013], they have relied on multi-algorithm paradigm. First, textual

TABLE 2.9: Various methodologies of character recognition and corresponding performances along with their experimental dataset

Publication	Methodology	Dataset	Performance
[Adam et al., 2000]	Analytic prolongation of the Fourier-Mellin transformation applied at component level	Technical documents from the France Telecom network including arbitrary oriented characters	–
[Pal et al., 2006]	Rotation invariant Minimum enclosing circle based feature extraction. recognition based on modified quadratic discriminant function	18232 multi-oriented characters	98.34% accuracy
[Roy et al., 2008a]	Convex hull based feature incorporated into SDVM classifier	multi-oriented English character set	Average accuracy approx. 97%
[Chen et al., 2009]	Radon, dual-tree complex wavelet and Fourier transforms based invariant descriptors based recognition	85 printed Chinese characters	–
[Pezeshk and Tutwiler, 2010a]	Recognition based on training of models based on defect that may arise in the context	125 images of street labels	93.2% accuracy
[Yao et al., 2014]	Recognition from existing literature approaches along with dictionary based error correction	Multi-oriented ICDAR 2011, Chars74K, MSRA-TD500, HUST-TR400, Images harvested from the Flickr website	Average 70% accuracy

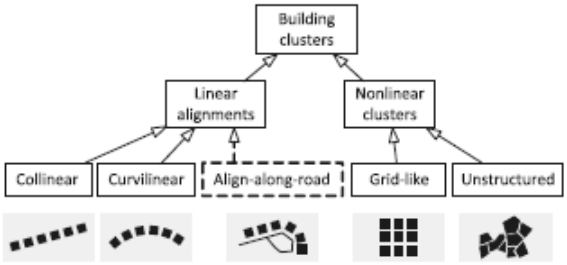


FIGURE 2.8: Possible alignments of multiple similar components (Figure credit : [Zhang et al., 2013])

components are extracted and proximity graph is formed using Delaunay triangulation [Anders, 2003] method. Then minimal spanning tree is derived to extract the group of characters of a particular word using similar path angle analysis between those components, orientation similarity validation and size similarity validation of those objects. In 2016, [Xu et al., 2016b] proposed a very similar approach of grouping based on the designed consistency constraints of the character color, size, spacing and direction.

2.4.5.2 Semantic layer extraction from graphical documents

In [Ramel and Vincent, 2003], different strategies for localization and recognition of graphical entities in line drawings are presented. Most systems include first a segmentation step of the document followed by a sequential extraction of the graphical entities. Some

other systems try to recognize symbols directly on the bitmap image using more or less sophisticated techniques. In this work, an intermediate structural representation of the document provides a precise description of all the shapes present in the initial image. Thereafter, this representation constitutes the main part of a shared resource that can be compared to structural signatures. This structural representation is used by different processes achieving the interpretation of the images. The actions (recognition of specific layers) done by these different specialists are scheduled in order to read and understand the content of the document. The knowledge that is provided by the shared representation is used instead of the bitmap image material to drive the interpretation process. In this system, the specialists are trying, during several cycles to interpret the documents in an **incremental way by interpreting the simplest parts first** and making the shared representation evolve until the total understanding of the document.

This method [Ramel et al., 2000] also defines a powerful structural representation able to describe the different types of content of a graphical document. For that, the binary image is **vectorized** to get a fine description of the content using vectors and quadrilateral primitives. A **structural graph** is then built from these primitives. The objective is to manage features relative to elementary objects so as to provide a description of the spatial relationships (inclusion, junction, intersection, etc.) that exist between the elements inside the images. This representation provides a global vision of the image. Thereafter, an analysis of the structural representation can be achieved by using a set of processes that deals with some specific task (linked to a **specific semantic layer**).The task of each of this process (the extraction of a specific layer) is simplified because it only solves basic problems first and it is made aware of the extracted context for solving more difficult problems later in an **incremental way**.

Figure 2.9 shows an example of semantical layers that can be extracted from cadastral maps using this method.

In this state of the art about Graphical document analysis, we should also mention the QGAR project done by Rendek et al. [Rendek et al., 2004]. Qgar project aims at providing stable and robust implementations of state-of-the-art methods and algorithms for graphics recognition within an intuitive and user-friendly environment. The resulting software system is open, so that our applications can be easily interfaced with other systems. The main algorithms available in this platform are dedicated to **layer extraction** inside graphical documents. We can mention : text-graphic separations, thin-thick separations, text extraction, vectorizations, image degradation, and basic symbol recognition techniques.

2.5 Summary and Synthesis

In this chapter, we have studied the proposed works related to different problems to solve when dealing with word spotting into graphical documents. According to all these previous works, we can draw the following conclusions.

FIGURE 2.9: Examples of extracted layers (a) Initial Image, (b) Vectorized Image, (c) Text, (d) Lines, and, (e) Hatched areas (Figure credit : [Ramel et al., 2000])

2.5.1 Regarding segmentation and ROI detection

When dealing with graphical documents, we have seen that segmentation free spotting systems would have better results than others.

We have also mention that visual features could be efficiently complemented with struc-

tural and lexical signatures. Then Text/Graphic separation problem have to be turned into the text segmentation and ROI detection method. There are many systems available to solve partially this task, mostly inspired from researches on line drawing understanding on one side and from text detection in natural images from another side. Systems dedicated to natural images are even capable to detect multi-oriented, multi-scaled text but these systems are not designed for binary images and then are inefficient to detect text which appear as touching with other graphical components for example. Another problem is coming from symbol or part of textures that do have high resemblance with text. In that case, identifying them separately from text is tough.

To counter with these situations, single binary decisions associated to discriminate between ROI and background of an images seems not a good solution to decide about the opportunity to compute signatures and index for a region or not. It could be preferable to design a more soft and incremental strategy based on more probabilistic approach during the computation of the index and able to use or to learn on-line useful information during the spotting process in order to generate first hypothesis that could be confirmed later with more adapted local information.

2.5.2 Regarding signature and index computation

As already mentioned several times, we would like to design a system that uses multiple types of features and signatures (visual, structural, lexical). Concerning visual features dedicated to spotting or CBIR, we can take benefit of the numerous previous works mainly based on bag of visual words and key point detection techniques.

Structural features can be very useful to represent graphical content and to discriminate between the different layers that constitute a graphical document. We have to try to incorporate such information in the indexing scheme of our system.

When coming to the point of lexical features, we could see that many methods are available for character segmentation and recognition in textual documents, but it is not the same when we need to process text by involving different rotation, scale invariant methods. Then, the problem is not yet completely solved and still need some improvements. Moreover, we have observed that recognition process are getting overfitted on the training model. During character recognition scenario the state-of-the-art features are getting over fitted to the training model. Resultantly, we still have difficulties to recognize the characters those are belong to scanned geographical document images. Furthermore, in existing literature rare works are available to deal with multi-oriented touching character segmentation. However, the existing methods are mostly learning based, where segmentation points are optimized dynamically based on the segmented parts recognition through classifier. Consequently, efficiency of character recognition approaches is also a bottleneck for the character segmentation problem.

During the analysis of methodologies to be applied to our problem, we have crawled through the state-of-the-art regarding all part of the problem. For character recognition, we have compared existing approaches in our dataset and finally used the out-performed feature from those approaches. The corresponding comparative results are given in Chapter 6.

2.5.3 Regarding system architecture and on-line retrieval step

A fair comparison is difficult to perform between learning based and learning free approaches as each of them have their own set of advantages and disadvantages. In the case of learning based approaches, ground truth (GT) is a mandatory requirement for training the system whereas in the case of learning free approaches, GT is not required and it could be more independent of data, language and scripts. As a counterpart, learning based approaches have most often high accuracy than learning free approaches and query-by-string based word spotting can be adapted in this category of techniques whereas learning free approaches is generally achieved by query-by-example based techniques. That is why we would like to propose a system that can provide a trade-off between these two category based on an incremental retrieval process that can include pseudo on-line learning capabilities. Still now, there is no architecture available that propose a multi-level indexing with learning free for pixel level analysis and learning based for lexical level analysis. In our proposal, we have combined both the learning free and learning based approach to achieve an effective retrieval system.

For that, during the retrieval step, a global approach can be usefully coupled with some local approaches when it is needed. A global approach corresponds to apply one global solution for all images or for a complete image without any adapted parameters or tuning with respect to the specificities of part of images. On the contrary, a incremental (local) approach could be dependent on heuristics or parameters tuned according to specific characteristics of an image or parts of an image. For retrieval, the architecture is allowing incremental seed based retrieval. This remarks tend to prove that the on-line retrieval step should then not be any more considered as a static process but should be defined as an incremental and adaptive process. It means **learning free or on-line learning (incremental) word spotting approaches** might be the only option available when dealing with heterogeneous graphical document.

Furthermore, the solution can be to compute, during the indexing step, a maximum of indices (visual, structural, lexical features/signatures) on potentially interesting regions (ROI), according to the characteristics of the ROI (probability to be part of a specific layer between text, graphic, symbol etc.). All these signatures constitute available information (indexes) that it can be useful or not to use during the retrieval step according to the type or the content of the query. We can even imagine that query by string as well as query by example could be possible at the same time.

Chapitre 3

Dataset and Annotation Tool

"Experts often possess more data than judgment"
- Colin Powell

Contents

Abstract

In this chapter, we present a brief discussion about the datasets, that we have used in our work. We have also generated ground truth information of the dataset for formal reporting of performance evaluation. Finally, we have presented the annotation tool that is used to generate our ground truth data.

3.1 Introduction

We need proper dataset to experiment and develop our approaches. In this regard, for every incremental stages of our system, there was lack of relevant datasets for experiments. Moreover, proper ground truth was also missing [Tombre et al., 2002] [Ahmed et al., 2011] [Li et al., 2000]. In [Karatzas et al., 2013], a dataset with ground truth of relevant context is given but the dataset is comprised of scene images, whereas we are working on document images. Also, [Pal et al., 2010] published a synthetic dataset for character segmentation using synthetic maps. In addition, Indian scripted dataset from different perspective is barely available for research purpose. Additionally, nowadays researchers are demanding right of access to the underlying data that supports the conclusions in the published materials. The publication of dataset encourages replication of findings in published works which also helps to compare existing solution with different solutions in the similar issues.

To develop a word-spotting system dedicated to graphical documents (maps), a sufficient amount of data is needed, which is a number of geographical document images with associated GT. Thereafter, to develop the system through incremental stages, we go through text and graphics separation followed by partial character recognition and ended up with a full word spotting system. Consequently, for the sake of performance evaluation we need, at every stage,ground truth quantifying text and graphic inside the images. Also, we need ground truth, identifying characters belonging to different documents and positioning words which belong to the maps. To build a multi-oriented and multi-script character recognition system, we need a huge amount of samples of every characters of each particular script. Here, we have taken, Bangla and English scripts.

In the above mentioned context, we have used few existing dataset and importantly, published multiple new datasets. The existing datasets are of English printed characters and numerals. Whereas the new datasets consist of the following data :

- ✍ Real English noisy printed numerals and characters collected from geographical document images,
- ✍ Bangla printed basic characters with multiple orientations, multiple scales, multiple fonts and multiple styles,
- ✍ Scanned geographical document images of Bangla scripts,
- ✍ Scanned geographical document images of English scripts,

Along with these datasets, we have generated ground truth data prepared with our personal tool. Our complete database is publicly available online [1] : for the use of other researchers. The data is obtained, mainly based on the documents available in selected school study books as part of subject, Geography.

1. `https://github.com/arundhati87/fluffy-pancake`

3.2 Dataset Description

3.2.1 Multi-Oriented characters

3.2.1.1 English text character

To evaluate and build a model for English character recognition, we have collected four different sets of English character data. These data are a complete set of English alphabet and numerals that consist of 26 uppercase letters, 26 lowercase letters and 10 numerals i.e. total 62 characters. Moreover, to develop a rotation and scale invariant character model, our dataset comprises multi-oriented and multi-scaled characters. One part of our dataset is based on scanned graphical documents. The data are scanned at 300 DPI in 256 different gray levels. We have manually labeled these data. The three other parts are synthetic data, collected from a publicly available dataset as mentioned in the work of Roy et al. [Roy et al., 2010], constructed by three different fonts i.e. Arial, Courier and Times New Roman (font size ranging from 12 to 30 units). The size of each of these three datasets is 5000 symbols. The total number of data is 17,000. Few sample characters are shown in Figure 3.2a. In our experiments both English uppercase and lowercase alpha-numeric characters are considered. Consequently, we must have total 62 unique characters (26 for uppercase alphabets, 26 for lowercase alphabets and 10 for numeral digits).

However, to generate a rotation invariant data, some of the characters like 'd' and 'p'; 'b' and 'q'; etc. are grouped together because of shape similarity due to unknown orientation. This ambiguity is managed as described later during online retrieval. Finally, we have considered 40 groups of unique character shapes. Table 3.1 above specified about these 40 groups. Each group contain the characters which are similar in shape under every rotation of the character.

Table 3.1: Complete English character groups

Character Groups			
0, O, o	A	g	N, z, Z
1, i, l, I	b, q	G	Q
2	B	h	r
3	c, C	H	R
4	P, p, d	j, J	s, S
5	D	k	T
6, 9	e	K	v, V
7, L	E	m	x, X
8	t, f	M, w, W	y
a	F	u, U, n	Y

3.2.1.2 Bangla (Indian script) text character

Basic characters of the modern Bangla script comprises of 11 vowels and 39 consonants [Chaudhuri and Pal, 1998]. In Figure 3.1, first, 11 characters are vowels and remaining are consonants. Other than these basic characters, in Bangla scripts, if a vowel follows a consonant, it takes a modified shape and is placed at the left, right, both left and right, or bottom of the consonant. These modified shapes are called modifiers. These modifiers add extra difficulty for segmentation or recognition of characters. Along with modifiers, there could lay around 300 Bangla characters. But with basic characters and very basic modifiers, we have defined only 52 different groups of characters in our experiment. For Bangla multi-oriented and multi-scaled text characters, we have generated the dataset using almost all open fonts available (more than 10). The size of the characters lies within the range starting from 12 to 72. Characters are randomly rotated between 1 to 180 degree using existing image rotation algorithm. We have also noticed that, characters lying in opposite way i.e. lying in more than 180 angle is a rare phenomenon.Therefore, we have not considered angles more than 180 degree during random rotations of characters. As we have collected English Text characters from publicly available dataset and from real maps, we have not done any modification in that. On the other hand, while developing Bangla character dataset we have added few more steps such as noise insertion. We have inserted noises synthetically in around 0.4 percent data for each different group of characters. The inserted noises are Salt and Pepper, Poison and Gaussian. Certain Bengali characters are collected from real maps. Total number of samples generated are 313,573. Total number of groups for Bangla character identification are considered as 52 and average 6000 characters per group.

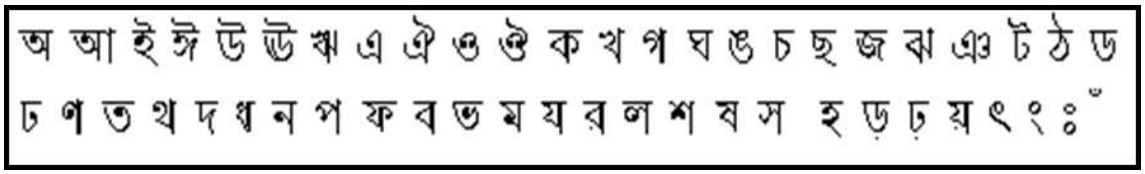


FIGURE 3.1: Basic characters of Bangla alphabet are shown. The first eleven characters are vowels and rest are consonants in the alphabet sets.

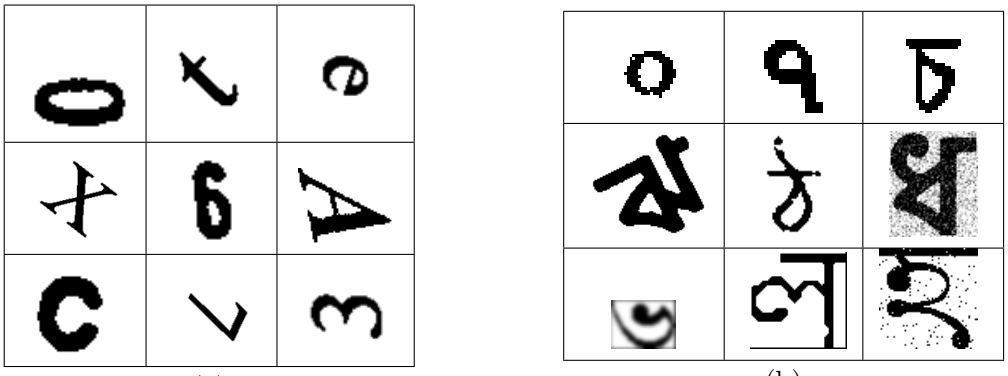
Few examples of English and Bengali characters are shown in Figure 3.2.

3.3 Map Dataset and Annotation

In the following section, we discuss on the process of data and corresponding ground truth preparation for spotting evaluation inside maps.

3.3.1 Map Documents (multi-script)

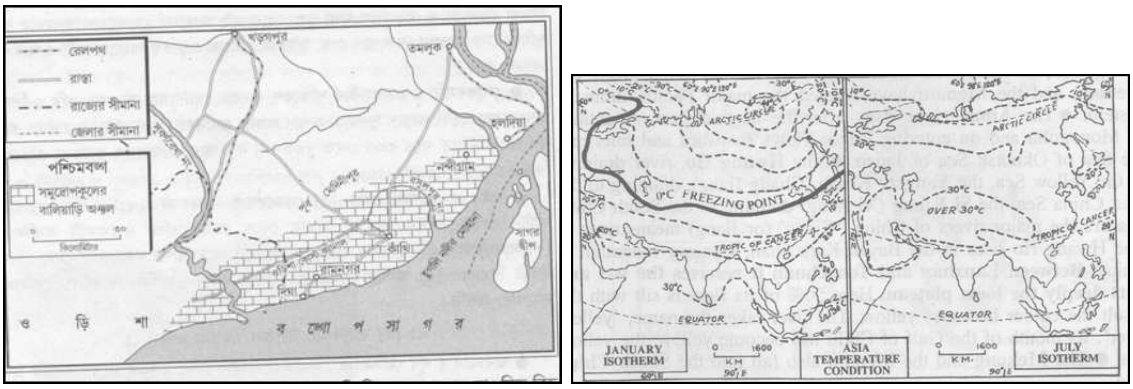
We have considered geography school books to get different scanned gray scale geographical maps. In recent advancement, documents are in color mode to make it more acceptable for viewing and understanding. But till date, document images mostly contain



(a) (b)

FIGURE 3.2: (a) Few samples of our English dataset, (b) Few samples of our Bangla dataset.

information in black and white or gray mode. Here, we have confined our study to grayscale document images. Total 108 English maps and 259 Bangla maps are scanned. Two samples of English and Bangla maps are shown below in Figure 3.3a and 3.3b respectively. All documents are digitized in gray tone at 300 dpi in a flatbed scanner. For the resolution, 300 dpi is chosen as it is a globally accepted resolution standard for maintaining an image quality with its details.



(a) (b)

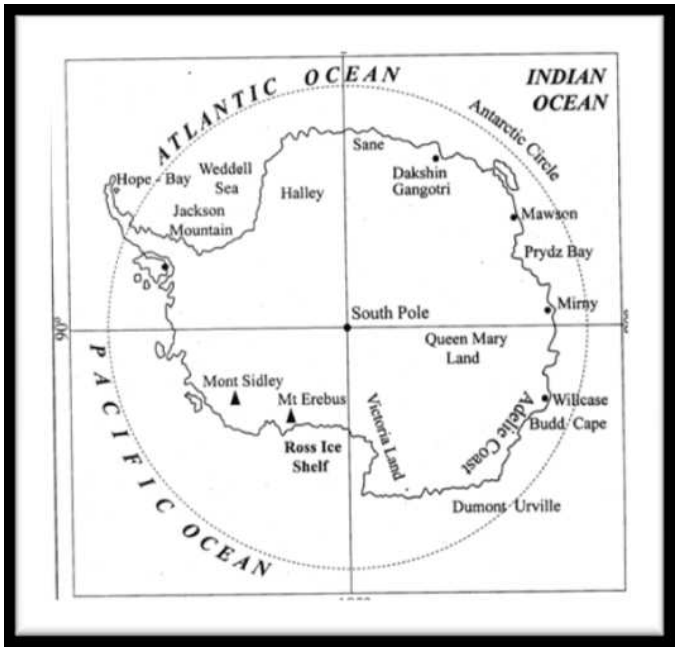
FIGURE 3.3: (a) One sample Bangla map, (b) One sample English map.

3.3.2 Annotation

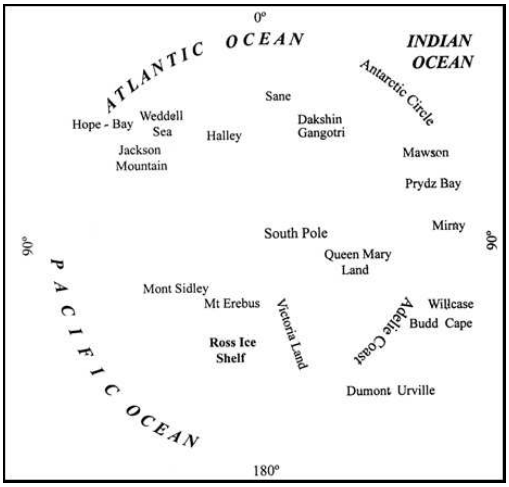
The ground truth dataset is based on 49 English maps and 47 Bangla maps. We have done our experiments incrementally. Text-graphics separation is the first step of analysis afterward which is progressed towards end solution i.e. word spotting. Ground truth for text-graphics separation is represented as binary images and in .xml files as well.

For text region detection evaluation, ground truth images contain binary images where black pixels are text pixels of corresponding maps. On the contrary, for graphic regions, ground truth images contain binary images where black pixels are remaining foreground

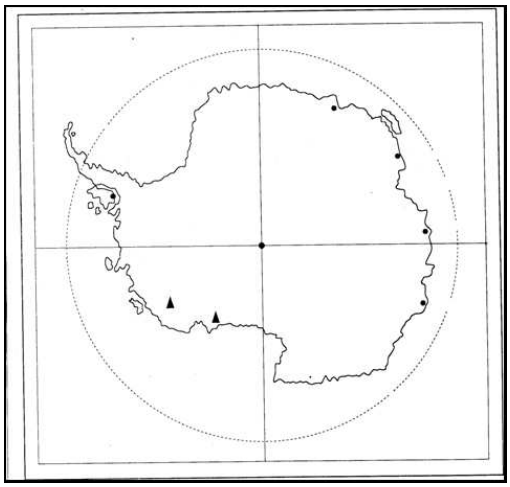
pixels of corresponding maps. These data are generated manually using "Paint" software on the binarized version of the original map. For generating text only map, we have manually erased all other non-text pixel from the map using the software. We have used "Paint" software as it is a simple erasing of data from an image. For generating graphic only map, we have subtracted text only map from the binarized version of original map. One such example with ground truth data is shown above Figure 3.4.



(a)



(b)



(c)

FIGURE 3.4: (a) One English map, (b) Corresponding text ground truth map, (c) Corresponding graphic ground truth map.

Furthermore, for text region and graphics region detection evaluation, along with graphical representation corresponding ground truth xmls are also generated. These xmls explicitly contain each text (graphic) pixel. Moreover, to represent same information with abstraction, another set of xmls are also generated. In spite of storing each pixel coordinates, these xmls contain position of each horizontal bounding box, which are bounding individual connected text (graphic) components horizontally. We have not maintained any text transcription here, as it is a raw data for text-graphics separation. Furthermore, for the evaluation of the next stages, we have generated another set of xmls with the detailed bounding box information and its corresponding text transcription.

In Figure 3.5, we have given a snippet of both of the xml files storing pixel based ground truth and bounding box based ground truth information. For storing pixel information in .xml files, the xmls contain one connected component is described by the tag "connectedObject". Its content is described by its pixel positions. Each pixel is denoted by the tag "taggedPixel", which is described by its position x and y. The result image and the ground truth image must have the same name. Moreover, for storing bounding box information in .xml files, we have maintained similar format as used in ICDAR 2003 reading competition [Lucas et al., 2003] [Wolf and Jolion, 2005] evaluation method. The xmls contain one rectangular bounding box is described by the tag "taggedRectangle". Its geometry is described by x, y, width and height. The result image and the ground truth image must have the same name. In the subsequent ICDAR competitions [Karatzas et al., 2013] [Karatzas et al., 2015], xmls are not used for ground truthing, rather they have used direct ground truth images.

Next, we have done ground truthing for character recognition and word spotting analysis. The ground truth is represented as using xml files. In the xml file, we have specified word content, position and type. For each word we have given a tag "object id" to provide an unique identifier to the word and mentioned its corresponding text transcription in a tag "VALUE".

Next,type of a word denotes how the word is lying or distributed in the map. We have provided two value to mention about the type of the word. We have described the details of ground truth information with the help of one example, given in Figure 3.6. In Figure 3.6 we have an example of one map, where we have marked four bounding boxes represents the ground truth information off all types of words.

The types are described as mentioned below :

1. If the complete word is a single connected component, we give "value1" is equal to "F" (Full)
   - ✍ The alignment of the word or connected component is horizontally or vertically straight as shown in Figure 3.7, here we have drawn a single bounding box and enter its corresponding text transcription. Consequently, the x, y position of top-left corner of the bounding box and the height and width of the box is generated and stored in the xml along with the text transcription. In xml, we give "value2" is equal to "S" (Straight) and provide tag "BoundingBox", under which tag "bbox" is described with its "x", "y", "height" and "width".
   - ✍ The alignment of the word or connected component is curved or straight in other than horizontal or vertical direction as shown in Figure 3.8, here we have

```
<?xml version="1.0" encoding="UTF-8"?>
<tagset>
        <image>
                <imageName>001_text</imageName>
                <taggedPixels>
                        <connectedObject>
                                <taggedPixel x="3" y="330" />
                                <taggedPixel x="4" y="330" />
                        </connectedObject>
                </taggedRectangles>
        </image>
</tagset>
```

(a)

```
<?xml version="1.0" encoding="UTF-8"?>
<tagset>
        <image>
                <imageName>001_text</imageName>
                <taggedRectangles>
                        <taggedRectangle x="1" y="325" width="1" height="1" />
                        <taggedRectangle x="1" y="329" width="1" height="1" />
                </taggedRectangles>
        </image>
</tagset>
```

(b)

FIGURE 3.5: Code snippet of (a) pixel level ground truth xml, (b) bounding box level ground truth xml

drawn a single bounding box and enter its corresponding text transcription. Consequently, we have generated its minimal enclosing bounding polygon (no of sides of the polygon is decided through an automated procedure). The generation of the polygon is achieved by water reservoir method [Pal et al., 2003] from the enclosed bounding box. Resultantly, x, y positions of polygon points are stored in the xml along with its text transcription. In xml, we give "value2" is equal to "C" (Curved) and provide tag "BoundingPts", under which tag "bboxvalue" is given to give an unique identifier to each polygon points, mentioned with its "x" and "y".

2. The whole word is comprised of multiple segments or multiple connected components. For this kind of word, in xml, we give "value1" is equal to "P" (Partial) and "value2" is equal to "N" (None). Segments are set of individual characters or set of multiple connected characters as shown in Figure 3.9 and 3.10 respectively. Here we have drawn a single bounding box covering the complete word and enter its corresponding text transcription. Next, we have marked bounding boxes separately covering all of

the individual segments and enter their text transcription. Consequently, individual x, y position of top-left corner, height and width of all individual bounding boxes are generated and stored in the xml along with every individual text transcription. If, a segment is a single character then in xml, we provide tag "BoundingCharPts", under which segment is denoted by an unique identifier in tag "compValue" and its corresponding text transcription is given in tag "char". Also, its bounding information is given in "bbox" with its "height", "width", "x" and "y". Otherwise, if a segment is a set of connected characters then we provide tag "BoundingBox", under which segment is denoted by an unique identifier in tag "compValue" and its corresponding text transcription is given in tag "char". Also, its bounding information is given in "bbox" with its "height", "width", "x" and "y".

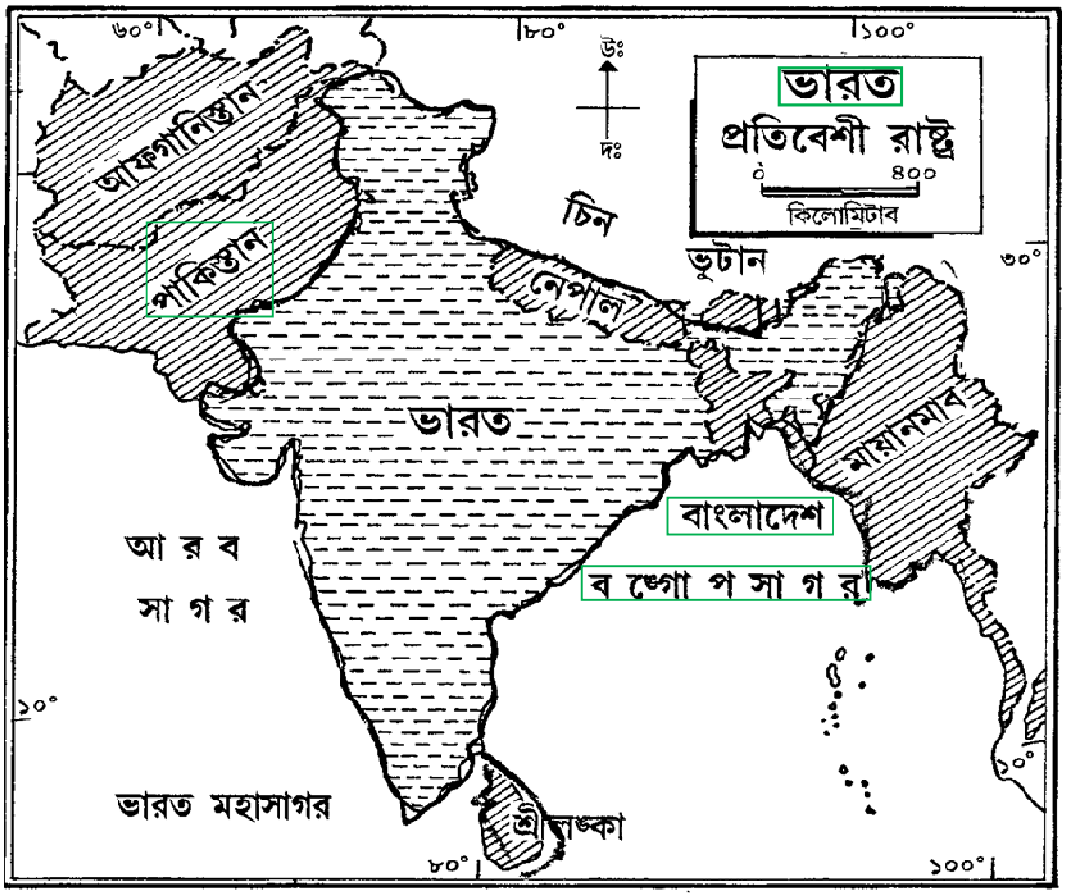
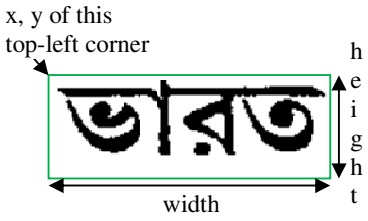


FIGURE 3.6: Sample map for ground truthing for word spotting evaluation. Four sample sets of bounding boxes shown in green color for four types of words for ground truhing. All the arrow marks are pointing towards the values or positions which are stored in xml for performance evaluation.

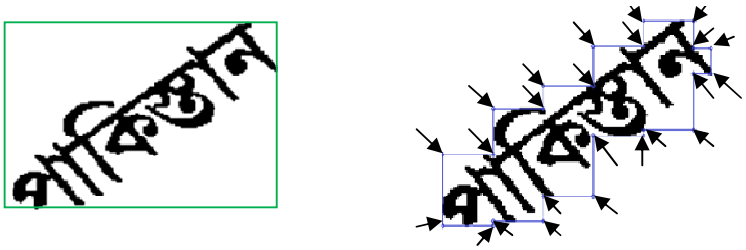


(a)

```
<object id="6" VALUE="ভারত">
        <attribute name="Connected" value1="F" value2="S"/>
        <attribute name="BoundingBox"/>
        <bboxheight="42" width="127" x="487" y="512"/>
</object>
```

(b)

FIGURE 3.7: (a) One horizontally straight connected word referenced from Figure 3.6, (b) Corresponding ground truth code snippet to store its text content and position. All the arrow marks are pointing towards the values or positions which are stored in xml for performance evaluation.

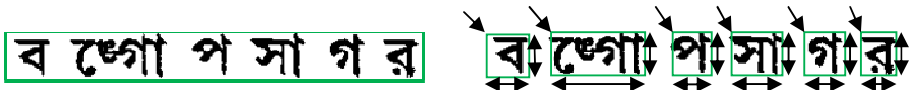


(a) (b)

```
<object id="2" VALUE="পাকিস্তান">
        <attribute name='Connected' value1='F' value2='C'/>
        <attribute name='BoundingPts'/>
        <bboxvalue='1' x='176' y='296'/>
        <bboxvalue='2' x='177' y='361'/>
        <bboxvalue='3' x='203' y='361'/>
        <bboxvalue='4' x='203' y='339'/>
        <bboxvalue='5' x='229' y='339'/>
        <bboxvalue='6' x='229' y='328'/>
        <bboxvalue='7' x='255' y='328'/>
        <bboxvalue='8' x='255' y='309'/>
        <bboxvalue='9' x='281' y='309'/>
        <bboxvalue='10' x='281' y='297'/>
        <bboxvalue='11' x='307' y='297'/>
        <bboxvalue='12' x='307' y='310'/>
        <bboxvalue='13' x='316' y='310'/>
        <bboxvalue='14' x='316' y='322'/>
        <bboxvalue='15' x='307' y='322'/>
        <bboxvalue='16' x='307' y='349'/>
        <bboxvalue='17' x='281' y='349'/>
        <bboxvalue='18' x='281' y='352'/>
        <bboxvalue='19' x='255' y='352'/>
        <bboxvalue='20' x='255' y='381'/>
        <bboxvalue='21' x='229' y='381'/>
        <bboxvalue='22' x='229' y='393'/>
        <bboxvalue='23' x='203' y='393'/>
        <bboxvalue='24' x='203' y='395'/>
</object>
```

(c)

FIGURE 3.8: (a) One curved connected word referenced from Figure 3.6, (b) Corresponding minimal enclosed bounding polygon generated, (c) Corresponding ground truth code snippet to store its text content and position. All the arrow marks are pointing towards the values or positions which are stored in xml for performance evaluation.

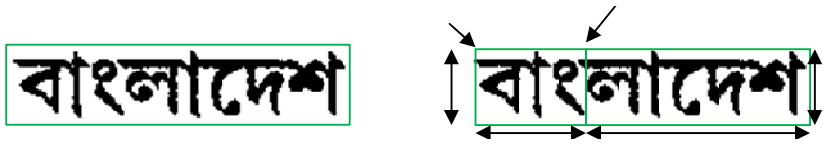


(a) (b)

```
<object id="13" VALUE="বঙ্গোপসাগর">
        <attribute name="Connected" value1="P" value2="N"/>
        <attribute name="BoundingCharPts" compValue="1" char="ব"/>
        <bboxheight="35" width="33" x="701" y="715"/>
        <attribute name="BoundingCharPts" compValue="2" char="ঙ্গো"/>
        <bboxheight="35" width="82" x="747" y="715"/>
        <attribute name="BoundingCharPts" compValue="3" char="প"/>
        <bboxheight="32" width="33" x="849" y="717"/>
        <attribute name="BoundingCharPts" compValue="4" char="সা "/>
        <bboxheight="36" width="44" x="898" y="717"/>
<attribute name="BoundingCharPts" compValue="5" char="গ"/>
        <bboxheight="38" width="29" x="962" y="715"/>
        <attribute name="BoundingCharPts" compValue="6" char="র"/>
        <bboxheight="29" width="27" x="1004" y="719"/>
</object>
```

(c)

FIGURE 3.9: (a) One segmented word referenced from Figure 3.6, (b) Corresponding individual bounding rectangle for individual characters, (c) Corresponding ground truth code snippet to store their text contents and positions. All the arrow marks are pointing towards the values or positions which are stored in xml for performance evaluation.



(a) (b)

```
<object id="7" VALUE="বাংলাদেশ">
        <attribute name="Connected" value1="P" value2="N"/>
        <attribute name="BoundingBox" compValue="1" char="বাং"/>
        <bboxheight="37" width="60" x="815" y="630"/>
        <attribute name="BoundingBox" compValue="2" char="লাদেশ"/>
        <bboxheight="36" width="121" x="870" y="630"/>
</object>
```

(c)

FIGURE 3.10: (a) Another segmented word referenced from Figure 3.6, (b) Corresponding minimal enclosing bounding rectangles for individual components, (c) Corresponding ground truth code snippet to store their text contents and positions. All the arrow marks are pointing towards the values or positions which are stored in xml for performance evaluation.

Annotation Tool

To generate the ground truth information of character recognition and word spotting, we have developed a suitable annotation tool. The snapshot of this annotation tool is given in Figure 3.11. Using this tool, we could select an image and choose the underlying writing script of the image either as English or Bangla.

Next, we have to select two diagonal points of a word for automatic drawing of the bounding box of the word using the two diagonal points. The diagonal points must be the topmost-leftmost and bottommost-rightmost corners of the bounding box. Further, we choose whether the bounded portion is a :

- ✍ curve or straight in other direction than horizontal (vertical) direction, or
- ✍ horizontally (vertically) straight component, or
- ✍ comprises of multiple components.

If the text is a curve or straight in other direction than horizontal (vertical) direction component, then from the bounding box information, the tool would automatically generate the minimal bounding polygon around the word and store all the vertices of the polygon. For all other aligned words, the tool simply stores the corresponding minimal bounding rectangular box information.

Next, we have to enter the corresponding text transcription of the word in a text field. Finally, using the tool the ground truth xml would be generated with all these detailed information.

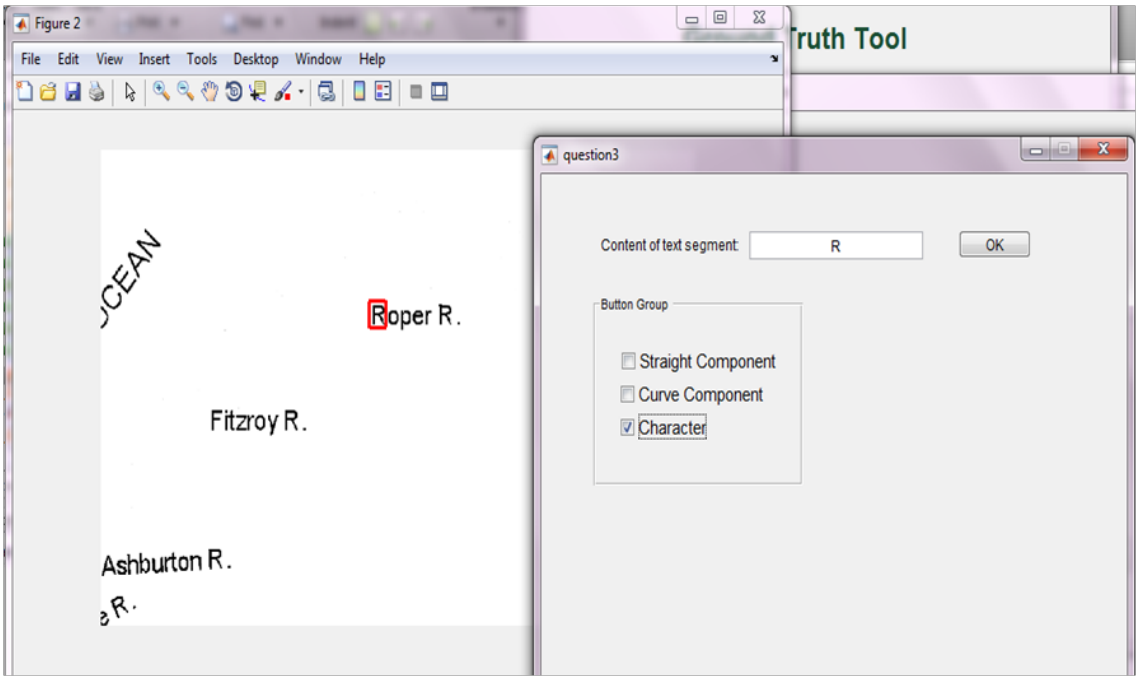


FIGURE 3.11: Screenshot of the graphical user interface of our annotation tool

3.4 Summary

In this chapter, we have mentioned about all the dataset, those are required for testing the feasibility and performances of our work. We have tried to bridge the gap of availability of public datasets regarding our line of work. During the data generation, we have concentrated only on geographical scanned documents. Despite of being a narrow field, geographical documents make the problem multi-dimensional and due to availability of our dataset publicly along with the annotation tool, future researchers can exploit it further to maximize its expediency.

Chapitre 4

From Text-Graphics Separation to a Multi-level indexing

"Research is to see what everybody else has seen, and to think what nobody else has thought."
- Albert Szent-Gyorgyi

Contents

Abstract

In this chapter, we have thoroughly described our proposed approaches for the indexing stage with some recalls of existing theories used inside this framework. After a presentation of the global workflow of the proposed method each step of the indexing is described in details. Each stage corresponds to a level of analysis included inside our indexing scheme. A final section explains how the information and signatures extracted at each level are merged together inside a final indexed data structure that will be used during the retrieval stage.

4.1 Introduction

To deal with the difficulties associated to Text/Graphics separation that we have mentioned in the previous chapters, we propose a new workflow including several analysis and signature extraction process at multiple levels instead of selecting specific and definitive ROIs (based on a segmentation step or not) on which several specific signatures (visual or more semantical) could be computed to generate the final index files.

The Figure 4.1 shows the global workflow of the proposed method. This Figure includes the retrieval stage that will not be discussed in this chapter but in Chapter 5.

Focusing on the indexing stage, we can see that all the input images (to index) will be analyzed at three different levels (pixel level, structural level and lexical level). This **multiple level analysis** is performed gradually so that next stage can take benefit of the previous one. Anyway, it is noticeable that information extracted at each stage will be used to generate the final index files (data structure). In this sense, the Figure shows that classical ROI selection and characterization are replaced by probability maps resulting from the two first levels of analysis.

What we call **probability map** is a data structure used to store probability values associated to pixels and corresponding to the plausibility to be part of a specific class of content inside a graphical document like for example the *Text class* or the *Graphics Class*. Even if, in this work, we have reduced our study on probability maps associated to the *Text* and *Graphic* layers, it is noticeable that this concept can be "easily" expanded to more semantical classes (like *River Class*, *Building Class* etc.). In most of the cases, the probability values will come from the output of a classifier in charge of deciding if a pixel or an area of the initial image is part of a specific class or not according to specific features extracted beforehand.

To do their jobs, these classifiers need **training data** to know how to make difference between the classes they need to recognize. As our actual system deals only with word spotting in graphical document (for the moment), only character level training data are needed. It means labeled character images corresponding to the fonts/scripts present in the images to index have to be provided for the learning. As shown in Figure 4.1 this training dataset will be used into two different ways :

- First, it will be used for individual character recognition during the **lexical indexation** step,
- Second, it will be used during the incremental retrieval step to localize more difficult or ambiguous parts of query word based on **more visual (low level) information** using SIFT key point matching technique. That is why SIFT key points have to be extracted beforehand (during an off-line learning step) on all the images of the training dataset (see Figure 4.1).

Of course, the training dataset has to be adapted to the images to index. Latin and Indian (Bengla, ...) characters have to be selected according the scripts used inside the images to index. The details of each of these training dataset for character recognition regarding English and Bengali script is discussed in Section 3.2.

In contrary to the two first levels, the lexical level indexing is providing a more semantical signature as an output of the indexing stage. As described in Section 4.4, the signature

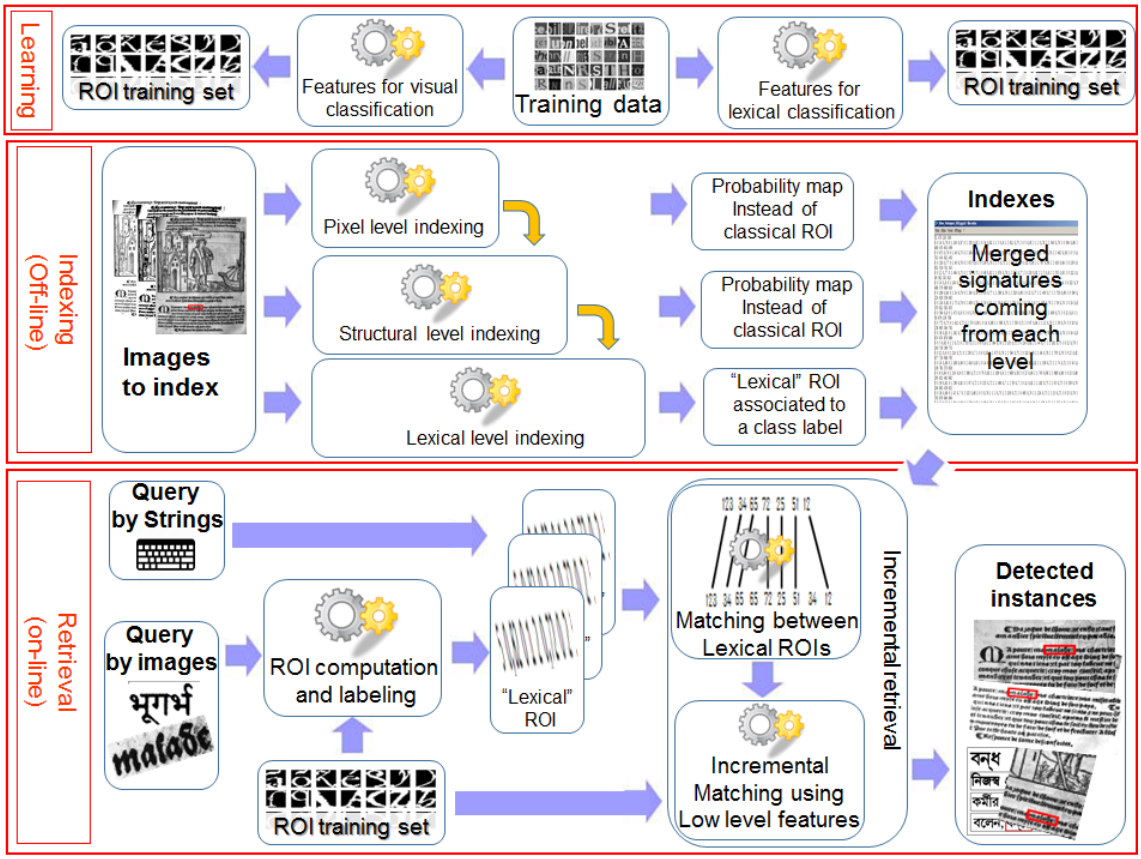


FIGURE 4.1: Global workflow of the proposed system. At the top, the learning stage; in the middle, the multi-level indexing stage and the incremental retrieval step at the bottom (discussed in Chapter 5)

associated to this level correspond to the list of the labels of the recognized individual characters that could have been extracted and recognized without too much ambiguity inside the graphical document (associated to extra information like position, size, orientation etc.).

4.2 Pixel Level Indexing

4.2.1 Introduction

Before describing our proposed approaches for the pixel-level indexing step, we have to explain few terminologies.

Images can be analyzed into two domains. One is the spatial domain, which analyzes the image in the normal image space, where we process or calculate based upon direct pixel intensities of image matrix. In addition, frequency domain refers to the frequency of the image intensity value changes with respect to the change in pixel locations. Image features with high spatial frequency (such as edges) are those that change greatly in intensity over

short image distances. On the other hand, frequency domain is a space in which each image value at specific image position represents the change of intensity values in spatial domain over a specific distance related to that position. For example, higher frequency components correspond to edges in an image, whereas, lower frequency components in an image correspond to smooth regions.

We would require the use of filters in spatial domain or frequency domain for image processing for specific requirement. A filter is a mask used to call attention to certain features of an image. For example, we can smooth, sharpen, and enhance edge by using image filters. Filtering is a neighborhood operation, in which the value of any given pixel in the output image is determined by applying some algorithm to the values of the pixels in the neighborhood of the corresponding input pixel. filtering of an image is accomplished through an operation called convolution. A convolution is done by multiplying a pixel's and its neighboring pixels' intensity value by a matrix, named as kernel. If we let $\mathtt{f}$ be the image we want to filter, $\mathtt{g}$ the corresponding output image, and let $\mathtt{h}$ be the convolution kernel, we have :

$$\mathtt{g(x,y)} = \sum_{\mathtt{i}=-\mathtt{w}}^{\mathtt{w}} \sum_{\mathtt{j}=-\mathtt{w}}^{\mathtt{w}} \mathtt{h(i,j)f(x-i,y-j)} \tag{4.1}$$

where the size of the kernel is $(2\mathtt{w}+1) \times (2\mathtt{w}+1)$. The size of a kernel is arbitrary but 3×3 is often used. In Frequency domain, the image is Fourier transformed, multiplied with the filter function and then re-transformed into the spatial domain. Attenuating high frequencies results in a smoother image in the spatial domain, attenuating low frequencies enhances the edges. We can replace filtering in frequency domain by using a simple kernel convolution in spatial domain. Filters can be basically divided into three types as follows : Low pass filtering, otherwise known as "smoothing", is employed to remove high frequency from a digital image. Generally, noise removal is done by low pass filter. High pass filtering, is to remove low frequency from a digital image. Generally, edge detection is done by high pass filter. A band pass filter attenuates very low and very high frequencies, but retains a middle range band of frequencies. Band pass filtering can be used to enhance edges (suppressing low frequencies) while reducing the noise at the same time (attenuating high frequencies).

Now, coming to our point of investigation that is, for the Pixel level, **Text-Graphics separation**, we have already mentioned several properties of textual contents in Section 1.3 during the discussion over difficulties in graphical documents. We face all those difficulties during the first part of the proposed process

In a first step (Filter based Layer Analysis - section 4.2.2), we have used band pass filter to separate the *Text layer* from the others. The filter is convoluted with the grayscale intensity values on the image. The size of the filter depends upon the standard deviationof the filter. A larger standard deviation leads to larger and spreader convolution. This particular methodology has already been applied [Jain and Farrokhnia, 1991] to solve the problem of texture segmentation using $2-D$ Gabor Filter, a band pass filter. We have first tried to replicate the same solution to solve our problem of Text layer separation.

As, Gabor filter is having several parameter dependencies, in order to get rid of that, we have also tested to use a lesser parameter dependent similar functional filter : Laplacian of Gaussian (LoG) filter, which is a derivative filter. Likewise, the application of Gabor filter, we have introduced a rotation invariant form of LoG filter to treat with multi-orientation of the texts in those graphical documents. According to their convoluted data point values (Gabor or LoG features),the pixels are clustered to bring text pixels into a single cluster. K-means algorithms has been used for this clustering of pixels to generate the $\mathbb{K}$ different layers.

In a second step (Self-Learning based Layer Analysis - section 4.2.3), we have proposed an incremental method that uses information obtained from Text areas that can be easily extracted inside a specific image to learn how to extract more difficult areas. Moreover, with the concept of probability map, we can attach a probabilistic measurement to the possible textual or graphical elements inside the indexed images.

4.2.2 Filter based layer analysis

4.2.2.1 Band pass filter

We propose a system using Gabor Filter to compute Pixel based features. These features can be used to separate text from graphics from grayscale graphical document images. Gabor filter [Gabor, 1946], is a linear band pass filter used for edge detection. See Appendix A.1.1 for detailed discussion. Several parameters are used to design a Gabor filter. Those are γ (aspect ratio), λ (wavelength), $\mathtt{b}$ (bandwidth), θ (orientation), Ψ (phase offset), σ (standard deviation). The setting for all these parameters can be done mentioned as follow : γ (aspect ratio) - aspect ratio defines the elliptical shape of the filter. The ratio is ratio of size of the filter towards $\mathtt{x}$ axis direction and $\mathtt{y}$ axis direction. Ψ(phase offset) $= 0$; θ(orientation) - It would give best result when orientation of the filter will be same as text orientations. In our experiment, θ taken as 0°, 30°, 60°, 90°, 120° and 150°. λ is taken [Zhang et al., 2002] as mentioned below,

$$\begin{aligned}
&\lambda = 1/\mathbb{F};\\
&\mathbb{F} = [(0.25 - \mathbb{J}(0.25 + \mathbb{J}];\\
&\mathbb{J} = 2^{(\mathbb{N}-.5)}\mathbb{M};\\
&\mathbb{N} = [0, 1, 2... \log_2 \frac{M}{8}], (\mathbb{M} = \text{no of columns of Input Image})
\end{aligned} \tag{4.2}$$

For example, for an image with 256 columns [Seo, 2006], a total of 60 filters can be used with each of 6 orientations and 10 frequencies ($\frac{1}{\lambda}$).

Alternatively, we have also used Laplacian of Gaussian (LoG) Filter to compute the pixel based feature. The benefit of using LoG filter is that it require lesser parameters for the same text segmentation problem. Similar to Gabor filter, LoG Filter also amplify the difference between grayscale intensities of an image. LoG, or Laplacian of Gaussian, is a blob detector. First, it blurs the image to ensure it is not too affected by noise, and then apply a Laplacian filter to highlight regions of quick intensity variation.

Laplacian filter [R. Fisher et al., 2003] is a $2-D$ isotropic measure of the $2nd$ spatial derivative of an image. This isotropic operator works equally well in all directions in an image, with no particular sensitivity or biasness towards any set of directions. The $2nd$ derivative finds rapid intensity change mainly to emphasize the edges in an image. Because these kernels are approximating a second order derivative measurement, they are very sensitive to noise. A picturization of LOG filter is shown in below Figure 4.2.
The equation of LoG filter is,

$$LoG(\mathtt{x},\mathtt{y}) = -\frac{1}{\pi\sigma^4}[1-\frac{\mathtt{x}^2+\mathtt{y}^2}{2\sigma^2}]\exp(-\frac{\mathtt{x}^2+\mathtt{y}^2}{2\sigma^2}) \tag{4.3}$$

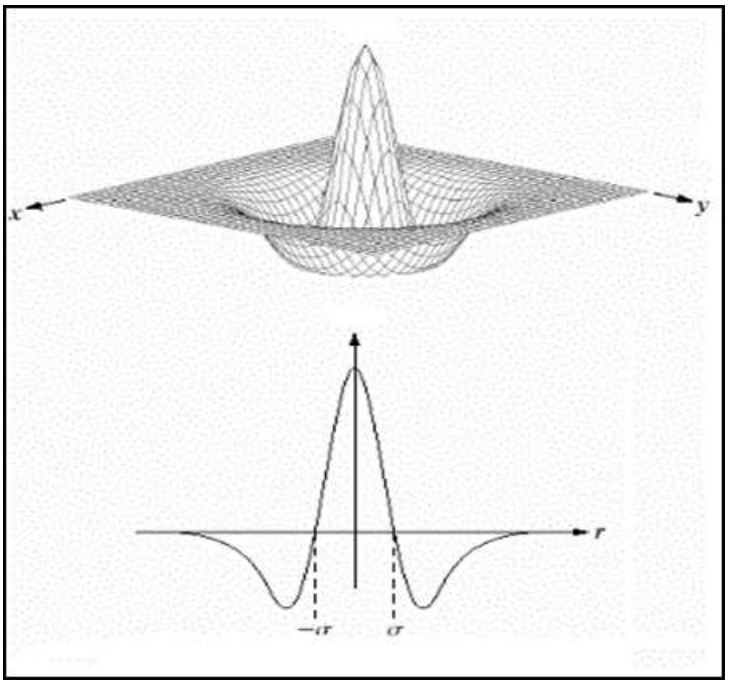


FIGURE 4.2: Picturization of LoG filter. (Figure credit : [Weis, 2009])

The operator response is strongly dependent on the relationship between the size of the blob structures in the image domain and the size of the Gaussian kernel used for pre-smoothing. In order to automatically capture blobs of different (unknown) size in the image domain, a multi-scale approach would be better. Therefore, we have chosen multiple sigma for the LOG operator. We have a set of sigma values as used in Gabor filter. All the parameters range contains same value as used in Gabor filter. Likewise, in LoG filter, for handling multi-oriented text, we have introduced aspect ratio factor i.e. elliptical shape of the filter and multi-oriented LoG filters.

After introducing aspect ratio and multiple orientation, the modified LOG equation could be written as,

$$LoG(\mathtt{x},\mathtt{y}) = -\frac{1}{\pi\sigma^4}[1 - \frac{\mathtt{x}'^2 + \gamma^2\mathtt{y}'^2}{2\sigma^2}]\exp(-\frac{\mathtt{x}'^2 + \gamma^2\mathtt{y}'^2}{2\sigma^2}) \tag{4.4}$$

where, $\mathtt{x}' = \mathtt{x}\cos\theta + y\sin\theta, \mathtt{y}' = \mathtt{y}\cos\theta - \mathtt{x}\sin\theta$

4.2.2.2 Pixel feature computation using these filters

Being a band pass filter or, derivative filter as well, Gabor filter or, LoG highlights specific frequencies of the images whereas other frequencies get diluted. The convoluted filtered image is smoothed using Gaussian Filter to blur the effect of noises in the image, that is normalized into range of 0 to 1. In $2-D$, an isotropic (i.e. circularly symmetric) Gaussian has the form as shown below, where σ is the standard deviation of the distribution :

$$\mathbb{G}(\mathtt{x},\mathtt{y}) = \frac{1}{2\pi\sigma^2}\exp(\frac{-\mathtt{x}^2 + \mathtt{y}^2}{2\sigma^2}) \tag{4.5}$$

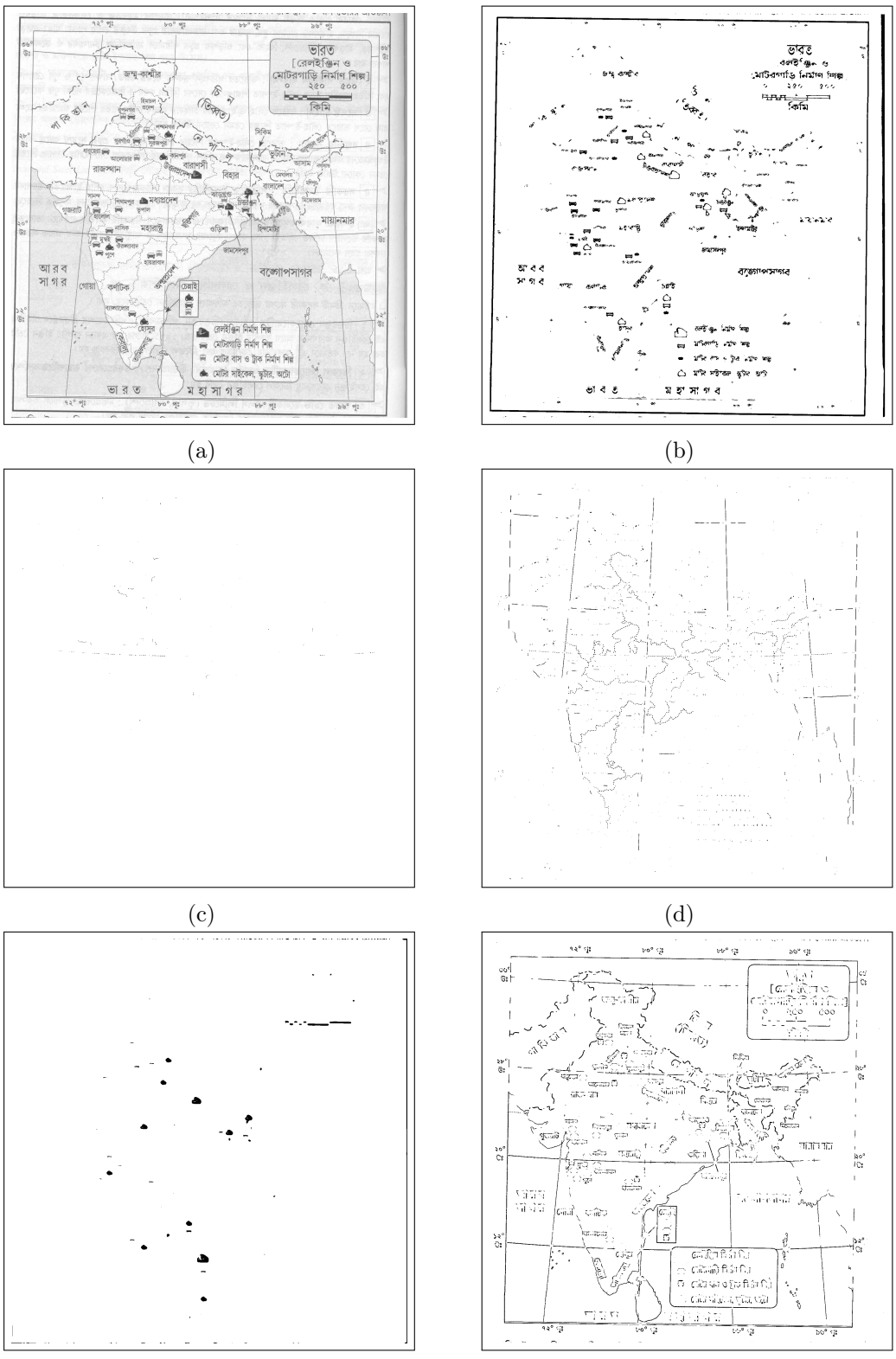
The multiple convolutions of the original image with the variants of the filter generate multiple data points for each pixel position. The number of data points for each pixel positions would be the (number of orientations x number of standard deviations) that are used to generate the filter. Those data points are used as a vector of features representing a pixel of the original image.

The obtained vectors can be grouped inside clusters having frequencies in a close proximity. Resultantly, we are able to fetch text pixels in a cluster. To improve the quality of the clustering, proximity information between pixels can be added inside the features vectors as proposed in [Seo, 2006].

4.2.2.3 Clustering

We have experimented classical K-means and K-means++ [Arthur and Vassilvitskii, 2007] to group pixels with similar properties. K-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. K-means++ proposes a method for choosing the initial centroid values (or "seeds") for the k-means clustering in place of choosing random initial centroid value. After experimenting with a set of 40 maps, collected by scanning maps from school books of English (Roman) and Bengali scripts, we have seen that K value as five or four, work efficiently to obtain pertinent clusters according to our goal (Text / Graphics separation).

Figure 4.3, Figure 4.4, show two set of examples, using Gabor filter of five clusters and four clusters K-means respectively, corresponding to one map.

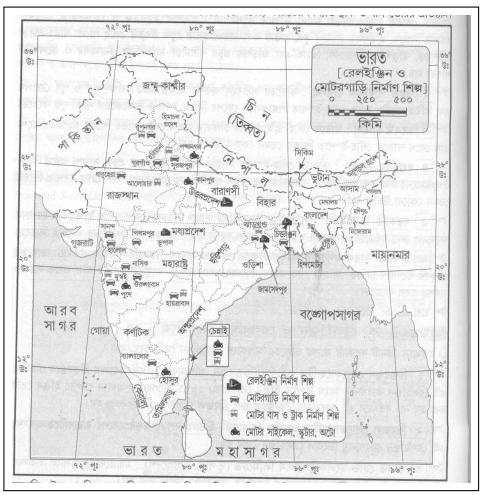


(a) (b)

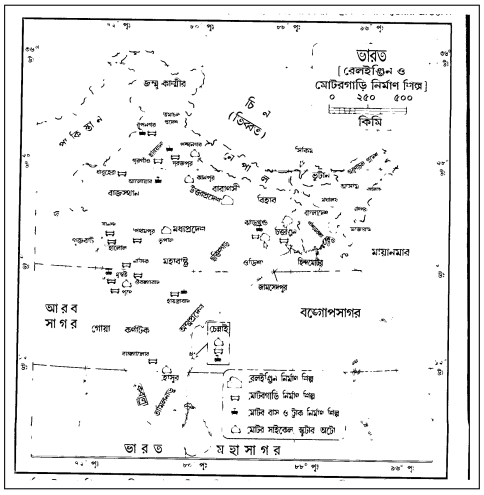
(c) (d)

(e) (f)

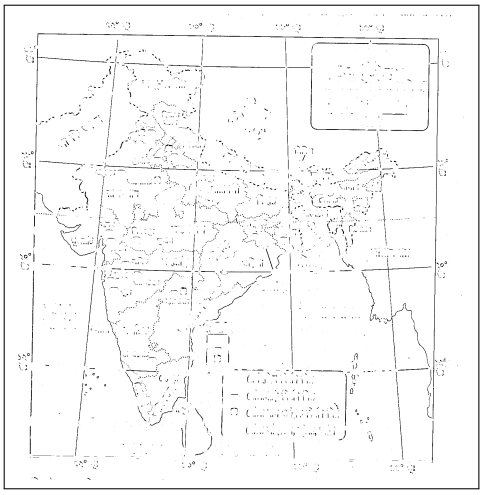
FIGURE 4.3: One map and its corresponding five clustered outputs using K-means



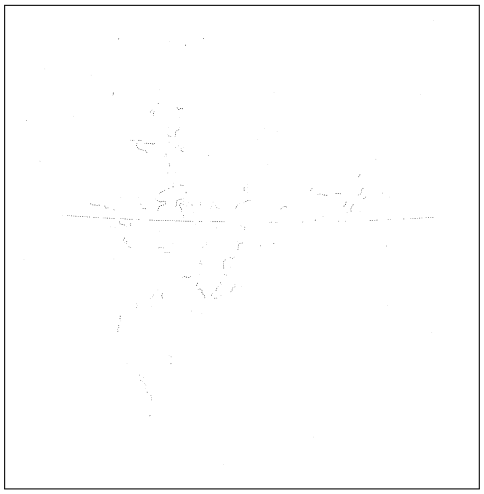
(a)



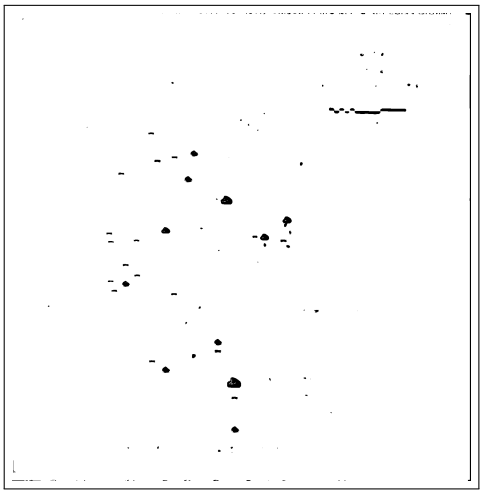
(b)



(c)



(d)



(e)

FIGURE 4.4: One map and its corresponding four clustered outputs using K-means

The two requested clusters (the one corresponding to Text layer and the one corresponding to Graphics) have to be identified from all the four/five clusters. Heuristics can easily be set to identify these two clusters based on the number of elements (pixels) inside the clusters and the size and shape of the connected components generated from these pixels. For example, from images in Figure 4.3b and 4.3f can be easily selected using these kind of criteria. The cluster corresponding to graphical objects, distinctively, contains few components which are much longer in length compared to the objects of cluster that contain mostly textual objects. Similarly, in Figure 4.4b is the corresponding to text cluster of map image given in Figure 4.4a.

Block diagram of this text-graphics separation process is shown below in Figure 4.5.

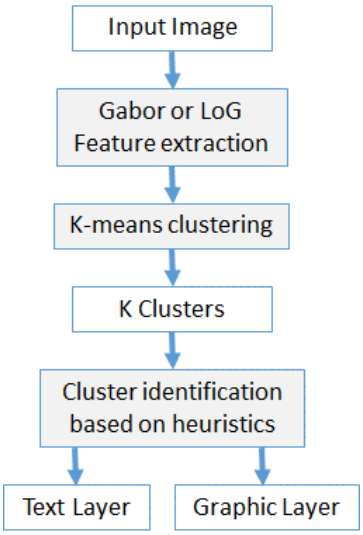


FIGURE 4.5: Block diagram of Filter based Layer Analysis step

Again, we have applied K-means clustering algorithm to the connected components using a specific set of features. The features correspond to their area, equivalent diameter size, extent and length of major axis. Area of the component is the count of number of pixels under a connected component region, whereas, equivalent diameter size gives diameter of the circle with the same region, extent is the ratio of area of connected component region divided by area of complete bounding box of the same component, and finally, length of the major axis is the length of major axis of the ellipse that has same normalized second central moments as the region of the corresponding component. It is noticed that a geographical document images generally contain two set of font sizes. One is to label the complete document ; another is to label detailed places. Consequently, here number of clusters is taken as three, assuming third group belongs to non-text objects. As the base image is a text objected cluster, so, if we group that image.

components into three groups, non-text group would contain rare number of objects, whereas text groups would contain many number of objects. Consequently, number of components in every group is counted. If any group contain many (taken, minimum 20 in our experimentation) components, we have assumed those groups comprises of textual components leaving the longer and unique graphical components in other rare groups which

has fewer components belong to them. From the groups contain minimum 20 components, maximum size of component belong to them is determined. As shown in Figure 4.6, three clusters comprises of different components denoted by three different colors. Among them only one cluster (deep blue colored) is having many components compared to other clusters. Consequently, that particular cluster would be taken as group of text components and other groups are removed from the image.

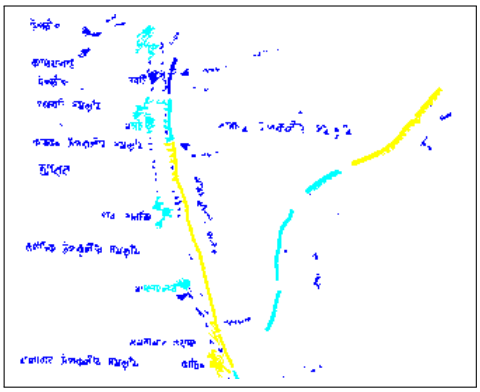
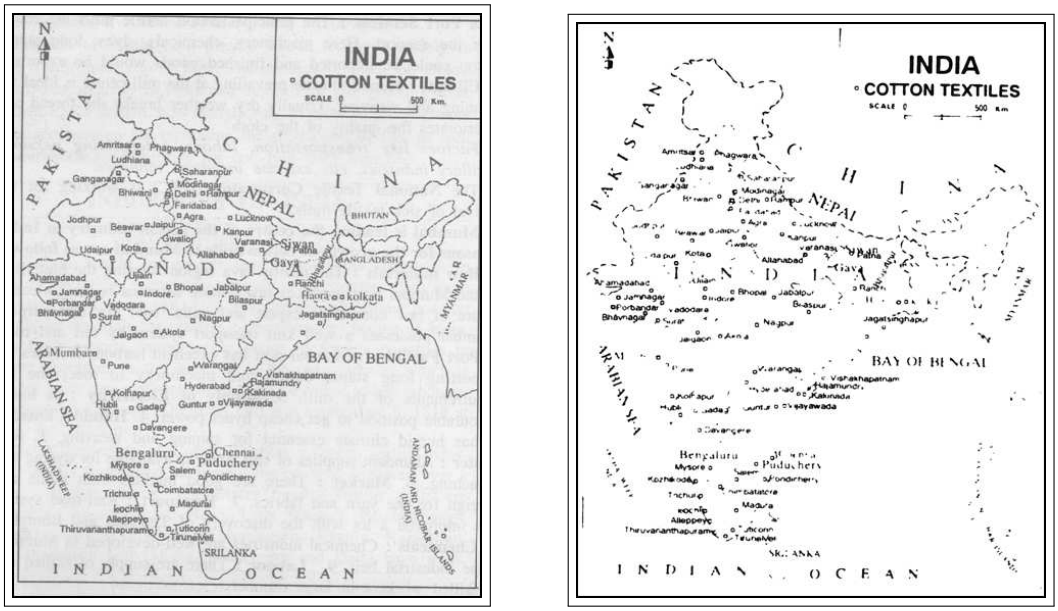


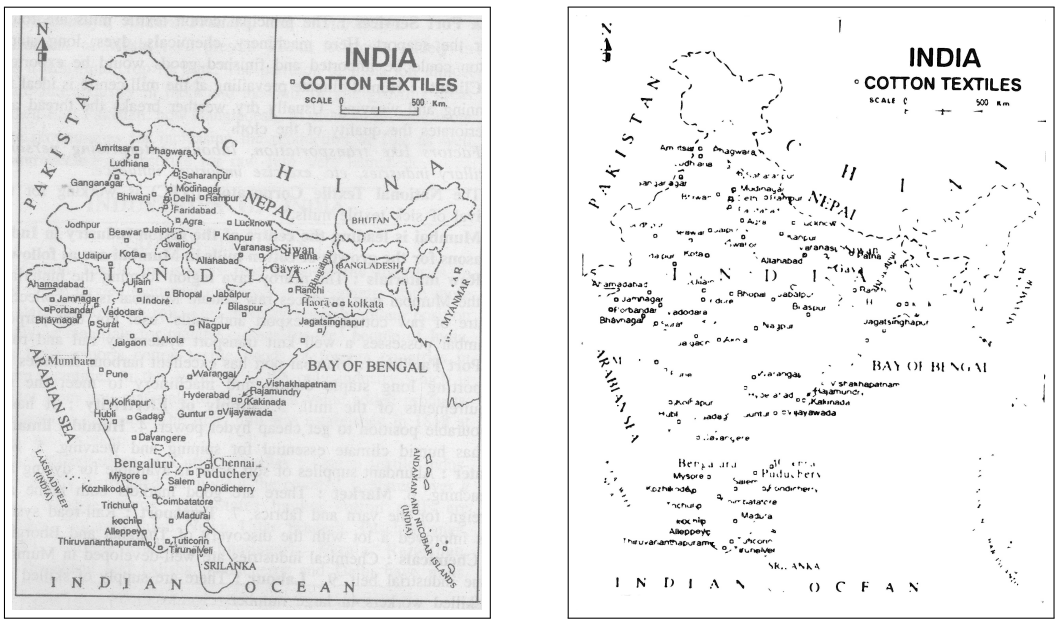
FIGURE 4.6: Part of a map with component clustering (shown by different colors)

Using above explained approach, one example map with corresponding text layer output is given in the Figure 4.7 and Figure 4.8 by applying Gabor filter and LoG filter respectively incorporated with K-means++ clustering. The tabulated results using our annotated data of this approach is given in Chapter6.



(a) Input Image (b) Corresponding Text Layer

FIGURE 4.7: One sample map and its corresponding Text layer using Gabor Filter and K-means++



(a) Input Image (b) Corresponding Text Layer

FIGURE 4.8: One map and its corresponding Text layers using LoG and K-means++

4.2.3 Self-Learning based layer analysis

The proposed first step mentioned in Section 4.2.2, is based on specific filters working at pixel level. As this approach is not based on Connected Components Analysis (CCA), the problems of touching text detection does not occur here. On the contrary, as, we can see in the example Figure 4.7 or Figure 4.8, the resultant *Text Layer* contains false alarms : non-text pixels detected as text elements.

This second step tries to overcome this drawback with a more sophisticated framework including a self-learning strategy. The Pixel Level Analysis is based on a fact that text pixels share some common range of grayscale intensities around them (texture). This self-learning method want to take benefit on the fact that Text elements should also share some geometrical ot topological similarities. The geometrical or topological properties will be analyzed based on an original way to proceed to a CCA.

The main idea of the proposed method is to dynamically extract intrinsic properties of easily detected *Text* elements inside an image to detect the more difficult ones or to invalidate false detections. Intrinsic properties of the *Text* elements will be learned from easily extracted (non-ambiguous) *Text* elements and will be use to classify ambiguous ones.

Non ambiguous *Text* elements are determined by confronting the CC coming from the initial image with the CC coming from the associated *Text Layer* generated during the first step explained at the previous section. A high level of overlapping (a matching) between two CCs coming from each of these two sources of information is considered as a validation of a text detection (probability of being a *Text* element = 100%). These elements will then constitute the training data of a classifier that will be used to decide if the ambiguous

remaining elements (non-matched CC from the *Text Layer*) should be considered as *Text* elements or not. The work flow of this self-learning process is described Figure 4.9.

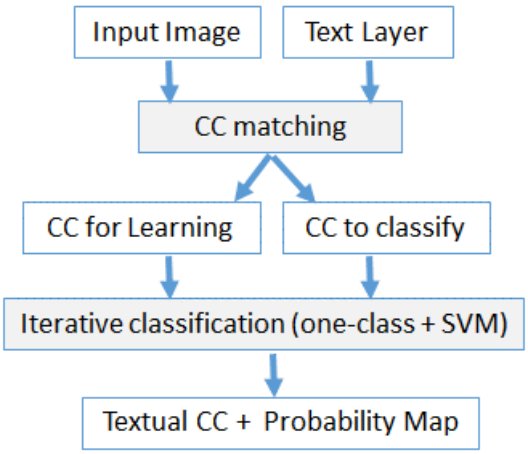
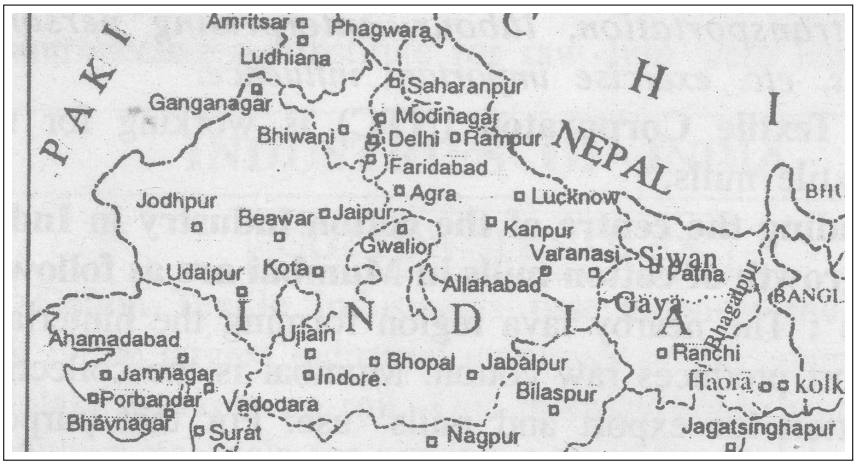
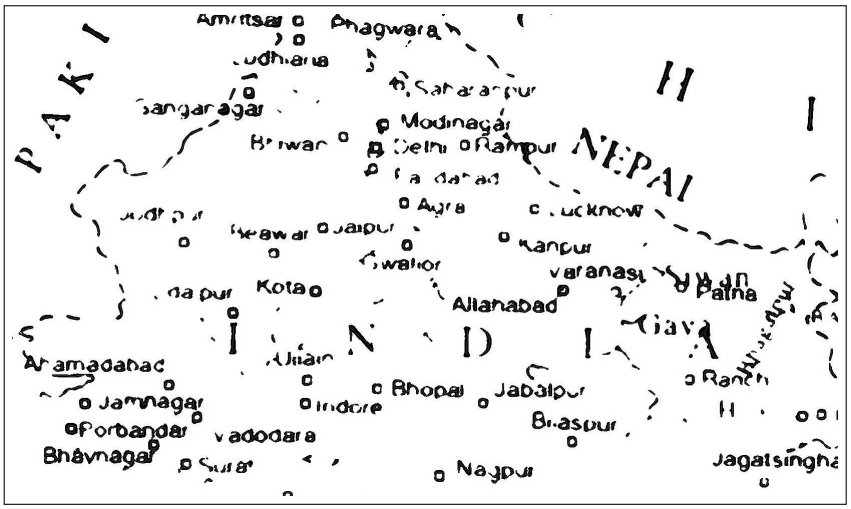


FIGURE 4.9: Work flow of the second stage (self-learning stage)

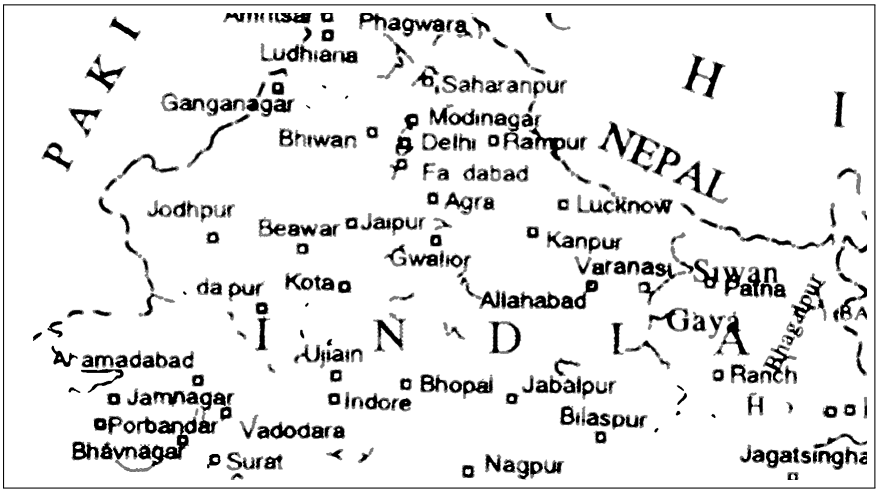
The next sections provide some details about the features and the classifiers used inside this work flow (of self learning stage). Figure 4.10 illustrates an input map with corresponding initial text image and corresponding modified text image after replacing with the CCs of the input image.



(a)



(b)



(c)

FIGURE 4.10: (a) A portion of input map, (b) corresponding initial text layer, (c) corresponding modified text layer

4.2.3.1 Features used for ambiguous CC classification

To choose a best suitable feature to be extracted from each CCs to refine *Texts* and *Graphics* components, we have compared several CC based features, which are Multi-Level Histograms of Multi-Scale Local Binary Pattern with Spatial Pyramid [Liao et al., 2007], Fourier-Mellin transform [Adam et al., 2000] and GIST-ARP [Liu et al., 2012]. The choice of features is based upon the fact that these features are meant to extract information about the topology and geometry of a pattern. Brief descriptions of these features are given in Appendix A.

4.2.3.2 Classifier model selection and definition

For the classification of the ambiguous CCs, we have selected a fundamental classifier, SVM [Vapnik, 1995]. SVM (Support Vector Machine) is one of the classic supervised classification approach, introduced by Vapnik [Vapnik, 1995]. This classifier is originally a binary classifier. The objective of SVM is to draw a hyper plane as the decision plane, which would be marked as the margin of separation between positive and negative examples of a class. Deciding the position of hyper plane should be in such a way that the separation between those examples would be maximized by utilizing optimization approach. Generally, this optimization technique is solved by a linear function. Non-linear functions are also used where positive and negative examples are not linearly separable. Examples of non-linear functions are polynomial function and radial-basis function. In our approach, linear SVM is used for the features as mentioned previously. A brief review of the of SVM is given [Cortes and Vapnik, 1995] in Appendix A.

4.2.3.3 The self-learning methodology

As we want to define a self-learning system, as shown in Figure 4.9, only positive training samples (coming from the actual image) will be available to learn the classifier. A one-class classification [Girard et al., 2016] method is then mandatory for the first iteration of the classification process. In our system, we have used SVM [Rätsch et al., 2000] as a One-class classifier. This first iteration allows to obtain non text examples (CC rejected by the One-class classifier) that can be used as negative training samples to learn a second classifier (classical SVM) used during the second iteration. This second iteration can be seen as a "revalidation" of the positive decisions taken during the first iteration. As explained in the next section, this iterative process also allow us to obtain **soft decisions instead of binary decisions** for the *Text layer* extraction.

Touching text extraction Other than revalidation of the CCs, to merge the touching or broken information with the result of second iteration, we perform a third iteration only to test initial touching text layer by taking training of two classes (text and non-text) from the result of second iteration. Those tested fragmented data are then merged with the result of second iteration. Finally, using such reinforcement, a final text image is constructed by merging fragmented text objects validated through same trained classifier with the text CCs. The Figures below show the effect of each iteration along with the final result. In Figure 4.11, we have shown example of a part of map. In Figure 4.12, the corresponding

output using previous/ first proposed approach of pixel based detection, is shown. In Figure 4.13, the corresponding modified text map after replacing with overlapping CCs from input image, is shown. In Figure 4.14, the corresponding fragmented data collected from initial text layer from previous approach, is shown. In Figure 4.15, the result of first iteration through one-class classification, is shown. In Figure 4.16, the result of second iteration through two-class classification, is shown. In Figure 4.17, the result of third iteration for final fragmented data merging through two-class classification is shown. In the Figures, we have marked few of the examples of improvement of the corresponding proposals.

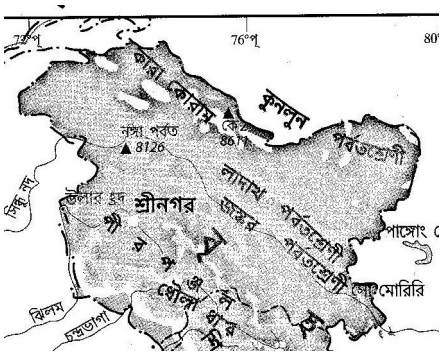


FIGURE 4.11: Example of a part of map

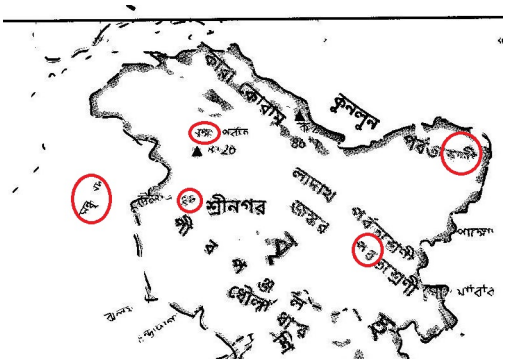


FIGURE 4.12: The corresponding output using Pixel level step (Gabor filter) detection. Red color circles denoted examples of some fragmented data, which are modified further by overlapping calculation

FIGURE 4.13: The CC from initial image matched with the result of first step that can be assimilated to non-ambiguous Text Elements (Positive Training samples for second step)

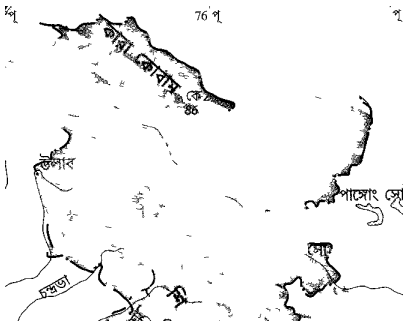


FIGURE 4.14: Non-matched CC corresponding to ambiguous element that need be re-classified / validated during the second step (self learning step)

FIGURE 4.15: The result of first iteration through one-class classification. The violet color markings are examples to denote the improvement of result through classifier.

FIGURE 4.16: the result of second iteration through two-class classification. The green color marking are examples to denote the modification from Figure 4.15

FIGURE 4.17: the final result after all iterations by selecting CC for which SVM provide a probability score $> 50\%$. Brown color markings are examples to show the modification happened by this iteration.

4.2.3.4 From binary decision to soft decision (probability maps generation)

Due to structural complexities, separating text and graphics from graphical document images does not always give an exact accurate decision. It is sometimes hard to draw an exact boundary between all sort of text and graphics present in the image. To encounter this issues, in place of a hard bounded binary decision about all the component portion to be *Text* or *Graphics Layer*, we have chosen to propose a more probabilistic solution. In this probabilistic approach, a probability (of being text) is assigned to the ambiguous component portions or pixels. Knowing that non-ambiguous CC used for the training of the One-class classifier receive a probability value equal to 1, the output of the SVM classifier is used in our approach to generate the Text probability map. This probability map is stored as part of the index in order to be used, if necessary, during the retrieval / spotting step. The results with probability associated to color map scheme is shown in below images in Figure 4.18. If a component is highly probable (more than 75%) of being text, it is colored in red. If it is more than 50% and less than 75% confident to be a text component then, it is pulled toward green. If a pixel is initially chosen as part of text but resulted with low

confidence (below 50%) to be part of text then it is pulled toward yellow. If a component is not whole but part of a connected component which part shows more than 75% confident of being text then the pixels consisting that part are colored in blue. Otherwise, if that part is more than 50% confident but less than 75% confident of being text then those pixels are colored in pink.

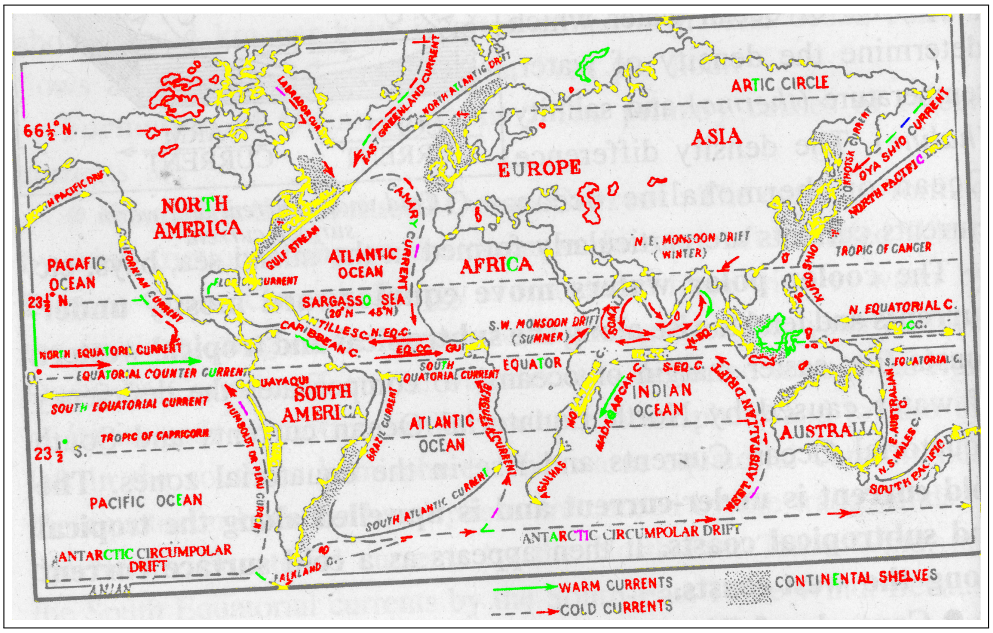


FIGURE 4.18: (a) Sample map colored to denote probabilities of being text

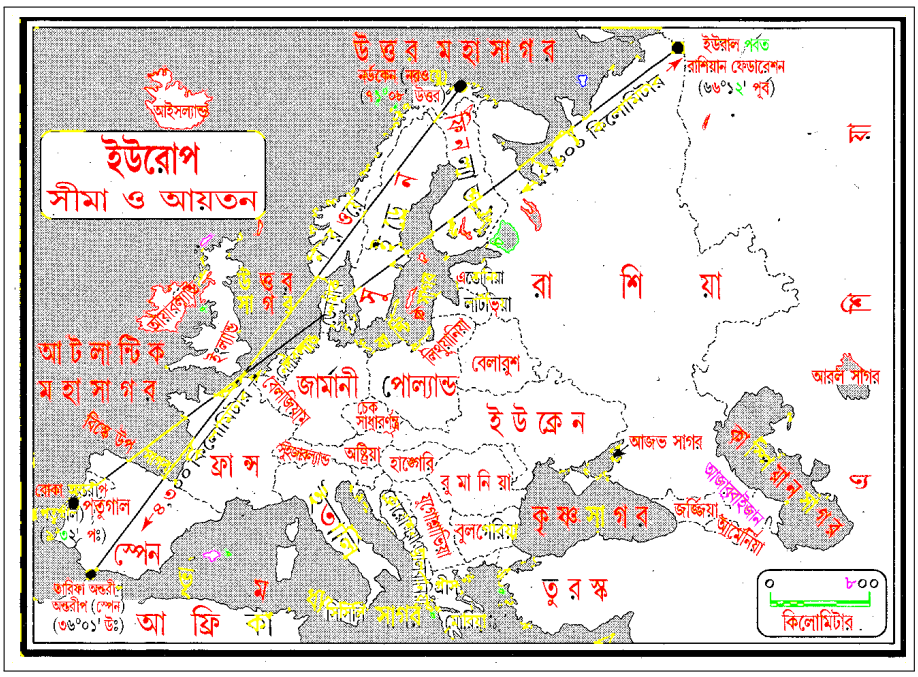


FIGURE 4.18: (b) Another sample map colored to denote probabilities of being text

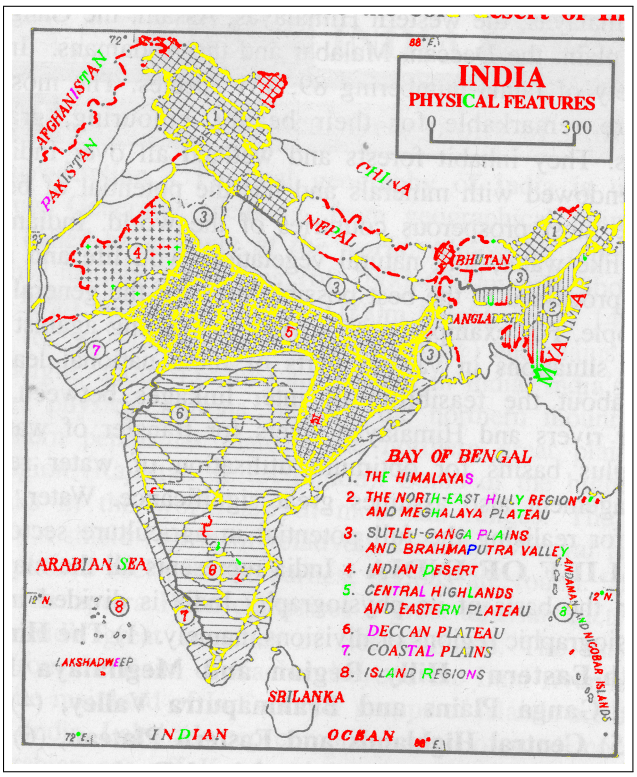


FIGURE 4.18: (c) Another sample map colored to denote probabilities of being text

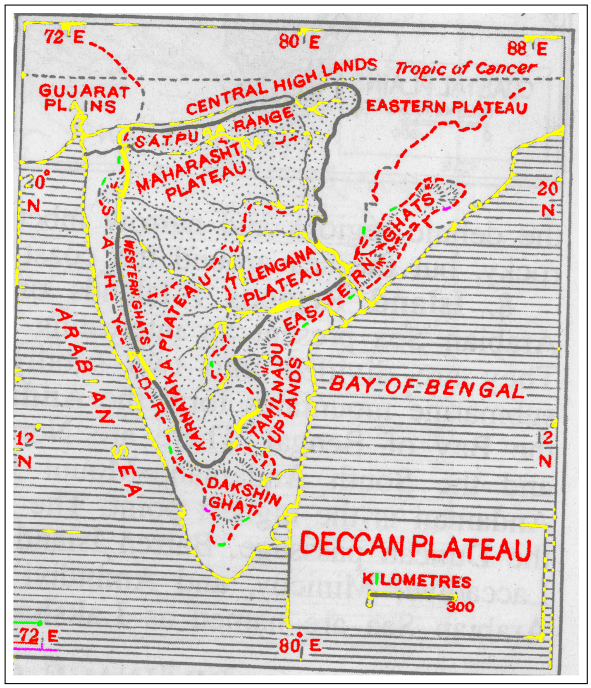


FIGURE 4.18: (d) Another sample map colored to denote probabilities of being text

4.3 Structural Level Indexing

4.3.1 Introduction

In addition to the Pixel level analysis, a more structural method can be used to differentiate between Text and Graphic. Figure 4.19 can illustrate how graphics part are corresponding to long structural lines whereas Text part are corresponding to set of connected very small line segments.

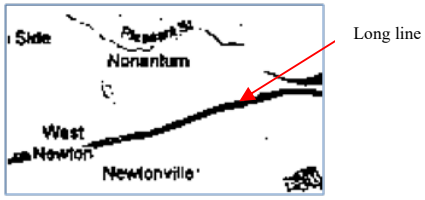


FIGURE 4.19: Graphics = long lines or curves and Text = small connected segments

This approach is similar to the previous one but it uses structural information extracted from the images instead of intensity or texture information. The structural representation of the image content will be obtained based on a vectorization technique applied on the binary version of the original Graphical documents. The proposed method contains three main blocks of

- ✍ Structural primitive extraction based on vectorization,
- ✍ Numerical Feature computation,
- ✍ classification of the pixels.

Block diagram of the proposed indexing method is shown in Figure 4.20.

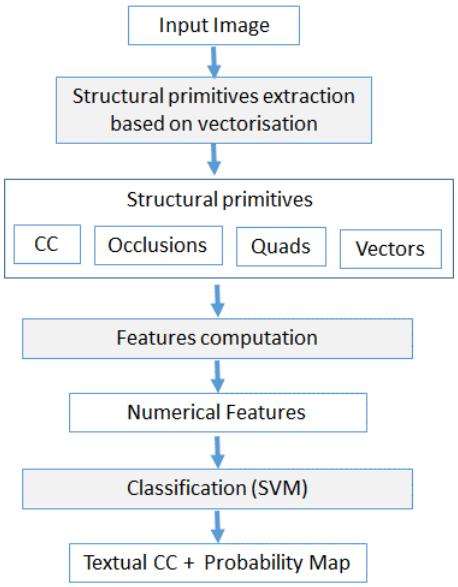


FIGURE 4.20: Block diagram of the Structural Level Analysis

4.3.2 Image vectorization

We use an original vectorization method based on contour tracking instead of classical skeleton techniques. Grey level images are first binarized for vectorization. The Otsu's method is used here to handle binarization, since this algorithm uses an adaptive threshold for the binarization process. The vectorization algorithm used in this work, is described in detail in [Ramel et al., 2000]. This technique is based on a polygonal approximation method, proposed by Wall and Danielson [Wall and Danielsson, 1984]. It is used here to approximate contours, extracted from the binary image. The contour of each component (set of pixels) is detected. These contour points are then approximated by a sequence (or even just one) of vector(s). Besides these vectors, the system also returns some additional structural primitives such as connected components, occlusions, and quads as shown in Figure 4.21. A detailed explanation of those primitives is provided below and in Figure 4.21.

- ✍ Connected components : that need no explanation
- ✍ Occlusions : It is defined as a region of connected background pixels. It is also known as background blobs ; they are separated from the others with pixels that are outside the surrounding range of foreground pixels.

- ✍ Vectors : Line segment defined by a starting and ending points and corresponding to the vectorization of a contour of a shape inside the image
- ✍ Quads : Each quad is defined by a couple of vectors (on each part of the two opposite frontiers of a thin shape). Quads are generated from two vectors that are closed and almost parallel. These two vectors are joined to draw a quadrangle shape.

(a) Input Image (b) Vectors (c) Occlusions (d) Quads

FIGURE 4.21: A sample word image and its corresponding structural primitives

4.3.3 Numerical feature computation

Using the result of the vectorization (lists of structural primitives described above), a set of features are extracted for characterizing parts of the image as text or graphics. To have a clear idea about the type of features, which can be extracted from the map images, and to easily discriminate text and graphics, a numerical feature study was performed on a set of 22 map images.

The features extracted after vectorization, are listed here :

- ✍ histogram plot of the size of connected components in textual and graphical parts
- ✍ quads sizes
- ✍ occlusions sizes and
- ✍ vectors sizes.

The size referred in connected component size, quad size, occlusion size and vector size means Euclidean distance between two corners most (either, top left and bottom right or, leftmost and rightmost) points of the unit (occlusion/quad/vector/connected component). It has been observed that words are mostly made of groups of small strokes composed of black pixels, glued together. Whereas, graphical entities in map images form long lines, big set of black pixels, dispersed over a wide region. For each black pixel group, we would have one connected component (in an ideal scenario). In such case, these connected components are filtered according to their size, which is very different from graphical components size. As a result of the numerical feature study, the features which are finally selected to filter text components are synthesized in Figure 4.22.

4.3.4 Structural classification of the pixels

Like pixel level indexing as mentioned in 4.2.3.4, in structural level indexing also, we have implemented the concept of probability map. The structural features described before are computed for all the foreground (black) pixels of the original images in order to be submitted to the classifier. As before, we selected a binary SVM classifier to discriminate between text and graphical part of the image and the result are stored inside a probability

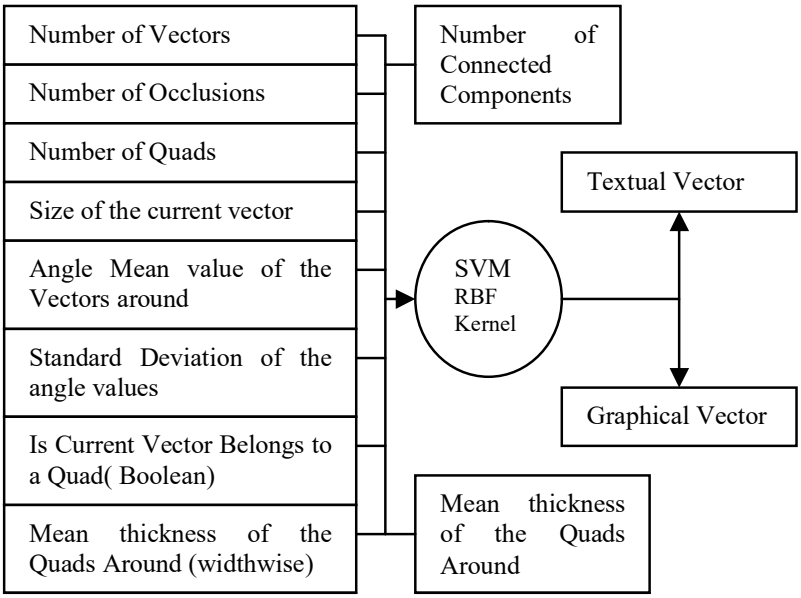


FIGURE 4.22: Block diagram of the Structural Level Analysis

map that is stored in order to be used during the retrieval step. The same 22 images as mentioned before have been used to generate the training data used to learn the classifier. Figure 4.23 shows an example of probability map coming from the structural level analysis of a map.

4.4 Lexical Level Indexing

4.4.1 Introduction

Unlike pixel or structural level indexing, the lexical level indexing work from a different perspective which provide a semantical labeling as an output of the indexing stage. As an end solution of our work, we would like to produce a word spotting system that can work as well with query by string (keyboarded query) and query by example (query image). When the query (for word spotting) is a ASCII string, the better way to index document is to transcribe them using an OCR system in a similar way as Google for the web pages. The indexes generated on line are then based on lexical analysis (dictionary of the present words / characters) of the initial data. It seems evident for us to integrate also a such possibility in our indexing scheme as some text parts of graphical documents can easily be transcribed automatically. Of course, we are aware that not all the parts of a graphical document can follow this process whereas we should not speak any more of a spotting system. But as already mentioned before, there is a significant chance that any document contains several non-ambiguous, easily detectable and recognizable parts that it would be a shame to not use to better index the content. That is the reason why we incorporate the Lexical Level Indexing stage inside our indexing system with the goal to generate an index

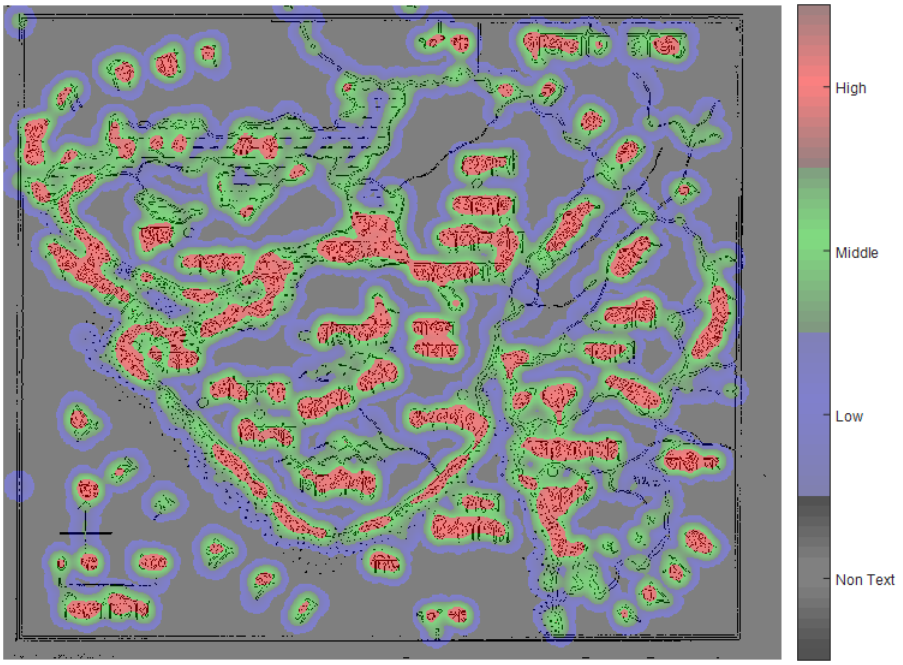


FIGURE 4.23: Sample probability map denoting high, middle and low probable text zone and non-text zone using heat map scheme

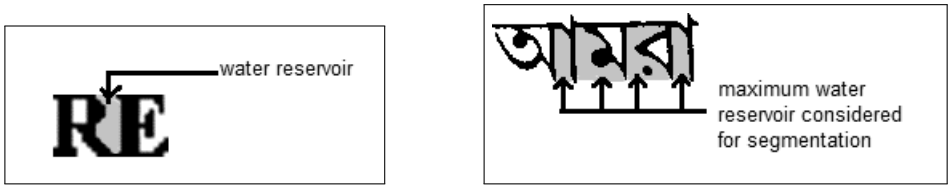
file containing information about the recognized characters and words inside the images to index (knowing that not all will be present because of missing or ambiguous regions). The lexical level analysis has to be done on the text layers and tries to extract individual characters in order to submit them to a dedicated OCR. The two *Text Layers* generated by Pixel Level and Structural Level Analysis can contain different types of text areas or components according to the scripts and fonts used inside the graphical documents (Bangla, Latin etc.) or worse still could correspond to non-text elements! In this regard, the areas or components categorized as non-isolated characters (full words or touching characters) have to be segmented before submission to the OCR.

4.4.2 Potential text areas selection from probability maps

Different strategies are possible in order to decide which parts of each image have to be considered as potential text areas in order to be submitted to the Lexical Level Analysis. Only connected components identified as text during the Pixel Level Analysis can be kept here if the strategy is to limit at maximum the number of potential text areas. It is also possible to define a threshold on the probability maps to define in another way potential Text areas. Only pixels or areas with probability higher than the selected threshold in the probability map will constitute potential text elements that could be submitted to the Lexical Level Analysis. Finally, more complex strategies can be imagined based on a fusion between all these sources of information to decide which areas should be submitted to the Lexical Level Analysis.

4.4.3 Character segmentation

Characters in a word generate big cavity regions in between two characters if they touch each other in both, English and Bangla scripts. In Figure 4.24, two such examples of segmentation in English and Bangla scripts are shown respectively. However, unlike English script, generally, in Bangla script characters are connected with each other by a straight line known as headline to form a word. Consequently, for both of the scripts, the possible boundaries of character segmentation belong to these big regions and these big regions of the background portion are detected by water reservoir principle.



(a) (b)

FIGURE 4.24: (a) English touching component successfully segmented using water reservoir principle, (b) Bangla word successfully segmented using water reservoir principle

4.4.3.1 Water reservoir principle

When two characters touch each other, generally they create a big cavity region in between themselves. This big cavity region get created by generation of open contours around the touching point. In water reservoir concept, the open contours/edges of connected components are assumed as obstacle/wall of the reservoir. If some imaginary water can be poured from the direction of open portion of the contour ; that water would be reserved/stored in the open cavity regions of the components. Here, those cavity regions work like water reservoirs. For example, if we refer Figure 4.25, we could visualize the concept of water reservoir. Now, to deal with multi-oriented words, initially four directions are considered to simulate pouring of water. Those directions are, top, bottom, left or right of the image. Consequently, the direction of water falling, is taken into consideration, in which the highest amount of water could be stored. The orientation of a touching component (word) would be same as that particular direction. This method is already used successfully for segmentation purpose [Pal et al., 2010] but not yet used for an integrated word spotting system. Next, depending on orientation and the reservoir based-region of the touching component, a set of candidate envelope points is selected from the contour points of the component. We can detect the orientation of the component using the direction of the corresponding water reservoirs. For example, the component shown in Figure 4.25, is in portrait mode because the reservoirs obtained from bottom side of the component have maximum area. Next, for every reservoir where the reservoir height is maximum, those points on contour are chosen as candidate points for segmentation. Consequently, candidate points obtained from bottom reservoir are used for segmentation of this word. Likewise, the touching component is finally segmented into individual characters. Figure 4.26 below shows some results of wrong segmentations. But even if some wrong segmen-

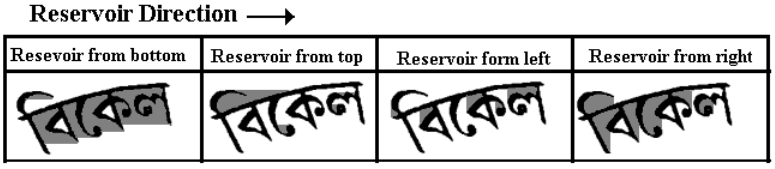


FIGURE 4.25: Component with its reservoirs from bottom, top, left and right sides are shown here. Reservoirs are marked by grey shade. Here Portrait, Reverse portrait, Landscape and Reverse landscape directional reservoirs are shown respectively.

tation occurs, this problem could be resolved using missing candidate analysis by SIFT, discussed later during online retrieval procedure.

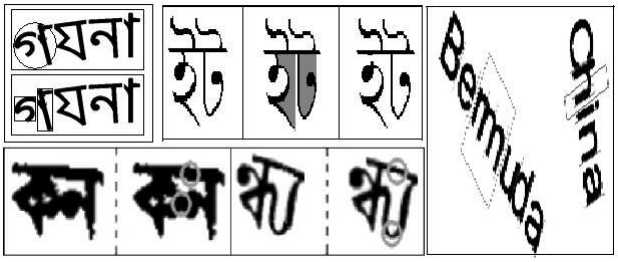


FIGURE 4.26: Examples of wrong segmentation results using water reservoir principle in Bangla and English script

4.4.3.2 Segmentation criteria

The criteria for deciding whether a component is comprised of multiple characters and to be considered for segmentation are discussed here. The very first criteria related to segmentation decision is the aspect ratio of the particular connected component. Naturally from a distant view, we may hypothesize that the aspect ratio of two or more than two characters, touching together, would be higher than the aspect ratio of single/isolated characters. To have a closer look and verify our hypothesis, we have calculated aspect ratio of all isolated characters and all combination of two characters touching of English and Bengali script manually generated from one or two popular fonts. We have noted that, for, English script, font Times New Roman, the value ranges of aspect ratio for two characters touching is from 0.809 to 2.706 and for font Arial, these values range is from 0.783 to 3. Whereas for, Bengali script, the value ranges of aspect ratio for two characters touching is from 0.837838 to 1.246154. If we assume, threshold of aspect ratio for two characters touching set identification, as greater than the highest aspect ratio of isolated characters of the particular script, then we would fail to notice majorities of two characters touching units of the particular script. But if we change the threshold to a bit lower value and set a number which is above than aspect ratios of most of the isolated characters and below than aspect ratios of most of the two characters touching unit, then the threshold would have a maximized effect to detect touching characters. For example, for English script, the

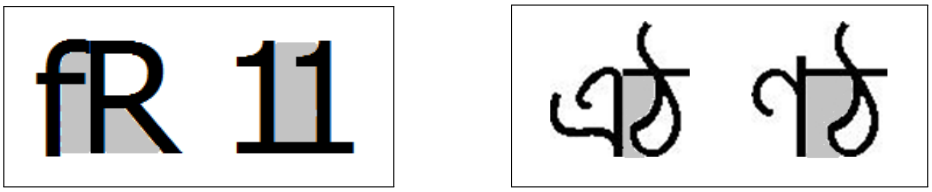
highest aspect ratio of isolated characters is 1.6 and for Bengali script, the highest aspect ratio of isolated characters is 1.286. Here, for English script, the threshold could be set as 1.09 and for Bengali script, the threshold could be set as 1.05. For English script, among 3844 two-characters touching pairs, only 169 character pairs are having aspect ratio more than 1.6 whereas, 1187 character pairs are having aspect ratio more than 1.09 and among 62 isolated characters, only 4 isolated characters are having aspect ratio more than 1.09. For Bengali script, among 2809 touching character pairs, 213 pairs are below 1.286 aspect ratio, whereas only 37 pairs are below 1.05 aspect ratio and among 53 isolated characters only 9 characters are having aspect ratio more than 1.05.

Considering this scenario, we could decide the criteria for segmentation of a component is that, if the aspect ratio of the component is greater than 1.6 (for English) and 1.3 (for Bengali) then system would directly go for segmentation. Otherwise, if it is greater than 1.09 (for English) and greater than 1.05 (for Bengali), then system would check for remaining criteria for segmentation.

Now, we have given a closer look to the set of isolated characters and touching pair characters where aspect ratios are overlapping and difficult to decide about them as isolated or touching one depend on their aspect ratio values. In case of English script, we have noticed that the touching character set which are consist of at least one almost linear isolated character i.e. i, l, t, f and/or j, then the resulting aspect ratio will be under the aspect ratio 1.09.

There are few properties of the components consisting of i, 1, t, f or j are discussed below :

1. They comprise of a vertical bar and after touching with some other isolated character, they mostly generate a deep water reservoir which is almost equal to the height of the character. Two of such example pairs of Roman script (English) can be seen below in Figure 4.27a. The gray color portion shown in the image is the water portion that could be reserved using the water reservoir principle. Using this condition, we have set criteria two for segmentation decision regarding English script, which is, if the component contain vertical bar and water reservoir, and both of their heights are greater than 50% of component height then that reservoir or the component is considered for segmentation.
2. Moreover, for both of the Bengali script and English script, character combination with digit and alphabet or, character combination with lowercase followed by uppercase (for English script alone), are impracticable combinations. If we follow the bi-gram frequencies [Jones and Mewhort, 2004] of all other combinations then also, we may discard/ignore few of the possible combinations with low aspect ratio, for example, fd, fh, jc etc. which have extremely rare/no chances to occur according to their bi-gram frequencies.
3. For both of the English and Bengali script, other than incorporating all the above conditions to be validated for segmentation, we can generate training set using those particular few combinations to be recognized along with isolated character recognition. The particular touching character pairs of Bengali script is given in below Figure 4.27b. The gray color portion shown in the image is the water portion that could be reserved using the water reservoir principle.



(a) (b)

FIGURE 4.27: (a) Water Reservoirs generated by touching character pairs, (b) Touching pairs of Bengali script having aspect ratio lower than 1.05

Block Diagram of this character segmentation process shown in Figure 4.28.

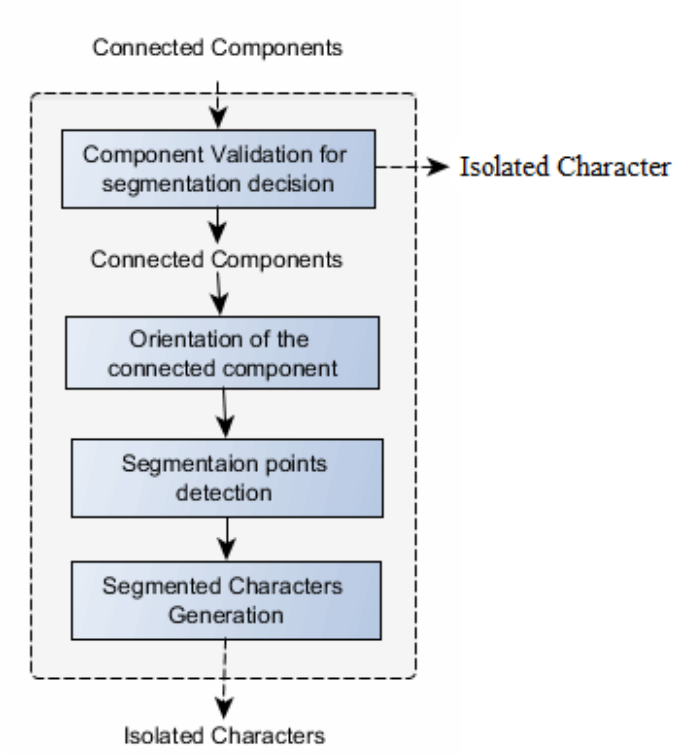


FIGURE 4.28: Block diagram of Character Segmentation

4.4.4 Character recognition

As already mentioned, in case of graphical documents, the feature used for character recognition should be efficient enough to distinguish and recognize the multi-oriented, multi-scaled characters of English or Bengali script. However, recognition of text characters in multi-oriented and multi-scale environment is a challenging task and different shape descriptors like Angular Radial Transform (ART) [Ricard et al., 2005], Hu's moments [Hu, 1962], Zernike moments [Khotanzad and Hong, 1990], Multi-Level Histograms of Multi-Scale Local Binary Pattern with Spatial Pyramid (MLHMLPSP) [Liao et al., 2007], GIST-

ARP [Liu et al., 2012] and Fourier-Mellin Transform [Adam et al., 2000], etc. Brief details of these features are given in AppendixA.

4.4.4.1 Classification of isolated characters

The appropriate feature is fed to support vector machine (SVM) [Chang and Lin, 2011] classifier for the recognition process. We have already described about the classifier in earlier Section 4.2.3.2. Besides, we have discussed about training data set of isolated characters of both the English and Bengali script in Section 3.2. Now, considering the recognition part through classifier, we test **any individual textual component** incorporated with the feature to provide a confidence score i.e. score of membership to any particular character class. If the score is greater than 30% in SVM prediction, then that component is considered as recognized one. The threshold value is experimentally set and kept low i.e. 30% because we are handling characters which are present in scanned maps which may contain degraded, partially broken or noisy characters. Otherwise, if component give a recognition score with even lesser confidence than 30%, then component is considered as unrecognized one and left as it is to be dealt in a later stage, if required as part a query word spotting.

4.4.4.2 Clustering for word reconstruction

It has been noticed that graphical documents contain characters of certain sizes, whereas the words in such documents contain characters of a particular size. As we do not have indefinite number of character sizes here, all recognized components are grouped in several clusters according to their sizes. Consequently, we can group the characters that belong to a word or belong to similar sized words, under a single cluster for easier access during online query spotting. Clustering is done based on two rotation invariant size measurements of the components. The measurements, which are taken into consideration are major axis length of minimum bounding ellipse of the component and [major axis length ÃŮ minor axis length] of minimum bounding ellipse of the component. Experimentally, it has been noted that in this clustering approach, among K-means [Hartigan and Wong, 1979], fuzzy C-means and hierarchical clustering approaches ; fuzzy C-means performs the best. Fuzzy c-means (FCM) is an unsupervised clustering algorithm. This grouping of characters is done to easier word retrieval from these documents.

4.5 The Resulting Indexed Data Structure

As a result of the works described in previous sections and as shown in Figure 4.1, we have collection of potential text components sometimes with a lexical label (recognized character) and a collection of probability maps providing information about the confidence we can have on this extracted knowledge Resultantly, an offline database (index files) is prepared containing information about the content present in a graphical document. The database is indexed to result in a faster online retrieval system. The data obtained and stored in databases which mark the final outcome of our offline process is shown in Table 4.1. The offline data contain the Bounding Box Information, Centre of Gravity (x, y coordinates), Major Axis Length of Minimum Bounding Ellipse, Minor Axis Length of Minimum

Bounding Ellipse and Unique serial number to label the corresponding particular connected component (assigned during CCA). Whereas, Label of isolated character component is extracted during classification through SVM. Cluster number is extracted during clustering of the components according to their size. Orientation of the component is extracted optionally through segmentation.

TABLE 4.1: Information stored as index and signature resulting the offline indexing

Label of isolated character components (resulted from SVM classifier)
Bounding Box Information
Centre of Gravity (x, y coordinates)
Orientation of the component (optional)
Major Axis Length of Minimum Bounding Ellipse
Minor Axis Length of Minimum Bounding Ellipse
Unique component number (assigned during CCA)
Cluster number (depends on component size)
Probability of being text coming from pixel analysis and structural analysis respectively

Moreover, we have automatic multi-level indexing for faster retrieval by accessing the database primarily through the label of isolated character component. Next, depending on the present retrieval we narrow down the search space to spot the other characters belong to the same word. Moreover, to spot or estimate the missing character we make use of probability map for faster retrieval.

4.6 Summary

In this chapter, we have done all the processing stages in order to prepare a efficient online word spotting system. This offline preprocessing stages include stages to develop a proper indexing for word retrieval system. The indexing first encompass pixel level indexing, then structural level indexing which consider text-graphics separation and second, it encompass lexical level indexing which involve character segmentation and character recognition. Finally, the system prepare an offline indexed data base to result in a faster and efficient word retrieval system. For effective characterization of the content, we have proposed both pixel based approach and CCA based approach. Moreover, we have relied on water reservoir approach for segmentation and partially recognition based approach for recognition. In the consequent chapter, we have elaborated the retrieval process using the processed information, that are discussed in this chapter.

Chapitre 5

Incremental Content Retrieval and Spotting

"If you want to have a strong structure, build the foundations the right way."
- Eraldo Banovac

Contents

Abstract

This chapter presents the on-line stage of the proposed word spotting system. After explaining that the starting point of the retrieval is based on the indexed lexical information, the main process involved are described in details. The first step is to generate seed regions of possible occurrences of the query. Then, an incremental retrieval of the eventual remaining parts of the query is realized in order to validate the previously selected seed regions.

5.1 Retrieval Method

5.1.1 Introduction

Figure 5.1 (similar to Figure 4.1) recalls the global workflow of the proposed system. In this chapter we will present in details the retrieval stages. As indicated in Figure 5.1, it

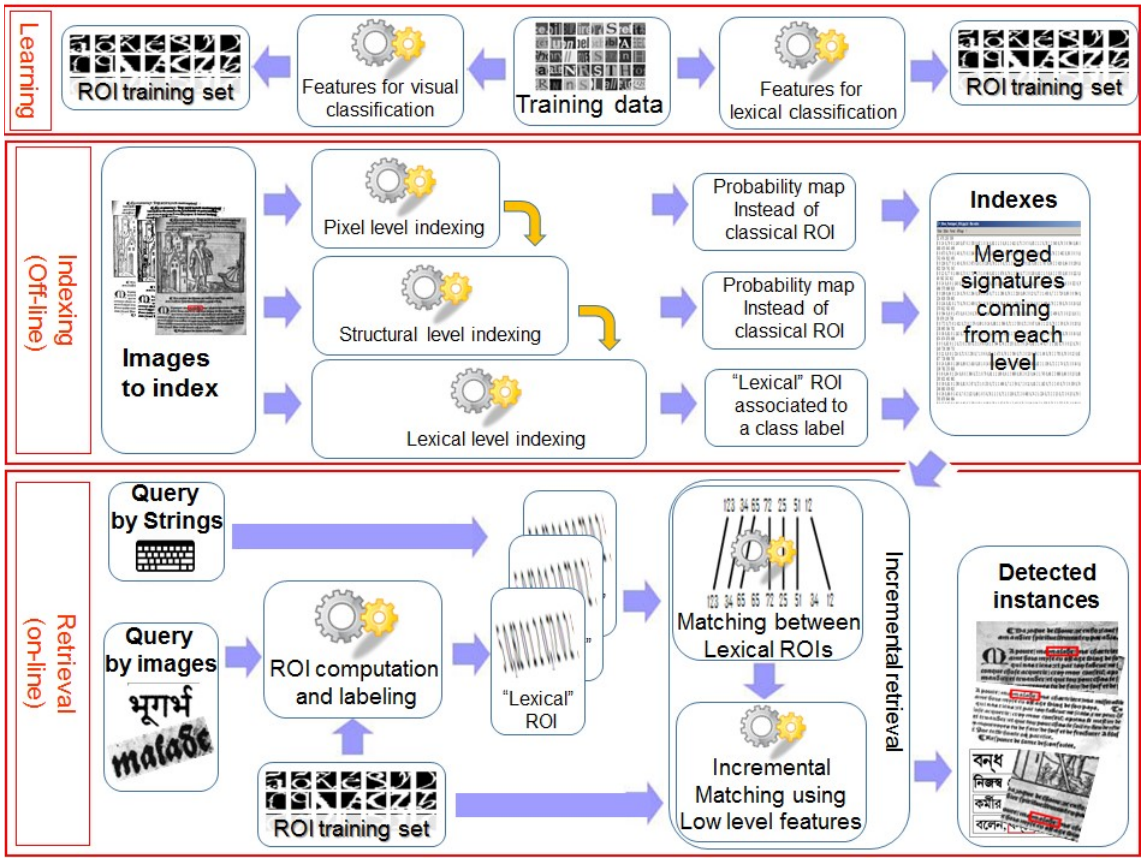


FIGURE 5.1: Global workflow of the proposed Retrieval system. At the top, the learning stage ; in the middle, the multi-level indexing stage (discussed in Chapter4) and the incremental retrieval step at the bottom

should be noticed that two kinds of query can be used to initiate the retrieval step : query by string and query by image. If a query by image is used, then a supplementary processing is needed to generate what we call the "Lexical ROIs" (in Figure 5.1) that correspond to the input of matching process in the both cases (query by string and query by image) The "Lexical ROI" are in fact the ordered sequence of labels associated to the isolated characters constituting the query word. This sequence of labels is directly corresponding to the query in case of query by string provided through a keyboard. In the other case, the query image has to be submitted to the segmentation and recognition process described in Section 4.4 to generate the initial ordered sequence of "Lexical Labels".

5.1.2 From the query to seed selections into the indexed images

During the online retrieval step, a query (word) represented by an ordered sequence of "Lexical Labels" has to be spotted inside the pre-indexed graphical documents. The first step of the retrieval process is quite simple and is based only on the Lexical information stored in the index files (see data structure presented Section 4.5). The data structure of each image from the dataset is parsed and the indexed images are ranked according to the number of characters and associated confidence rate corresponding to the one present in the query that is present inside the image. A first threshold (concerning number of characters and associated confidence rates) can be inserted here to limit or reduce the number of selected images to be analyzed in more details. Each selected image is then considered / analyzed and **seed regions** are generated according to the positions of the different elements labeled with similar labels as the characters constituting the query.These seed regions are computed using a nearest neighbor analysis against previously found character elements. After an initial character searching, spotting is progressed by neighborhood analysis using nearest neighbor analysis. Depending on the relative position of the character to be searched with respect to the position of current spotted character in the query, the number of nearest neighbors, K to seek for is defined according to the position of character inside the word. This searching and spotting process is described below in detail with the help of two examples of Bangla and Roman (English). For example, with the query word 'PRINSESSE', all individual connected components labeled as 'P' inside one of the pre-selected image are considered. Then, the next characters in the query are searched sequentially in the neighborhood of all the spotted 'P' using k-nearest neighbors. As shown in Figure 5.2, first character 'P'is spotted. Next, 'R' followed by 'I' are spotted and so on... The 'N'character will not be found as it has not been recognized during character recognition. Anyway, the method will try to find 'S'. Likewise, next 'E' and 'S', 'S', 'E' would be searched and spotted one-by-one.

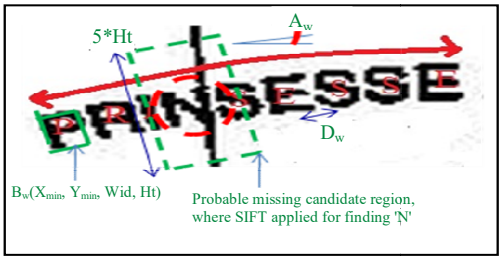


FIGURE 5.2: For keyword 'PRINSESSE' || only 'N' is not recognized in first stage, (green dashed box is estimated candidate region)

While spotting, these characters are taken into consideration if they satisfy the following three criteria :

1. Distances between the CGs (Centre of Gravity) of two consecutive characters should be similar,
2. Two consecutive characters should be similar in size and,
3. Angle between CGs of consecutive character pair with horizontal axis should be

similar to the earlier pair.

This spotting is called initial spotting. Of course if a seed region is containing all the "Lexical ROI" constituting the query and correctly ordered, then, this region is directly inserted in the top of the final ranked list of spotted regions. All the other seed regions, with missing characters, are submitted to the incremental retrieval technique presented in the next Section.

5.1.3 Incremental retrieval technique using neighborhood analysis

Following above procedure, if partial matching is found for a query word, we assume that there may be some missing characters lie in between. Some character information in the document may be missed due to noise created by touching / connection with graphical entities and other text components. These characters could not be considered for initial spotting. Always using the same example ("PRINCESSE"), Figure 5.2 shows an example of partially recognized text. We could see from Figure 5.2 that SVM recognizes all characters except 'N' due to it is touching with a line. Using recognized portions of a candidate word, but also associated part of the probability maps generated by Pixel Level and Structural Level Analysis during the indexing, the location of the query characters which are not spotted yet are estimated and this located region is called as **candidate region** of missing characters. After spotting of two characters, a rough angle (A_w) (shown in Figure 5.2 and in Figure 5.3) is estimated using the Centre of Gravity (CG) of the spotted characters. Also spacing or distance (D_w) between them and an approximate height (Ht) and width (Wd) of each character are obtained from their bounding box (B_w) information. The spotting of the successive characters constituting the query (to generate a spotted region) takes into consideration the following three criteria :

1. D_w should be similar for the pairs of characters for the same word,
2. B_w should be similar in size for characters belongs to same word,
3. A_w should be similar for the same word.

The level of similarity is quantized based on threshold set experimentally.

In our actual approach, at least two query characters must be spotted during initial spotting. If missing characters lie in the middle of a word, then it is easy to assume their position using positional information of last spotted (L_p) character before missing characters and next spotted (N_p) character after missing characters are considered. Starting from (L_p), a region up to (N_p) having height ($5 \times Ht$), as shown in Figure 5.2, is estimated from recognized query characters considering similar angular direction of (A_w). This estimated region is the **candidate region** of the missing character. To avoid chances of not retrieval of any of the missing characters, we consider maximum possible candidate region according values inside the probability maps and hence we also use the threshold of $5 \times Ht$. If any of the extreme characters of a word is missing then extreme portions with height ($5 \times Ht$), width ($4 \times Wd$) and similar angular direction (obtained from the two extreme spotted characters) are considered as **candidate region**. Figure 5.3 shows an example of Bengali script, where a similar process is used for the initial spotting of a candidate region by connected component analysis. Next, a similar method for a possible missing characters in which we will have to seek for the missing character with a more adaptive technique.

Here, the query word is "শারদ" and the missing character in the spotted area is "র", whereas other characters, "শা", "দ" in the same word are recognizable and so spotted.

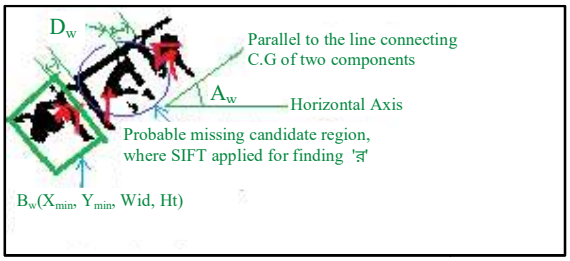


FIGURE 5.3: For keyword "শারদ" , where only "র" is not recognized in first stage, (green dashed box is estimated candidate region)

5.1.4 Seeking for missing character with SIFT

Recognizing an object using SIFT can be performed using the rotation-invariant Key-point matching techniques [Lowe, 2004]. This matching process involve nearest neighbor matching and indexing them through Best-Bin-First [Beis and Lowe, 1997] algorithm. In Figure 5.4, we could see one query character spotting using SIFT and its key points descriptor regions.

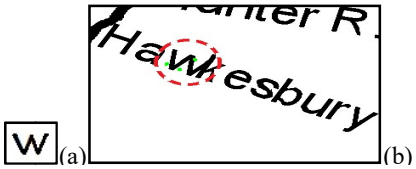


FIGURE 5.4: A matching in SIFT is shown in (b). Here query images for SIFT is 'W' as shown in (a). (Red dashed circle is our estimated candidate region where missing character 'w' may be present and green points shows matching)

If all characters from a query would be searched in whole document images by applying SIFT, many false alarms would be found [Roy et al., 2009a] due to nature and difficulties to handle of graphical documents. That is why, we propose to use SIFT only on some particular candidate regions selected by the help of lexical information. The proposed method uses SIFT to find missing query characters in precise estimated candidate regions. Furthermore, SIFT is applied in this precise region, knowing the label of the character we are looking for. On the basis of SIFT key-point matching in the precise region, we validate or reject the presence of the missing character in that region. The selected region, where the missing character should be sent to the SIFT keypoint extraction. Then the SIFT keypoints have to be matched with SIFT keypoint coming from a model of the selected character. We first consider the label of the missing character that suppose to be present in the precise region. If the corresponding whole input image contains any recognized character belong to the same label, before using a model of character from our existing training set, we try to use

the occurance of that seeked character already labeled inside that image,we are processing. Consequently, we consider that character as a template for SIFT keypoint matching in that precise region. Otherwise, if none of the recognized character from that image contain the particular label, then we use some template character for keypoint matching from our database, that belong to the same label (as shown in our global workflow presented Figure 5.1). In both of the cases, correspondingly, SIFT is applied only in the estimated region and for validating the presence of this known missing characters a reliability score is computed based on the SIFT key point matching score and also on the values stored in the probability maps for that precise region. After spotting using SIFT, the compatibility of the characters obtained by initial spotting made by SVM recognition and the candidate region spotting made by SIFT matching has to be checked. If the compatibility among the characters is satisfied, then the spotted word is inserted in the ranked list of spotted region. Compatibility checking is done in terms of position, size and orientation of the characters present in the spotted area.

5.2 Summary

In this chapter we have discussed about the exact query driven retrieval procedure. This procedure is mostly based on the offline data which are indexed and prepared by different preprocessing stages from raw map images. A small portion of the image could be processed during the offline procedure due to associated noise that are present with those data. Those small portion is processed incrementally here using recognition free approach by SIFT keypoint extraction and matching. Mainly, for word retrieval , we have used the spatial arrangement of the textual data that are present in the geographical documents.

Chapitre 6

Experimental Results

"It doesn't matter how beautiful your theory is, it doesn't matter how smart you are. If it doesn't agree with experiment, it's wrong."
- Eric Schmidt

Contents

Abstract

In this chapter, we present the experiments and results of the proposed approaches. The first part is concerned by the methods used during the off-line indexing stage that we can be compared with state of the art Text Graphics separation techniques. The second part deals with the evaluation of the full spotting system on different databases of images and with different types of queries.

6.1 Used Protocols, Metrics and Dataset Description

As described in Chapter 4, the proposed multi-level indexing is realized incrementally. Different steps are involved during the workflow : Pixel level-analysis, Structural level analysis and finally Lexical level analysis. The experiments presented below describes how we have selected the best parameters and features for each of these processing. Finally, as the proposed approach can be assimilated to a text-graphics separation stage, the results of our method is compared with results obtained with the QGAR framework [Tombre et al., 2002].

To evaluate the performances we use all along this chapter, the 2 sets of numerical measures for performance reporting, described in Section 2.2. The details of the dataset used for experimentation, are already mentioned in Chapter 3.

6.2 Experimental Results of Selected Approaches for Indexing

We have first experimented the different options for the pixel level analysis of the indexing stage. The question is here to decide between Gabor and LoG features and also to determine the best algorithm and parameters for the clustering of the pixels described by such features. The reported tests have been done with 4 clusters and 5 clusters and with K-means and K-means++ respectively. The choice of number of clusters is considered as 4 and 5 after multiple experimentation with different number of clusters. The different filter parameter values are already mentioned previously while describing the approach. For further stepped up experimentation as mentioned in Section 4.2.3, we have done classifier based testing comparison. The approaches are experimented using the available ground truth data as mentioned in Section 3.3. In the pixel based performance measurement, we have calculated the overlapping between the ground truth pixel and resultant pixels. In bounding box based performance measurement, we have calculated the overlapping using method (as mentioned in Section 2.2) proposed by Wolf and Jolion [Wolf and Jolion, 2005] by ground truth xml and resultant xml. We have generated such performance data with the ground truth of 47 Bengali maps and 49 English maps.

6.2.1 Filter based approaches

Pixel Level analysis

The tables below Table 6.1 and Table 6.2 contain five columns, the first one contains the name of the specific variation of the proposed approach, along with the number of clusters, it uses for clustering. Second column mentions about the clustering technique. Third, fourth and fifth column mentions the results in term of the precision, recall and harmonic mean respectively, of the **Text Layer** extraction after applying the steps described Figure 6.1.

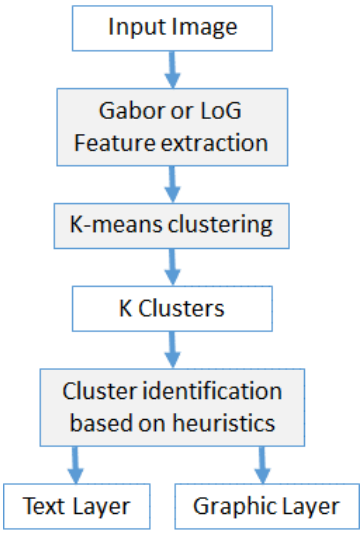


FIGURE 6.1: Block Diagram of the filter Based Approach

In Table 6.1 precisions, recalls and harmonic means are given using pixel-based calculation [Kumar et al., 2007]. In Table 6.2, we have given precisions, recalls and harmonic means based on bounding box [Wolf and Jolion, 2005] analysis.

TABLE 6.1: Performance values of initial text detection in 96 multi-script maps using Pixel based calculation

Proposed Approach	**Clustering**	**Precision**	**Recall**	**Harmonic Mean**
Gabor with 4 clusters	K-means	0.611364	0.467368	0.529755
	K-means++	**0.615894**	0.466503	**0.530889**
Gabor with 5 clusters	K-means	0.538641	0.46413	0.498617
	K-means++	0.541555	0.469094	0.502727
LoG with 4 Clusters	K-means	0.524534	**0.49078**	0.507096
	K-means++	0.511041	0.490431	0.500524
LoG with 5 Clusters	K-means	0.435522	0.472127	0.453086
	K-means++	0.42862	0.461798	0.444591

Conclusions about filter based analysis

In the Table 6.1 and Table 6.2, the best percentages are marked in boldface font. Accordingly, after reviewing the results from these tables, we may conclude that in filter based approach, LoG filter with four clusters K-means and K-means++ and Gabor filter with four or five clusters, K-means++ are giving top results either in precision or, recall or, harmonic mean. Consequently, further stages of the filter based approach using classifier is compared or experimented using these four variations of the initial proposed approaches. We could see the results of existing approach is not feasible enough that could be used for

TABLE 6.2: Performance values of initial text detection in 96 multiscript maps using Bounding Box based calculation

Proposed Approach	**Clustering**	**Precision**	**Recall**	**Harmonic Mean**
Gabor with 4 clusters	K-means	0.416213	0.487711	0.449134
	K-means++	0.436869	**0.544643**	0.473964
Gabor with 5 clusters	K-means	0.458525	0.477609	0.467873
	K-means++	0.454835	0.494773	**0.484839**
LoG with 4 Clusters	K-means	0.4134	0.47853	0.443587
	K-means++	**0.461542**	0.478793	0.470009
LoG with 5 Clusters	K-means	0.455535	0.476994	0.466018
	K-means++	0.461458	0.473151	0.467231

text extraction process. We have proposed and experimented further development of the extraction procedure on the top of this initial filter based analysis.

Results with the additional self-learning step (first iteration)
Below Table 6.3 and Table 6.4 show pixel based calculation and bounding box based calculation respectively of the amount of text extraction **after adding the self-learning step** in the indexing chain. In these tables, the result reported are of the amount of text which are the modification on the top of filter based analysis using their overlapping with the corresponding connected components of corresponding input document image, as mentioned in Section 4.2.3.

TABLE 6.3: Performance values of text detection through CC overlapping in 96 multiscript maps using pixel based calculation

Modification through CC overlapping (Example shown in Figure 4.13)			
Proposed Approach	**Precision**	**Recall**	**Harmonic Mean**
LoG with 4 clusters - K-means	0.705361	0.712964	0.709142
LoG with 4 clusters - K-means++	0.711332	0.711178	0.711254
Gabor with 4 clusters - K-means++	0.778527	0.69505	0.734424
Gabor with 5 clusters - K-means++	0.693986	0.714049	0.703874

Conclusions about self-learning step
In the above tables, the best percentages are marked in boldface font. Accordingly, we conclude that till this experimentation LoG filter with 4 clusters K-means and K-means++ and Gabor filter with 4 or 5 clusters K-means++ are giving top results, either in precision or, recall or, harmonic mean, among all the variations of the method. Consequently, further experimentation of the approaches using classifier is compared using these 4 variations of

TABLE 6.4: Performance values of text detection through CC overlapping in 96 multiscript maps using Bounding Box based calculation

Modification through CC overlapping (Example shown in Figure 4.13)			
Proposed Approach	**Precision**	**Recall**	**Harmonic Mean**
LoG with 4 clusters - K-means	0.638314	0.364677	0.464168
LoG with 4 clusters - K-means++	0.650421	0.356901	0.460897
Gabor with 4 clusters - K-means++	0.623578	0.414922	0.49828
Gabor with 5 clusters - K-means++	0.651331	0.368407	0.470620

the initial proposed approaches. Among these 4 variations, we have reported in the following tables the top two results in tabular form, where connected components of the modified images, reported in Table 6.3 and Table 6.4, are incorporated with MLHMLPSP-LBP variant feature and tested through one class classification by self learning as shown in Figure 6.2. The feature incorporation is also compared among multiple rotation invariant features, i.e. FMT, GIST-ARP, Hu moment etc. Description of all the features are given in Appendix A.3. Correspondingly, MLHMLPSP-LBP variant feature outperformed among all, in either precision or, in recall or, in harmonic mean. Consequently, further experimentation of the approaches using classifier is compared using these two variations of the approaches proposed so far.

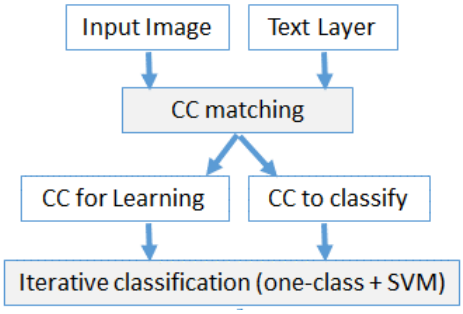


FIGURE 6.2: Description of the different steps of the evaluations

Results with the additional self-learning step (full)
After the first iteration done with the One-class classifier, the self-learning step is ended by using a second step of binary classification using a SVM. Table 6.5 and Table 6.6 are giving the pixel based and bounding box based results respectively of text image generation using this two-class SVM trained with the data already mentioned in Table 6.3 and Table 6.4.

Application of the proposed complete method to Architectural floor plan images
We have applied the proposed approach also to other grayscale graphical documents to see the applicability of the approach across different graphical documents. In this regard,

TABLE 6.5: Performance values of second level classified text detection in 96 multi script maps using pixel based calculation

Self Learning second iteration (mentioned in Figure 4.16) featured with MLHMLPSPLBP			
Proposed Approach	**Precision**	**Recall**	**Harmonic Mean**
LoG with 4 clusters - K-means	0.69909	0.812862	0.751695
LoG with 4 clusters - K-means++	0.710689	0.790896	0.748650

TABLE 6.6: Performance values of second level classified text detection in 96 multi script maps using Bounding Box based calculation

Self Learning second iteration (mentioned in Figure 4.16) featured with MLHMLPSPLBP			
Proposed Approach	**Precision**	**Recall**	**Harmonic Mean**
LoG with 4 clusters - K-means	0.63764	0.410929	0.499775
LoG with 4 clusters - K-means++	0.647493	0.394757	0.490481

we have collected few architectural floor plan images from Skysite [1]. Skysite is a cloud and mobile software solution available online that manages the workflow of construction documents. Using the qualitative analysis from the resulting images our text-graphics separation approach, we may conclude that the method is feasibly working for these kind of graphical documents. Moreover, the approach is efficient enough to detect normal text as well as detect text which are touching or intersected with other graphical components. For example, below figures are given as part of Figure 6.3 with one example input document along with their textual and graphical output.

1. `https://www.skysite.com/`

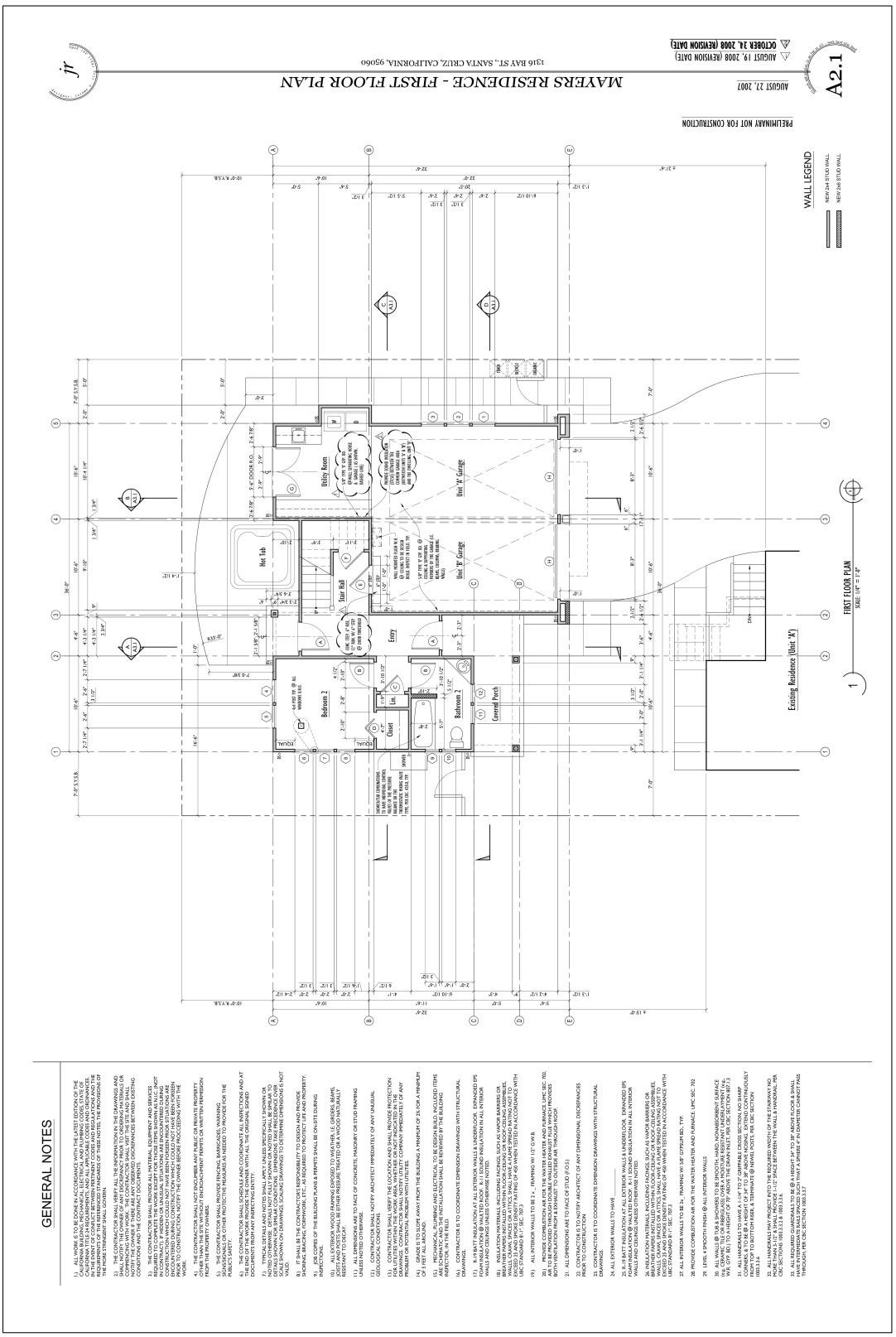


FIGURE 6.3: (a) A input Architectural Floor plan image

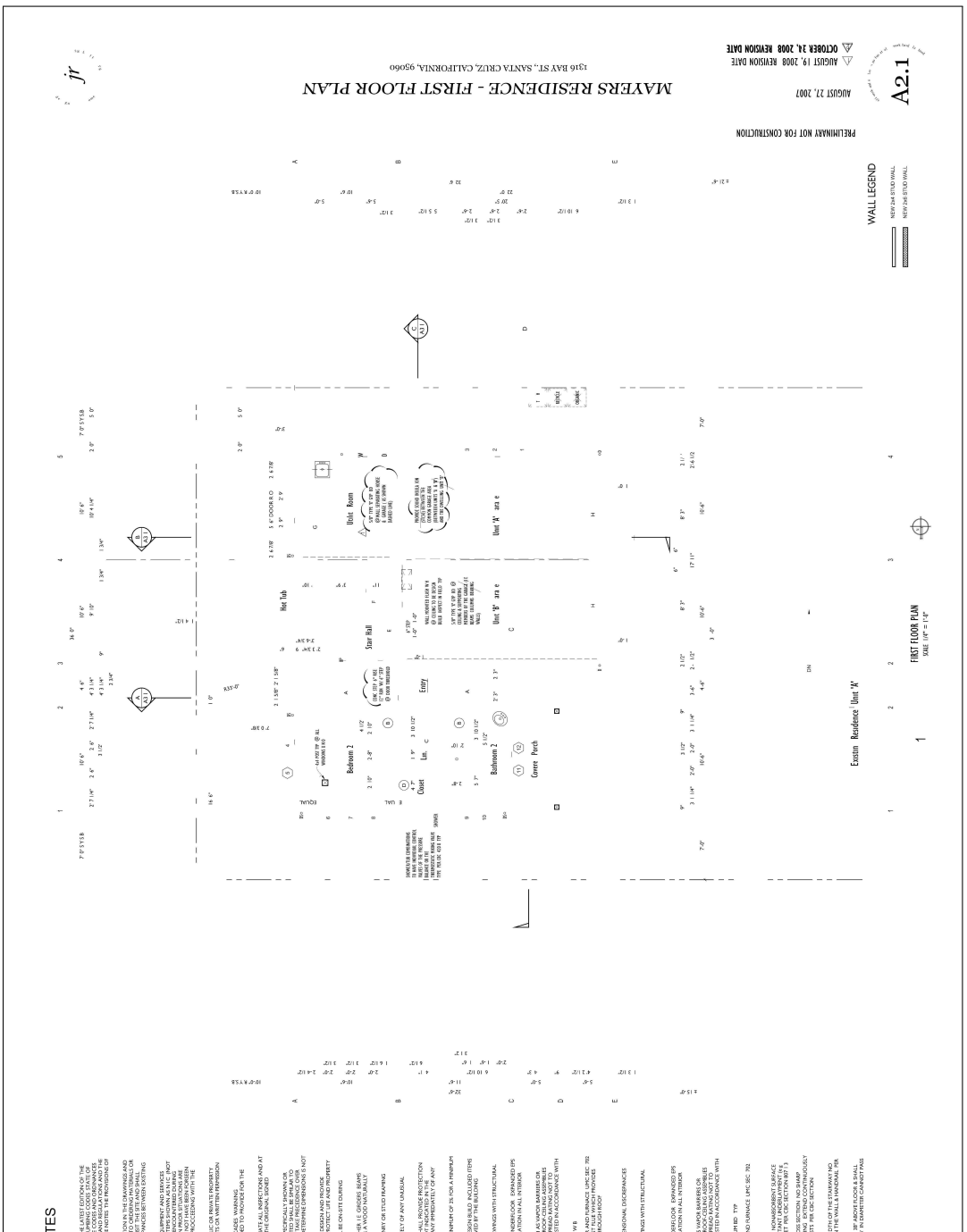


FIGURE 6.3: (b) Corresponding text layer

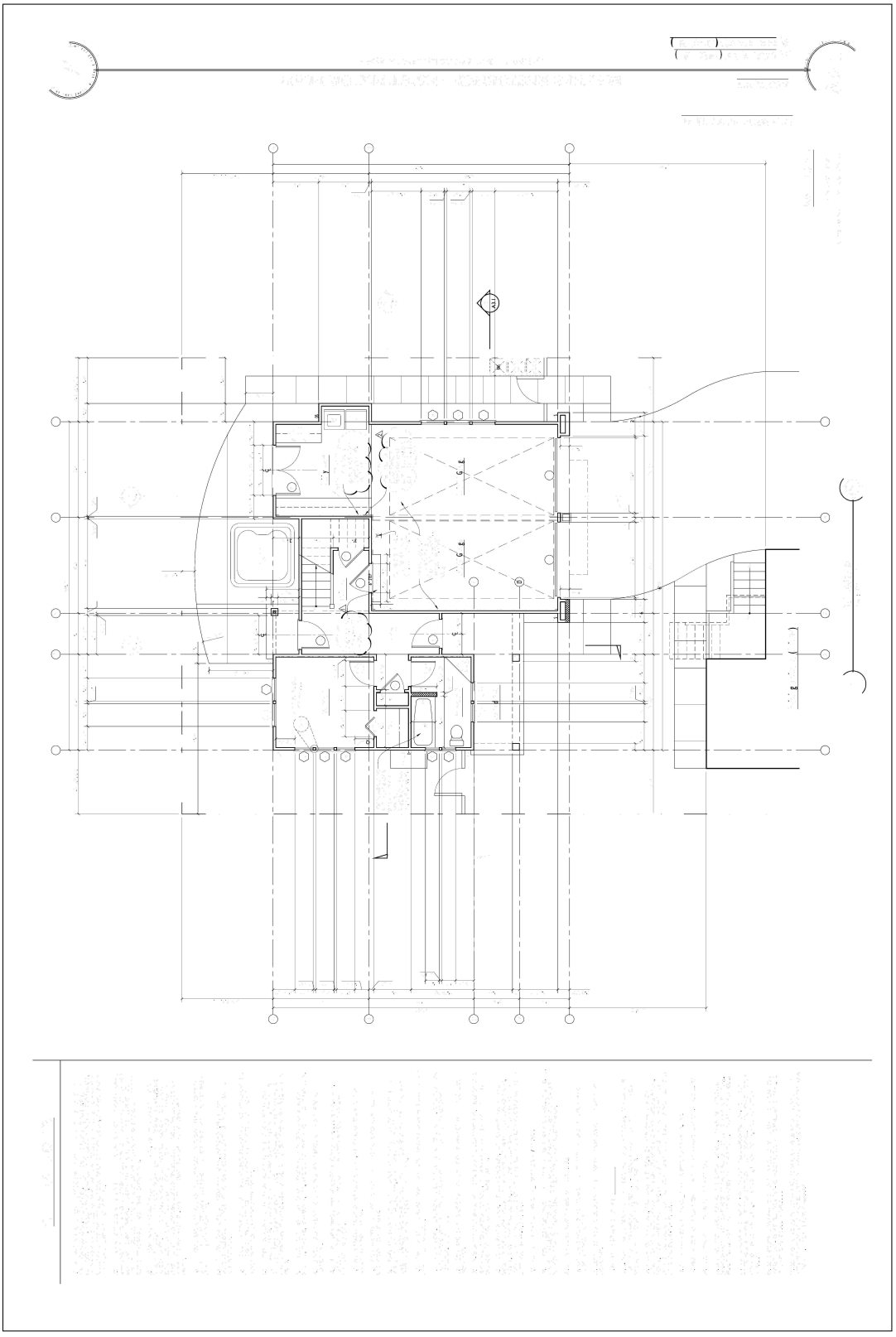


FIGURE 6.3: (c) Corresponding graphical layer

Final conclusion and decision to produce soft results (probability maps)
From above tables, we can conclude that the successive refinement steps make the recall increased but there is a decrease in precision. To overcome this problem, we have proposed the use of probabilistic information, where highly probable zones are considered to be definite text and low probable zones could be used in case of ambiguity. Table 6.7 and Table 6.8, show the pixel based and bounding box based results of the amount of data extracted to be text with their corresponding plausibility / confidence of being text. It is noticed that the components with $50-75\%$ confidence to be text actually contain many false positives. Also, fragmented data are more erroneous than component data. High probable text zone is good at both precision and recall.

TABLE 6.7: Performance values of probabilistic text detection in 96 multiscript maps using pixel based calculation

Text Image with probabilities (mentioned in Figure 4.18) featured with MLHMLPSP					
Proposed Approach	**Text type**	**Probability**	**Precision**	**Recall**	**Harmonic Mean**
Gabor with 4 clusters - K-means++	Components	>0.5, <=0.75	0.478685	0.060141	0.106857
LoG with 4 Clusters - K-means	Components	>0.5, <=0.75	0.405594	0.037081	0.06795
Gabor with 4 clusters - K-means++	Components	>0.75	0.728188	0.68820	0.707634
LoG with 4 Clusters - K-means	Components	>0.75	0.715953	0.765514	0.739904

TABLE 6.8: Performance values of probabilistic text detection in 96 multiscript maps using Bounding Box based calculation

Text Image with probabilities (mentioned in Figure 4.18) featured with MLHMLPSP					
Proposed Approach	**Text type**	**Probability**	**Precision**	**Recall**	**Harmonic Mean**
Gabor with 4 clusters - K-means++	Components	>0.5, <=0.75	0.509125	0.0533277	0.096543
LoG with 4 Clusters - K-means	Components	>0.5, <=0.75	0.535298	0.0608641	0.109301
Gabor with 4 clusters - K-means++	Components	>0.75	0.53516	0.196096	0.287021
LoG with 4 Clusters - K-means	Components	>0.75	0.652465	0.350962	0.456417

Comparative study with text-graphics segmentation methods
Even if our goal is not to produce another Text / Graphic separation technique, to provide an element of comparison between a / our (spotting) approach and a classical document retro-conversion system that will embed a text graphics separation and automatic transcription steps, we have generated performance values of one of the CCA based state of the art method proposed by [Tombre et al., 2002], mentioned in Section 2.3.1. on our dataset, Additionally, we have also implemented another proposed work of text extraction depends on Fourier-Laplacian filter [Shivakumara et al., 2011]. The corresponding performance values given in below Table 6.9 for both of pixel based and bounding box based calculation. **In every cases, we could see our method works far better than these works exist in the literature**.

TABLE 6.9: Pixel based and Bounding Box based performance values of QGAR method of text separation

QGAR Method					
Pixel Based			Bounding Box Based		
Precision	**Recall**	**Harmonic Mean**	**Precision**	**Recall**	**Harmonic Mean**
0.657907	0.576	0.6143413	0.229535	0.601114	0.332214

6.2.2 Structural level analysis

In addition to information coming from pixel intensity or texture information, it seems interesting to also take benefit of structural aspect of graphical and text elements in order to generate different types of indexes from the graphical document images. For that, an original vectorization technique is applied to the images and generate a set of structural primitives (vectors, quads, CCs, occlusions) to represent the content of the images. From all these structural primitives, numerical features can be computed to represent each foreground pixel of the images. A similar approach as before can then be used to classify pixels into different layers (Text, Graphics etc.) and also generate new probability maps. We have used a library made by Ramel et al. [Ramel et al., 2000] for image vectorization. To train the vector classifier, a set of 22 images were considered. The text and graphical components were manually separated to have two different layers of the images. As results, a set of 22 images containing only the graphical parts and a set of 1021 word images as textual parts were obtained. To summarize, we used a total number of 1043 images, including 22 graphical images, and 1021 word images for training the proposed text/graphic approaches. To train the SVM classifier, termination criteria, precision level, gamma, and C (cost) were respectively set to 3000 iterations, 10^{-9}, 10^{-5},and 10^{+5} by experiments. As for pixel level analysis, we evaluate the proposed approaches considering word level and pixel level accuracies. The results obtained from 6 test images at the word level are tabulated in shown in Table 6.10. These results were analyzed to get the accuracy based on the number of words detected by the probability map. Note that for text zone detection, we go through every occlusion during occlusion analysis. Resultantly, occlusion analysis consumes heavy computation, but it usually improves the text retrieval by almost 10%. It is worth mentioning that the proposed approaches achieved more than 91% word-level text extraction accuracy (Table 6.10). Our approach failed to successfully detect some words that were fully entwined with graphical pixels. One example of false detected word is shown in example in Figure 6.4. Here word detection is denoted by red color bounding box. We could see in the example that a portion of graphical component is detected as text here.

TABLE 6.10: Results of word extraction using the structural level features classification

Image sequence	**Ratio of spotted words**	**Percentage of detection**
1	42/43	97.7
2	50/56	89.3
3	59/62	95.2
4	48/67	71.5
5	119/121	98.3
6	54/57	94.7

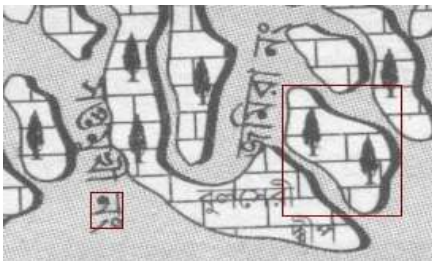


FIGURE 6.4: An example of falsely detected words

6.2.3 Lexical level analysis

At the lexical level, the proposed approach has to associate a (lexical) label to each character constituting an "non-ambiguous" detected text zone coming from the results provided by the Pixel and Structural Level analysis. In addition to the segmentation technique, for this step, we also need to define features to characterize the text zone and a classifier that will use these features to decide the label of the submitted text zone. For the learning and evaluation of SVM at character recognition level, we have considered four different sets of data consisting of multi-oriented and multi-scale text characters. One of the datasets is from graphical documents and its size is 2000 characters. The ground truth of this dataset has been generated manually from real maps for the performance evaluation. The other three datasets are synthetic data, constructed from Arial, Courier and Times New Roman font characters of font size ranging from 12 to 30. The size of each of these datasets is 5000. So, the total size is 17000 samples.

In our experiment both English uppercase and lowercase alpha-numeric characters are considered, so we should have 62 classes (26 for uppercase, 26 for lowercase and 10 for digits). But because of shape similarity due to orientation of some of the characters like 'd' and 'p' ; 'b' and 'q' ; etc. these are grouped together. Hence, in our approach we considered 40 classes of character shapes. Different fonts of characters including Times New Roman, Courier and Arial have been used to learn the SVM classifier for the experiment. A comparison has been done with different rotation invariant feature descriptors used in the literature, namely : ART, HU, Zernike moments, Fourier-Mellin, variant of LBP and

GIST-ARP. The SVM evaluation is done with 5-fold cross-validation over the set of 17000 samples given above shown in Table 6.11. So we have trained our SVM classifier with 17000 isolated characters including real and ideal data featured with Fourier-Mellin. The SVM classification is done with 5-fold cross-validation. Some confusions are observed between 'z' and '2', 'S' and '5', 'D' and '0', 'b' and 'h', 't' and '1'. We have used following parameter in SVM [Chang and Lin, 2011] : svm type - C-SVC, kernel type - radial basis function (RBF), gamma - 4.92458, cost - 65536, probability estimates - 1. All other parameters used are default ones.

As per the results reported in Table 6.11, we draw some observations. The recognition rate is high during 5-fold cross-validation of the data consist of mostly synthetic data made of 17000 samples already discussed in 3.2. Here, we got **variant of LBP, GIST-ARP and FMT, all are working as suitable rotation invariants feature which shows more than** 90% **recognition rate** during this cross-validation. Moreover, when we perform 5-fold cross validation of data collected from geographical maps, the recognition rate is feasible enough in case of GIST-ARP and LBP-variant feature. On the other hand, there is a decrease in recognition accuracies when we test the data from real geographical maps instead of cross-validating them. The reason might be, over fitting of the training model. Moreover, we have seen FMT is not working well at all for real data collected from geographical map, both for cross-validation and testing. The reason might be its overfitting to the training model and also its sensitivity to presence of noise in real map characters. Unlike other features, Hu moment has showed consistency throughout the testing and validations and even though containing low accuracy, in case of testing real data collected from geographical map, this feature is outperformed.

Table 6.11: Comparative result of character recognition

LBP - variant		GIST-ARP	Hu Moment	FMT	Zernike Moment
SVMKernel - RBF	SVMKernel - Linear	SVMKernel - RBF	SVMKernel - RBF	SVMKernel - RBF	SVMKernel - RBF
Bengali Synthetic DataSet - Cross Validation					
14.16%	**95.77%**	94.51%	54.03%	90.13%	77.82%
English Synthetic DataSet - Cross Validation					
88.71%	21.41%	95.40%	10.24%	**96.70%**	94.72%
English Dataset2(collected from map) - Cross Validation					
13.58%	**81.48%**	78.68%	48.37%	63.19%	51.80%
English Dataset2(collected from map) - Testing using Training model of English Synthetic DataSet					
2.17%	19.15%	23.22%	**36.34%**	1.07%	6.58%

Geographical maps may contain text of multiple font sizes. However, an individual word belongs to a map contain characters of same font size. To obtain a prior idea in word level, grouping of isolated characters according to their sizes is achieved by clustering the characters according to their sizes. In this regard, both K-means and Fuzzy C-means clustering were considered and depending on their performance the better one was chosen to group the characters for further ease of access in word level for our integrated system

shown in Table 6.12.

TABLE 6.12: Clustering Isolated Characters Based On Size

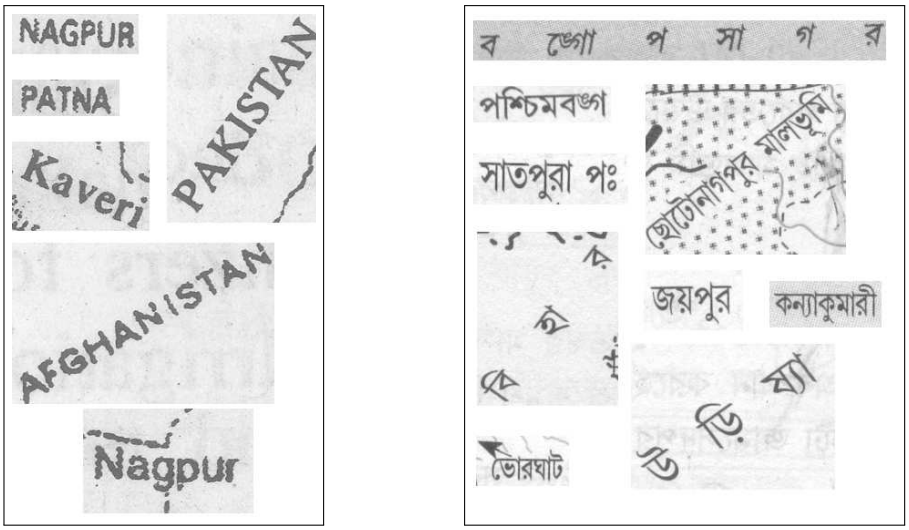
Successful Clustering in %	
K-means	**Fuzzy C-means**
93.4	97.2

6.3 Evaluation of the incremental spotting/retrieval system

For spotting itself, we have considered 30 different real graphical maps and 20 different query words to test our method. As there are very few proper grayscale maps available, for experimental purpose, we have used our published dataset that are available with annotation. There are approximately $35 - 40$ words in each of the documents. Documents are digitized in grey tone at 300 dpi and we have used Otsu binarization technique for their two tones conversion. QGAR library is used to extract characters from documents. Proper evaluation of the whole system depends on validation of each part of the system separately viz. accuracy of character segmentation and accuracy of the character recognition etc.

6.3.1 Few illustrations of the initial spotting

In Figure 6.5, we have shown few cases where our method is able to spot query word properly despite its touching between graphical components and character components. In Figure 6.6, we have shown few cases where our method is not able to locate those words at all because of its high association with noise or other graphical components.



(a) (b)

FIGURE 6.5: Examples of few words successfully spotted through our system (a) English (Roman) words, (b) Bangla words.

FIGURE 6.6: Examples of few words which are not successfully spotted through our system

6.3.2 Illustrations of the full spotting system

In Figure 6.7, we have shown few cases in English and Bangla documents where our method is able to spot query word properly despite its touching with graphical components or character components using incremental approaches.

In Figure 6.8, we have shown few cases where our method is not able to spot query word properly because of the poor resolution of the image and presence of noise in the image.

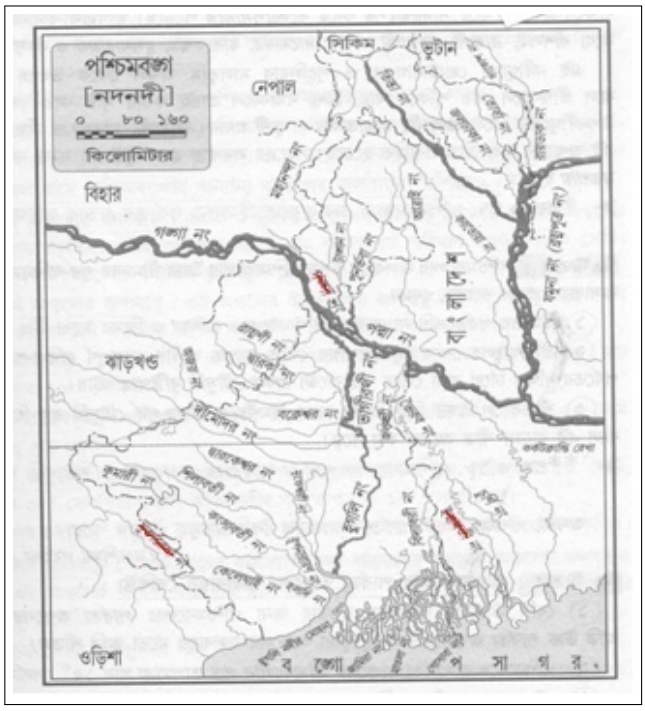
6.3.3 Quantitative evaluation

To evaluate precisely the performance of the system, we have used precision (P) and recall (R) for evaluation of spotting of candidate words. Each spotted word images is considered as relevant or not, depending on the ground truth of the data. For a given spotted result, the precision measure P is defined as the ratio between the number of relevant retrieved items and the number of retrieved items. The recall R is defined as the ratio between numbers of relevant retrieved items to the total number of relevant items in the collection. For spotting experiment itself, we have considered 30 different real graphical maps, 20 different query words for English, 10 different real graphical maps and 10 different query words for Bangla to test our method. Result are provided in Table 6.13 respectively with fixed 6 nearest neighbor analysis.

TABLE 6.13: Word Spotting Results in graphical documents

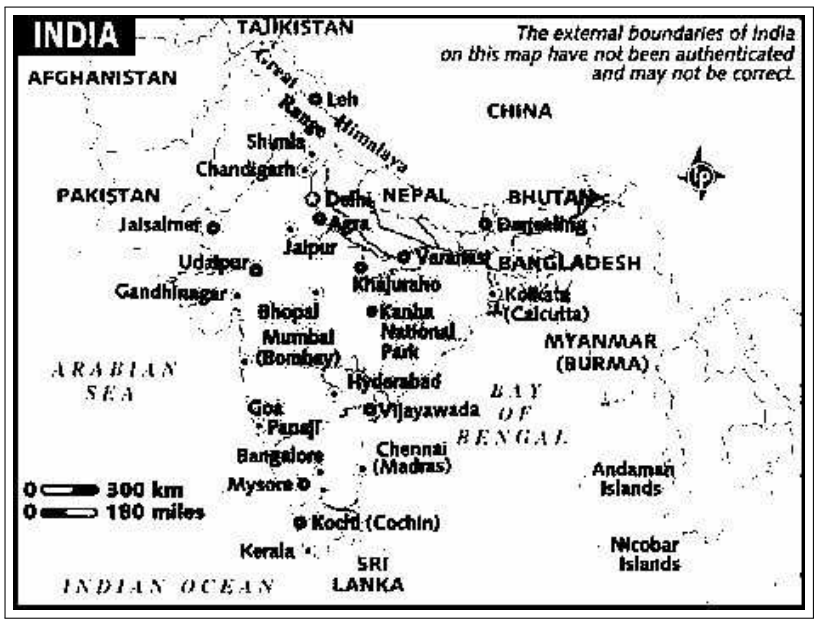
	Precision	**Recall**
Bangla	69.9	72.4
English	91.1	94.2

(a)

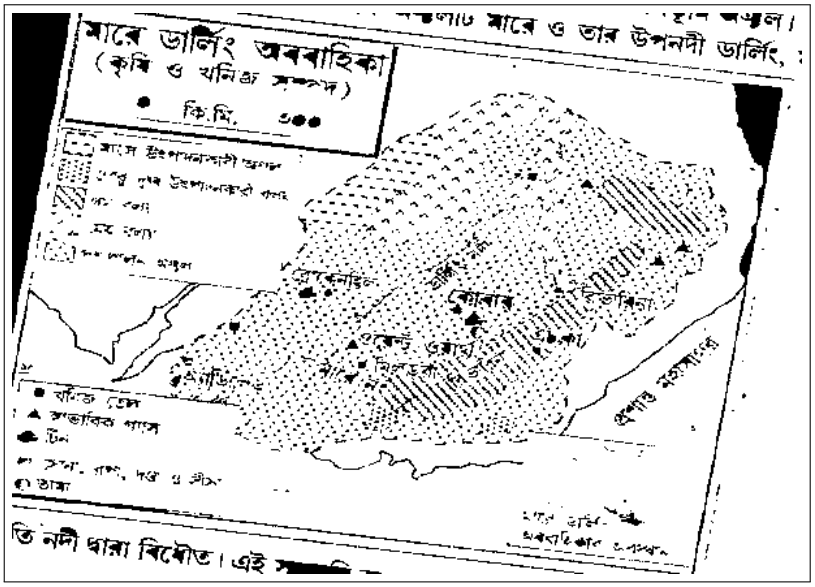


(b)

FIGURE 6.7: (a) Examples of an English graphical document where our system could spot for the query words "Italia" and "Romania" (red line is shown our spotted words), (b) Examples of a Bengali graphical document where our system could spot for the query words "গঙ্গা", " ইছামতী " and "সুবর্ণরেখা".



(a)



(b)

FIGURE 6.8: (a) Examples of a graphical document where our system could not spot for the query word "Udaipur", "Mysore" and "BHUTAN", (b) Examples of a Bengali graphical document where our system could not spot for the query words "ব্রোকেনহিল "and "গম" because of their poor quality.

Considering the evaluation of the spotting system, we can mention that our system is not giving any false positive in case of query word of length higher longer than 3, but we are missing few candidate words, especially when in graphical documents texts are not so smooth and they are much noisy. Each spotted word image is considered as relevant or

not, depending on the ground truth of the data.

6.3.4 Summary

In this section, we have demonstrated the advantages of our proposed work related to multi-level indexing of graphical documents. Regarding text-graphics separation, from the performance values, we could see that our method is working feasibly to solve the problem of text-graphics separation in grayscale document images. At the Pixel-level, we have also provided few comparative data to show that our system is working robustly for this problem better than other existing approaches. At the structural level, we have approached the same problem of text-graphics separation but from different perspective by applying the vectorization into component level. Due to highly intertwined between textual and graphical components, a single binary decision is not enough to decide about the presence of a text component. At the lexical level, we have processed the image information into more semantic manner i.e. textual images are segmented and labeled into known character levels. Moreover, those characters are clustered to be denoted as word. Finally, a combination of multi-level decisions in terms of probabilistic decision is proposed to make a full-grown indexing system. The low probable components can be used with the support of some neighborhood information during the retrieval step.

Chapitre 7

Conclusion and Perspectives

"A conclusion is simply the place where you got tired of thinking."
- Dan Chaon

This thesis deals with word spotting in graphical documents, specifically, word spotting in geographical documents. Geographical documents make our problem domain challenging and interesting for its various aspects. For example, geographical documents contain text which has multiple behaviors e.g. multi-orientation, multi-scaled, variable character spacing, characters are connected or over-connected or broken etc. On the other hand, graphical components in such documents may be of various types i.e. long or thin lines, short symbol, dashed lines, hatched area, textures, small curves etc.

Furthermore, presence of multi-oriented characters, connections between characters, intersections of texts and symbols with graphical elements, etc., are common in such documents. Consequently, word spotting in these documents proves to be a challenging task. We have focused on this area and worked for the proposition of a robust word spotting system dedicated to multi-oriented and multi-scaled text that can occur in geographical document images.

The originality of the proposed framework comes from the following ideas :

1. We propose to generate and use at the same time pixel, lexical and structural level information in order to better index the heterogeneous content of a graphical documents. Pixel level analysis is performed by combining texture features and connected components features submitted to different (self-learned) classifiers to generate soft decisions (probability maps) about the presence of text or graphics instead of doing a classical text segmentation (text/graphics separation) as in classical line drawing retro-conversion systems. Then, we can say that the proposed architecture corresponds to a segmentation-free spotting system rather than a segmentation based spotting system. We propose to extract and compute structural signatures with higher level information than only classical low level image features (like key points, shape descriptor etc.) to index some selected parts in the graphical documents. In the proposed system, lexical information coming from a multi-scripts, multi-oriented

and multi-scaled OCR engine based on water reservoir technique is also used for the computation of more semantical indexes.

2. Another strong objective was also to construct soft or probabilistic indexes during the off-line indexation step. The use of machine learning approaches during the generation of the signatures describing the Regions of Interest detected inside the images allows to incorporate these soft and probabilistic information (that we will call "probability maps") in addition to the multi-level signatures. Then, instead of storing binary decisions concerning the appurtenance of a pixel to a specific Regions of Interest (ROI) or information layer (text, graphical symbol, hatched area etc.), a set of probability maps can be computed individually or conjointly and can then constitute one part of the indexes that we will use during the querying step (on-line).All these different types of signatures and associated probability maps have to be merged in a unified representation of the images content and then, this multi-level characterization constitutes a new way to compare the content of an image with the content of a query (Image or String).
3. During the retrieval step, we also propose an incremental retrieval step. With the help of probability maps, an initial spotting of some parts of the queries becomes possible by selecting regions having high probability scores for a specific type of signature (visual or structural ones). These ROI could then be selected as seeds for a deeper analysis using complementary features (SIFT key-points). Incidentally, misrecognition can occur due to the structural complexity of graphical documents as well as due to the presence of multiple connected components. In such scenario, the intrinsic characteristics (position, size and orientation etc.) of the partially spotted regions matched with one part of the query could be a very useful source of knowledge to realize the final and entire spotting of the query (or for the rejection of a false alarm). This incremental spotting process using probability maps is proposed to cope with variability and heterogeneity present in graphical documents as well as noise and degradations. The probability values and the complementary features are used to better estimate the final regions containing the missing parts or just to confirm about the pertinence of the candidate regions for possible spotting.With the help of probability maps, we can choose the elements having good recognition score through classifier. Some of the query parts might not be spotted during initial spotting, due to misrecognition in classifier. Incidentally, misrecognition occur due to the structural complexity of graphical documents as well over connectivity between components. In such scenario, position, size and orientation of the partially spotted query are noted. Thereafter, using those parameters we estimate regions (candidate regions) of missing query parts.

We have performed experimentation on the dataset, both on English and Bangla scripted scanned maps. The experimental results demonstrate that :

- ✍ The proposed multi-level indexing scheme is very efficient and can even outperform state of the art Text/Graphic separation methods dedicated to various types of graphical document.
- ✍ The proposed architecture allow to spot multi-oriented and multi-scale and multi-script words present in graphical documents (geographical maps) as well from a query

string than from query image (even if a more precise evaluation of the performance is still needed).

Finally, we can mention our contribution to the research community concerning the topic of performance evaluation. To overcome the lack of ground truthed data, we have built a publicly available dataset of graphical documents to allow comparison between our system and future propositions in several languages. Additionally, for ease of access and for future use, we have developed one software solution to ground truth the data.

Limitations and future works :

- ✍ As mentioned previously, we know that the main limitation of this work is the lack of experimental results concerning the full spotting system. Qualitative results showing the performance of our full spotting system on Latin and Indian maps have been provided, but not enough images and associated queries have been used to compute the provided precision and recall rates presented in Section 6.3. Significant supplementary experiments have to be done, in a short future, to definitively validate the interest of the proposed approaches.
- ✍ A deeper comparative study is desirable at every milestone of the work with the help of public dataset both in the perspective of outcome and time complexity. Also, applicability or reusability of our proposed work should be thought of by crawling through specific real life issues regarding every kind of digitized documents that contain some amount of text and/or some amount of graphical content. Then, application areas may be dispersed from geographical documents to all the graphical documents like floor plans and other engineering drawings.
- ✍ During retrieval, we have relied upon SIFT matching to localize missing character for the completion of a query retrieval where query characters are partially recognized.SIFT can match in between two key points of candidate and template. But, to conclude about the presence of a character, by matching of multiple pairs of key points between candidate and template, SIFT is not capable of considering the relative position of the key points in the image. To counter this problem, the use of a filtering technique such as RANSAC [Fischler and Bolles, 1981] after SIFT may help for improving the system performance.
- ✍ The current system is dedicated only to textual and graphical layer but it should be possible to identify and index more than two layers like, for example,a textual layer, the filled shapes, the dotted lines, the hatched areas with more semantically meaningful for better results etc. The Figure 7.1 illustrates the possible semantic layers that can be found in geographical maps.

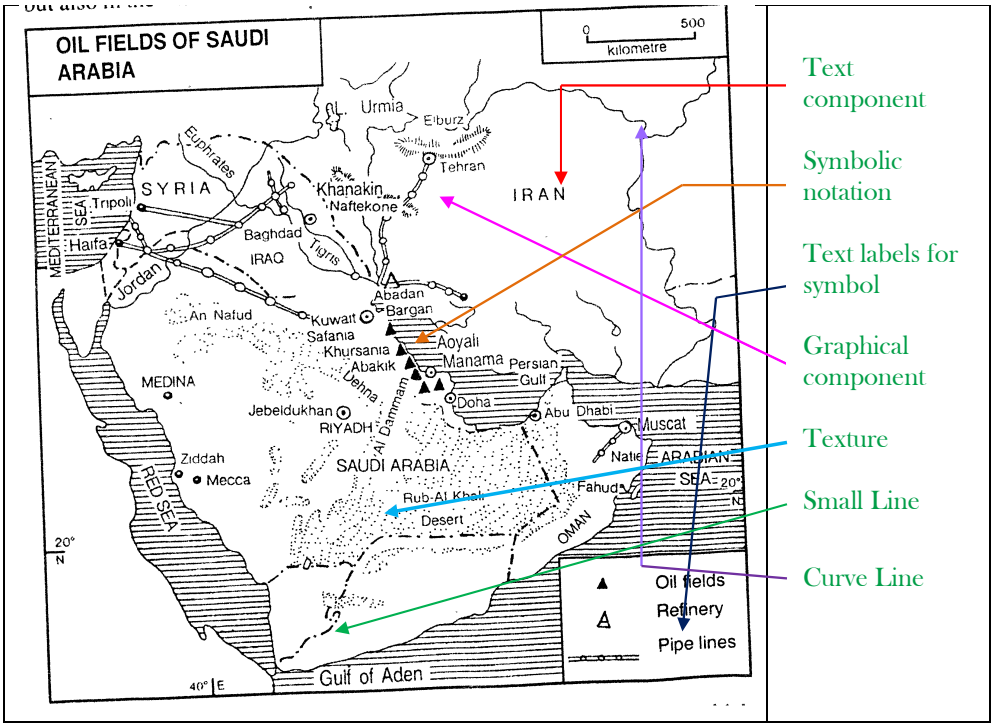


FIGURE 7.1: A sample map with denoted text, symbolic notation, text label for specific symbol, graphical component, texture, small line and curve line

- ✍ Then, we may imagine that the user can combine different items of graphical documents to create a query. For example, query can be parts of roads, or a combination of different graphical symbols, or it can be done by example etc. Such query can be handled in future.
- ✍ During text separation approach, we have proposed the concept of probability map, but this has not been exploited to the extreme. In the context, as discussed in above points we can make use of probability map to achieve a fuzzy separation between the semantic information, which is subject to modification depends on the context in the consequent stages.

Annexe A

Summary of Literature Used in Proposed Works

Abstract

Here, we have recalls few existing theories which are related to our framework described in the thesis. Related to our filter based approach of text-graphics separation, here we have discussed the theory of Gabor filter and K-means, K-means++ clustering. Moreover, we have discussed about few existing Rotation invariant features those are used for our text-graphics recognition for their classification or detail characters recognition.

A.1 Band Pass Filter

A.1.1 Gabor filter

Dennis Gabor proposed Gabor filter in 1946. The purpose of this filter imagined to be similar like human eye, which can distinguish different textures from images. From a mathematical point of view, a $2-D$ Gabor filter [Shen et al., 2007] is a Gaussian kernel function modulated by a sinusoidal plane wave as shown in Figure A.1. The frequency and orientation representations of Gabor filters are similar to those of the human visual system, and these have been found to be particularly appropriate for texture representation and discrimination. In connection with the point, the filter finds an abrupt transition (background to foreground) in an image and some fuzzy idea of where the transition is localized. Based on all these aspects, we have assumed that it may result into a system to deal with text segmentation problems well.

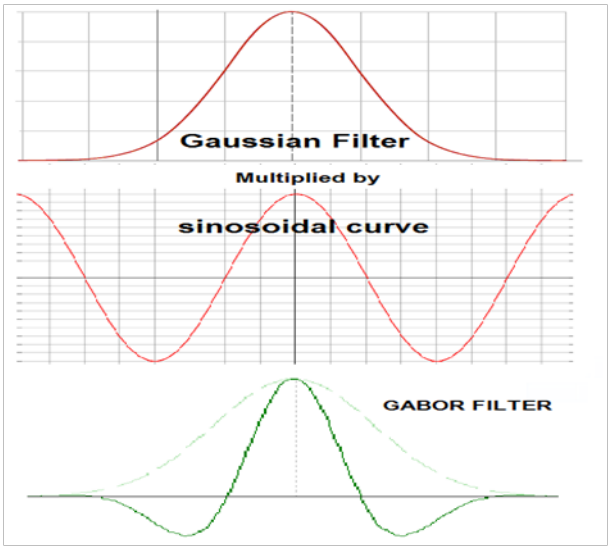


FIGURE A.1: Pictorial view of Gaussian and Sine wave convolution as Gabor Filter

The equation of Gabor Filter can be written as :

$$\mathtt{h}(\mathtt{x},\mathtt{y}) = \mathtt{s}(\mathtt{x},\mathtt{y})\mathtt{g}(\mathtt{x},\mathtt{y}) \tag{A.1}$$

where, $\mathtt{s}(\mathtt{x},\mathtt{y})$ - sinusoidal plane wave, $\mathtt{g}(\mathtt{x},\mathtt{y})$ - Gaussian kernel function

Gaussian function equation can be written as,

$$\mathtt{g}(\mathtt{x},\mathtt{y}) = \mathbb{A}\exp(-(\frac{(\mathtt{x}-\mathtt{x}_o)^2}{2\sigma_{\mathtt{x}}^2} + \frac{(\mathtt{y}-\mathtt{y}_o)^2}{2\sigma_{\mathtt{y}}^2}) \tag{A.2}$$

Here, the coefficient $\mathbb{A}$ is the amplitude of the Gaussian curve, $\mathtt{x}_o$, $\mathtt{y}_o$ is the centre of the curve and $\sigma_{\mathtt{x}}$, $\sigma_{\mathtt{y}}$ are the $\mathtt{x}$ and $\mathtt{y}$ spreads of the curve.

To derive simpler equation, if we assume $\mathbb{A} = 1, \mathtt{x}_o = y_o = 0, \sigma_{\mathtt{x}} = \sigma$ and $\sigma_{\mathtt{y}} = \sigma_{\mathtt{x}} \times \gamma$ (aspect ratio). Then Gaussian equation can be rewritten as,

$$\mathtt{g}(\mathtt{x}, \mathtt{y}) = \exp(-((\frac{\mathtt{x}^2 + \gamma^2\mathtt{y}^2}{2\sigma^2})) \tag{A.3}$$

Along with multiple orientations, Gaussian equation can be rewritten as,

$$\mathtt{g}(\mathtt{x}, \mathtt{y}) = \exp(-((\frac{\mathtt{x}'^2 + \gamma^2\mathtt{y}'^2}{2\sigma^2})) \tag{A.4}$$

where, $\mathtt{x}' = \mathtt{x}\cos\theta + \mathtt{y}\sin\theta, \mathtt{y}' = \mathtt{y}\cos\theta - \mathtt{x}\sin\theta$.

Next, sinusoidal plane wave equation can be written as,

$$\mathtt{s}(\mathtt{x}, \mathtt{y}) = \cos(2\pi\frac{x'}{\lambda} + \Psi) \tag{A.5}$$

where, $\mathtt{x}' = \mathtt{x}\cos\theta + \mathtt{y}\sin\theta$, Ψ - phase offset, if we assume $\Psi = 0^o or, 90^o$,

Then,$\mathtt{s}(\mathtt{x}, \mathtt{y}) = \cos(2\pi\frac{\mathtt{x}'}{\lambda})$ or, $-\sin(2\pi\frac{\mathtt{x}'}{\lambda})$ respectively.

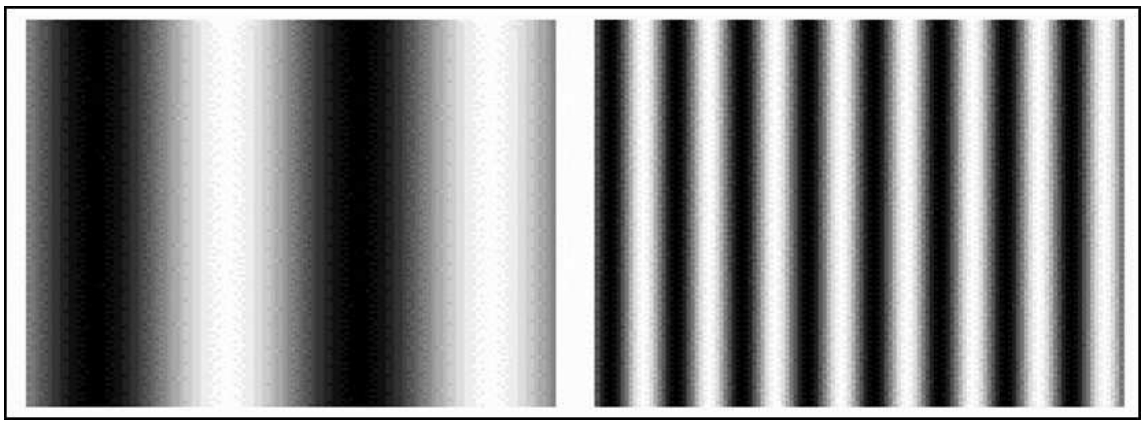
We have considered the fact that a Gabor filter with a specific orientation gives a strong response for locations that have structures in the given orientation. Consequently, for multi-oriented text detection, we use multi-oriented Gabor filter [Jain et al., 1999].

Resultantly, along with multiple orientations, finally Gabor filter of our use can be re-written as,

$$\mathtt{g}(\mathtt{x}, \mathtt{y}; \lambda, \theta, \Psi, \omega, \gamma) = \exp(-\frac{\mathtt{x}'^2 + \gamma^2\mathtt{y}'2}{2\sigma^2})\cos(2\pi\frac{\mathtt{x}'}{\lambda} + \Psi) \tag{A.6}$$

where, $\mathtt{x}' = \mathtt{x}cos\theta + \mathtt{y}sin\theta, \mathtt{y}' = \mathtt{y}cos\theta - \mathtt{x}sin\theta$.

The ratio $\frac{\sigma}{\lambda}$ (standard deviation/wavelength) determines the spatial frequency bandwidth and thus the number of parallel excitatory and inhibitory stripe zones which can be observed in their receptive fields. Below images from Figure A.2 shows example of images having low frequency and high frequency bandwidth respectively.



(a) (b)

FIGURE A.2: (a) Low frequency and (b) high frequency bandwidth image

The $\mathtt{b}$ (in octaves) and the ratio $\frac{\sigma}{\lambda}$ are related as follows :

$$\mathtt{b} = \log_2 \frac{\frac{\sigma}{\lambda}\pi + \sqrt{\frac{\ln 2}{2}}}{\frac{\sigma}{\lambda}\pi - \sqrt{\frac{\ln 2}{2}}}, \frac{\sigma}{\lambda} = \frac{1}{\pi}\sqrt{\frac{\ln 2}{2}} \cdot \frac{2^{\mathtt{b}}+1}{2^{\mathtt{b}}-1} \tag{A.7}$$

λ is the wavelength of the cosine factor of the Gabor filter kernel and herewith the preferred wavelength of this filter. We have used and specified the wavelength value in terms of pixels count. In order to prevent the occurrence of undesired effects at the image borders, the wavelength value [Hol et al., 2008] should be smaller than one fifth of the input image size.

A.2 Clustering

A.2.1 K-means and K-Means++

KMeans is a clustering algorithm based on iterative relocation that partitions a dataset into K clusters. The partitioning is done by locally minimizing the average squared distance between the data points and the cluster centers. K is disjoint subsets of $\mathtt{X}$, whose union is $\mathtt{X}$. For a set of data points $\mathtt{X} = \{\mathtt{x}_1; ...; \mathtt{x}_N\}, \mathtt{x}_i \in \mathbb{R}_{\mathtt{d}}$, the KMeans algorithm creates a K-partitioning $\{\mathtt{X}_l\}_{l=1}^{K}$ of $\mathtt{X}$ so that if $\{\mu_1; ...; \mu_K\}$ represent the K partition centers, then the following objective function

$$\jmath_{kmeans} = \sum_{l=1}^{K} \sum_{\mathtt{x}_i \in \mathtt{X}_l} \|\mathtt{x}_i - \mu_l\|^2 \tag{A.8}$$

is locally minimized. In case of simple K-means, the choice of initial cluster centers are random, whereas, in case of K-means++, it uses an heuristic to find centroid seeds for K-

means clustering. The choice of initial cluster somehow affect the final clustering solution. In case of K-means++, it select the first cluster center, $\mathtt{c}_1$ randomly. If we denote the distance between $\mathtt{j}$-th cluster center and data point, m as $\mathtt{d}(\mathtt{x}_\mathtt{m}, \mathtt{c}_\mathtt{j})$, then K-means++ will select centroid $\mathtt{j}$ at random from $\mathtt{X}$ with probability

$$\frac{\mathtt{d}^2(\mathtt{x}_\mathtt{m}, \mathtt{c}_\mathtt{p})}{\sum\limits_{\{h;\mathtt{x}_\mathtt{h}\in\mathtt{c}_\mathtt{p}\}} \mathtt{d}^2(\mathtt{x}_\mathtt{h}, \mathtt{c}_\mathtt{p})} \tag{A.9}$$

A.3 Scale and Rotation Invariant Features

A.3.1 Multi-Level histograms of multi-scale local binary pattern with spatial pyramid (MLHMLPSP)

In 2007, Liao et al. proposed this [Liao et al., 2007] variation of LBP feature. The local binary pattern (LBP) [Ojala et al., 2002] operator is highly suitable for texture description. LBP labels the image pixels by thresholding the 3×3-neighborhood of each pixel with the center value and summing the thresholded values weighted by powers of two. The thresholding opration is done by checking that when the center pixel's gray intensity value is greater than its neighbor's intensity value, LBP take 0 against that neighbor pixel. Otherwise, it takes 1 for that pixel. Likewise, for 8 neighbors, an 8-digit binary number is formed. The number is usually converted to decimal (i.e. any value within 0 to 255) for convenience. The frequency of each of the decimal number occurrences i.e. histogram is computed. This histogram is a 256-dimensional feature vector as range of decimal values are from 0 to 255. These feature vectors can be fed to the classifier to classify images.

LBP has several variations and advancements [Huang et al., 2004] [Liao et al., 2009] [Heikkilä et al., 2006]. *Multi-Level Histograms of Multi-Scale LBP* is one of these variations that proposes a multi-scale characterization of an image content. Depending on the considered number of scales and scaling factors, LBP feature vector is formed using 3×3-neighborhood and/or 4×4 neighborhood and/or 5×5-neighborhood and so on. In Figure A.3, we have given an example of the spatial pyramid concept.

We have selected this method for the Text CC identification. In our experiment, we have taken 4 scales. Further, the level of spatial pyramid is 2 and overlapping between the pyramids of level 2 is $\frac{1}{2}$. So, total number of pyramids are 1 for level 1 and 9 for level 2 i.e. 10. Consequently, the feature vector length of this variation of LBP with parameter settings, would be $256 * 4 * 10 = 10240$.

A.3.2 Fourier-Mellin Transform (FMT)

As symbols in graphical documents can have any orientation and size, the consequence is that we should desire a recognition system invariant with regard to rotation and scaling of a pattern. Fourier-Mellin transform (FMT) [Adam et al., 2000] corresponds to the decomposition of the pattern into circular (provided by the Fourier transform) and radial

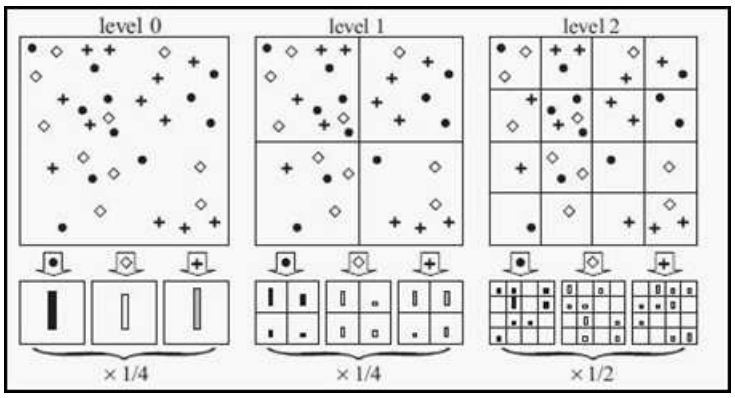


FIGURE A.3: Example of constructing a three-level pyramid. The image has three feature types, indicated by circles, diamonds, and crosses. At the top, we subdivide the image at three different levels of resolution. Next, for each level of resolution and each channel, we count the features that fall in each spatial bin. Finally, we weight each spatial histogram by LBP. (Figure credit : [Lazebnik et al., 2006])

(provided by the Mellin transform) harmonics which are both rotation and scale invariant. So, we have also selected this method for CC classification purpose.

The Fourier-Mellin transform is a helpful mathematical tool for image recognition because it is invariant in rotation, translation and scale. The Fourier Transform (FT) is translation invariant and its conversion to log-polar coordinates converts the scale and rotation differences to vertical and horizontal offsets that can be measured. A second FFT, called the Mellin transform (MT) gives a transform-space image that is invariant to translation, rotation and scale. Let f denote a function representing a gray-level image. The standard Fourier-Mellin transform of f [Adam et al., 2000] is given by :

$$M_f(v,q) = \int_{\rho=0}^{+\infty} \int_{\theta=0}^{2\pi} f(\rho,\theta)\rho^{-iv} \exp(-iq\theta) d\theta \frac{d\rho}{\rho}, \tag{A.10}$$

with $q \in Z, \theta \in R$, Experimentally, Fourier-Mellin feature and GIST-ARP feature outperformed among all these different rotations invariant shape descriptors for both of the Bengali and English character recognition and we have reported a comparative study in our experimental section for our work of isolated characters' recognition.

A.3.3 GIST Angular Radial Partitioning (GIST-ARP)

As mentioned by Liu et al. [Liu et al., 2012], an object can be passed through a cascade of Gabor filters in S scales with O orientations at each scale. Each of the resulting images is then divided into an N-by-N grid. Within each block on the grid, further partition is done into A bins using Angular Radial Partitioning (ARP) [Chalechale et al., 2004]. Then, the average intensity level is calculated in each angular bin, followed by a $1-D$ discrete Fourier transform on the angular bins in each block and then taking the magnitude of

the coefficients to achieve positional invariance. Finally, the feature vector is obtained by concatenating all the DFT transformed bins in the image across all the orientations and scales, resulting in an $S \times O \times N \times N \times A$ dimensional feature vector.

A.3.4 Angular Radial Transform (ART)

REFAngular radial transform (ART) gives a minimized and effective approach to express pixel dispersion inside a $2-D$ image object region. The ART coefficients [Ricard et al., 2005], F_{nm} of order n and m, are defined by,

$$F_{nm} = \int_0^{2\pi} \int_0^1 V_{nm}(\rho,\theta) f(\rho,\theta) \rho d\rho d\theta \tag{A.11}$$

where $f(\rho,\theta)$ is an image function in polar coordinates and $V_{nm}(\rho,\theta)$ is the ART basis function that is separable along the angular and radial directions :

$$V_{nm}(\rho,\theta) = A_m(\theta) R_n(\rho) \tag{A.12}$$

where, $\begin{cases} A_m(\theta) = \frac{1}{2\pi} exp(jm\theta) \\ R_n(\rho) = \begin{cases} 1, \text{ if } n = 0, \\ 2cos(\pi n \rho), \text{ if } n \neq 0 \end{cases} \end{cases}$

A.3.5 Hu's moments

An image moment is a definite particular weighted normal (moment) of the pixels' intensities, or an element of such moments, generally preferred for several eye-catching properties or interpretations. They are well-known for their application in image analysis and used to derive invariants with respect to specific image transformation. Recognition of alphabetical characters independently of position, size and orientation can be accomplished through such moment invariants. Hu [Hu, 1962] introduced seven nonlinear functions defined on regular image moments. For image I, the raw moment [Zhang et al., 2015b] of order $(m+n)$ is defined as,

$$M_{mn} = \sum_x \sum_y x^m y^n I(x,y) \tag{A.13}$$

where $m, n = 0, 1, 2, ...,$ and (x, y) the pixel position. The central moments μ are usually used in real applications to replace the raw moment in Eq. A.14.

$$\mu_{mn} = \sum_x \sum_y (x-\overline{x})^m (y-\overline{y})^n I(x,y) \tag{A.14}$$

where, $\overline{x} = \frac{M_{10}}{M_{00}}, \overline{y} = \frac{M_{01}}{M_{00}}$ The central moments are translational-invariant under this definition. Central moments can be extended to be both translational and scale invariant, by being divided by the scaled (00)-th moment. The results are called normalized central moment.

$$\eta_{mn} = \frac{\mu_{mn}}{\mu_{00}^{(m+n)/2+1}} \tag{A.15}$$

To enable invariance to rotation, above moments require reformulation. Hu [Hu, 1962] proposed the Hu moment invariants. Those expressions were derived from algebraic extensions of the moment-generating function under a pre-set rotation transformation. Hu Moment Invariants consist of a set of nonlinear centralized moment equations, which are also absolutely orthogonal (i.e. rotation) invariant. These moments are translation, scale, and rotation invariant. The drawback of moments is that they are global features as opposed to local, which makes them not suited for perceiving objects which are in incompletely obstructed.

A.3.6 Zernike moments

Zernike [Khotanzad and Hong, 1990] introduced a set of complex polynomials which form a complete orthogonal set over the interior of a unit circle and are specified in polar coordinates in terms of a real valued radial component. Zernike moments are better than the Hu moments in terms of rotation invariance. The two-dimensional Zernike moments [Mukundan and Ramakrishnan, 1995] of an image intensity function $f(r, \theta)$ are defined as :

$$Z_{nm} = \frac{n+1}{\pi} \int_0^1 \int_{-\pi}^{\pi} R_{nm}(r) \exp(-jm\theta) f(r, \theta) r dr d\theta \tag{A.16}$$

where, $r = \sqrt{x^2 + y^2}, \theta = \tan^{-1} \frac{y}{x}, -1 < x, y < 1$

$$R_{nm} = \sum_{s=0}^{\frac{(n-|m|)}{2}} (-1)^s \frac{(n-s)!}{s!(\frac{(n+|m|)}{2} - s)!(\frac{(n-|m|)}{2} - s)!} \times r^{n-2s} \tag{A.17}$$

and $0 \leq |m| \leq n, n - |m|$ is given ; $n > 0$

Since it is convenient to work with real functions, Z_{nm} is often split into its real and imaginary parts C_{nm}, S_{nm} as given below :

$$C_{nm} = \frac{2n+2}{\pi} \sum_0^1 \sum_{-\pi}^{\pi} R_{nm}(r) \cos(m\theta) f(r, \theta) r dr d\theta \tag{A.18}$$

$$S_{nm} = -\frac{2n+2}{\pi}\sum_{0}^{1}\sum_{-\pi}^{\pi} R_{nm}(r)\sin(m\theta)f(r,\theta)rdrd\theta \tag{A.19}$$

where, $m \geq 0, n > 0$

Thirty-six Zernike moments of order zero to ten in n and m are extracted from the normalized image, and the magnitudes are used as the descriptor.

A.3.7 SIFT feature

Scale Invariant Feature Transform (SIFT) (Lowe 2004) is a descriptor derived using the multi-scale extraction. The SIFT descriptors extract information around the key points, correspond to the local extrema on a specified level of scale. Each key point contains its location and its scale. Since these key points contain the location information, this mainly makes key points widely used for image matching locally. Each feature is composed by four parts : the locus (location in which the feature has been found), the scale, the orientation and the descriptor. The descriptor is a vector of 128 dimensions. A space scale image pyramid is first constructed, each level corresponding to a different smoothing level. A Difference of Gaussian (DoG) pyramid is constructed by subtracting the successive images from each other. Local extrema are detected in this DoG pyramid. For each of them, its location and level of smoothness is noted. In this form, the key points are invariant to any scale changes but not to rotation changes. To make these key points rotation-invariant, a local region is defined around the key point and the gradient orientation is calculated in this area. Then, the orientation histogram is constructed and the peak of this orientation histogram is assigned as the orientation of the key point.

A.4 Classifier

A.4.1 SVM classifier

Suppose, there is a set training n vectors of the form $(x_1, y_1), ..., (x_n, y_n)$ where, y_i is the target and $y_i \in 1, -1$ indicating the class to which the vector x_i belongs.

$$\begin{cases} y_i = 1, \text{if } x_i \text{ is text} \\ y_i = -1, \text{if } x_i \text{is non-text} \end{cases} \tag{A.20}$$

Each x_i is a p-dimensional real vector, p represents dimension of used vector descriptor. The "maximum-margin hyper plane" that divides the points x_i for which $y_i = 1$, and those for which $y_i = -1$, is defined so that the distance between the hyper plane and the nearest point x_i is maximized. These hyper planes can be described by

$$\begin{cases} \omega.x_i - b \geq 1, \text{if } y_i = 1 \\ \omega.x_i - b \leq -1, \text{if } y_i = -1 \end{cases} \tag{A.21}$$

w and b are weight and bias. These constraints state that each data point must lie on the correct side of the margin. This can be rewritten as :

$$y_i(\omega.x_i - b) \geq 1, \forall 1 \leq i \leq n \tag{A.22}$$

then the optimization problem is : Minimize $\|w\|$ subject to $y_i(\omega.x_i - b) \geq 1$, for$i = 1, ..., n$. The w and b that solve this problem determine the classifier,

$$sign(\omega.x + b) \rightarrow x \tag{A.23}$$

To implement SVM, we have used available open source SVM software packages LibSVM implemented by Chang and Lin [Chang and Lin, 2011].

Bibliographie

[Adam et al., 2000] Adam, S., Ogier, J., Cariou, C., Mullot, R., Labiche, J., and Gardes, J. (2000). Symbol and character recognition : application to engineering drawings. *International Journal on Document Analysis and Recognition*, 3(2) :89–101.

[Ahmed et al., 2012] Ahmed, S., Liwicki, M., and Dengel, A. (2012). Extraction of text touching graphics using SURF. In *Proceedings - 10th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 349–353.

[Ahmed et al., 2011] Ahmed, S., Weber, M., Liwicki, M., and Dengel, A. (2011). Text/graphics segmentation in architectural floor plans. In *Proceedings of the IEEE International Conference on Document Analysis and Recognition (ICDAR)*, pages 734–738.

[Alaei et al., 2016] Alaei, F., Alaei, A., Blumenstein, M., and Pal, U. (2016). A Brief Review of Document Image Retrieval Methods : Recent Advances. In *Neural Networks (IJCNN), International Joint Conference on*, pages 3500–3507.

[Almazan et al., 2014] Almazan, J., Gordo, A., Fornes, A., and Valveny, E. (2014). Word spotting and recognition with embedded attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(12) :2552–2566.

[Anders, 2003] Anders, K.-H. (2003). A hierarchical graph-clustering approach to find groups of objects. In *Proceedings 5th Workshop on Progress in Automated Map Generalization, Citeseer*, pages 1–8.

[Arthur and Vassilvitskii, 2007] Arthur, D. and Vassilvitskii, S. (2007). k-means++ : The advantages of careful seeding. *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 8 :1027–1035.

[Baird, 1993] Baird, H. S. (1993). Document image defect models and their uses. In *Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on*, pages 62–67.

[Bansal and Sinha, 2002] Bansal, V. and Sinha, R. M. K. (2002). Segmentation of touching and fused Devanagari characters. *Pattern Recognition*, 35(4) :875–893.

[Beis and Lowe, 1997] Beis, J. and Lowe, D. (1997). Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1000–1006.

[Biswas and Das, 2011] Biswas, S. and Das, A. K. (2011). Text segmentation from scanned land map images using radon transform based projection profile. In *Proceedings of the International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, pages 413–418.

[Biswas and Das, 2012] Biswas, S. and Das, A. K. (2012). Text extraction from scanned land map images. In *International Conference on Informatics, Electronics and Vision (ICIEV)*, pages 231–236.

[Biswas and Das, 2013] Biswas, S. and Das, A. K. (2013). Fuzzy Graph Modeling for Text segmentation from Land Map Images. In *Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing*, volume 8251 LNCS, pages 521–529.

[Biswas et al., 2014] Biswas, S., Kumar Das, A., and Chanda, B. (2014). Text Segmentation from Bangla Land Map Images. *Image Processing & Communications*, 19(1) :21–34.

[Bourbakis, 2001] Bourbakis, N. G. (2001). A methodology for document processing : separating text from images. *Engineering Applications of Artificial Intelligence*, 14(1) :35–41.

[Cao and Tan, 2001] Cao, R. and Tan, C. L. (2001). Text/graphics separation in maps. In *International Workshop on Graphics Recognition*, pages 167–177.

[Chalechale et al., 2004] Chalechale, a., Mertins, A., and Naghdy, G. (2004). Edge image description using angular radial partitioning. *IEE Proceedings - Vision, Image, and Signal Processing*, 151(2) :93–101.

[Chang and Lin, 2011] Chang, C.-c. and Lin, C.-j. (2011). LIBSVM : A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3) :1–39.

[Chaudhuri and Pal, 1998] Chaudhuri, B. and Pal, U. (1998). A complete printed Bangla OCR system. *Pattern Recognition*, 31(5) :531–549.

[Chen et al., 2001] Chen, D., Hervé, B., and Jean-Philippe, T. (2001). Text Identification in Complex Background Using SVM. In *Proceedings of the 2001 IEEE Computer Society Conference, Computer Vision and Pattern Recognition, CVPR*, pages 621–626.

[Chen et al., 2002] Chen, D., Odobez, J.-M., and Hervé, B. (2002). Text Segmentation and Recognition in Complex Background Based on Markov Random Field. In *Proceedings. IEEE 16th International Conference on Pattern Recognition*, pages 227–230.

[Chen and Bloomberg, 1997] Chen, F. R. and Bloomberg, D. S. (1997). Extraction of Indicative Summary Sentences from Imaged Documents. In *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, pages 227–232.

[Chen et al., 2009] Chen, G. Y., Bui, T. D., and Krzyzak, A. (2009). Invariant pattern recognition using radon, dual-tree complex wavelet and Fourier transforms. *Pattern Recognition*, 42(9) :2013–2019.

[Chen and Wu, 2009] Chen, Y.-l. and Wu, B.-f. (2009). A multi-plane approach for text segmentation of complex document images. *Pattern Recognition*, 42(7) :1419–1444.

[Chiang et al., 2014] Chiang, Y.-Y., Leyk, S., and Knoblock, C. a. (2014). A Survey of Digital Map Processing Techniques. *ACM Computing Surveys*, 47(1) :1 :1–1 :44.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20(3) :273–297.

[Csurka et al., 2004] Csurka, G., Dance, C. R., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, pages 1–22.

[Dang et al., 2015] Dang, Q. B., Luqman, M. M., Coustaty, M., Nayef, N., Tran, C. D., and Ogier, J. M. (2015). A multi-layer approach for camera-based complex map image retrieval and spotting system. In *2014 4th International Conference on Image Processing Theory, Tools and Applications, IPTA 2014*.

[Do et al., 2012] Do, T.-h., Tabbone, S., and Ramos-terrades, O. (2012). Text / graphic separation using a sparse representation with multi-learned dictionaries. In *International Conference on Pattern Recognition (ICPR)*, pages 689–692.

[Doermann et al., 1996] Doermann, D., Rivlin, E., and al, E. (1996). Applying algebraic and differential invariants for logo recognition. *Machine Vision and Applications*, 9(2) :73–86.

[Doermann, 1998] Doermann, D. (1998). The Indexing and Retrieval of Document Images : A Survey. *Computer Vision and Image Understanding*, 70(3) :287–298.

[Due Trier et al., 1996] Due Trier, Ø., Jain, A. K., and Taxt, T. (1996). Feature extraction methods for character recognition-A survey. *Pattern Recognition*, 29(4) :641–662.

[Epshtein et al., 2010] Epshtein, B., Ofek, E., and Wexler, Y. (2010). Detecting text in natural scenes with stroke width transform. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2963–2970.

[Farulla et al., 2016] Farulla, G., Murru, N., and Rossini, R. (2016). A fuzzy approach for segmentation of touching characters. *arXiv preprint, ArXiv ID :1612.04862*, pages 1–16.

[Fischler and Bolles, 1981] Fischler, M. a. and Bolles, R. C. (1981). Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6) :381–395.

[Fletcher and Kasturi, 1988] Fletcher, L. A. and Kasturi, R. (1988). A Robust Algorithm for Text String Separation from Mixed Text / Graphics Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, I(6) :910–918.

[Frinken et al., 2012] Frinken, V., Fischer, A., Manmatha, R., and Bunke, H. (2012). A novel word spotting method based on recurrent neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2) :211–224.

[Fu et al., 2006] Fu, H., Liu, X., Jia, Y., and Deng, H. (2006). Gaussian mixture modeling of neighbor characters for multilingual text extraction in images. In *Proceedings - International Conference on Image Processing (ICIP)*, pages 3321–3324.

[Gabor, 1946] Gabor, D. (1946). Theory of communication. *Journal of the Institution of Electrical Engineers-Part III : Radio and Communication Engineering*, 93(26) :429–441.

[Garain and Chaudhuri, 2002] Garain, U. and Chaudhuri, B. B. (2002). Segmentation of touching characters in printed devnagari and bangla scripts using fuzzy multifactorial analysis. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 32(4) :449–459.

[Garcia and Apostolidis, 2000] Garcia, C. and Apostolidis, X. (2000). Text detection and segmentation in complex color images. In *Proceedings. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 2326–2329.

[Gatos and Pratikakis, 2009] Gatos, B. and Pratikakis, I. (2009). Segmentation-free word spotting in historical printed documents. In *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, pages 271–275.

[Ghosh and Valveny, 2015] Ghosh, S. K. and Valveny, E. (2015). A sliding window framework for word spotting based on word attributes. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 652–661.

[Girard et al., 2016] Girard, N., Trullo, R., Barrat, S., Ragot, N., and Ramel, J. Y. (2016). Interactive Definition and Tuning of One-Class Classifiers for Document Image Classification. *Proceedings - 12th IAPR International Workshop on Document Analysis Systems, DAS 2016*, pages 358–363.

[Gllavata et al., 2003] Gllavata, J., Ewerth, R., and Freisleben, B. (2003). A robust algorithm for text detection in images. In *Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis (ISPA)*, volume 2, pages 611–616.

[Gloger, 1992] Gloger, J. (1992). Use of the Hough Transform to Separate Merged Text/-Graphics in Forms. In *Proceedings of 11th IAPR International Conference Conference B : Pattern Recognition Methodology and Systems,*, pages 268–271.

[Gomez-Bigorda and Karatzas, 2016] Gomez-Bigorda, L. and Karatzas, D. (2016). TextProposals : a Text-specific Selective Search Algorithm for Word Spotting in the Wild. *arXiv preprint arXiv :1604.02619*.

[Goto and Aso, 2000] Goto, H. and Aso, H. (2000). Character Pattern Extraction Based on Local Multilevel Thresholding and Region Growing. In *Proceedings. IEEE 15th International Conference on Pattern Recognition (ICPR)*, pages 2–5.

[Hartigan and Wong, 1979] Hartigan, J. a. and Wong, M. a. (1979). Algorithm AS 136 : A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society C*, 28(1) :100–108.

[Hase et al., 1997] Hase, H., Shinokawat, T., Yoneda, M., Sakai, M., and Maruyama, H. (1997). Character String Extraction by Multi-stage Relaxation. In *Proceedings of the IEEE 4th International Conference on Document Analysis and Recognition (ICDAR)*, pages 298–302.

[He and Abe, 1996] He, S. and Abe, N. (1996). A Clustering-Based Approach to the Separation of Text Strings from Mixed Text/Graphics Documents. In *Proceedings of the IEEE 13th International Conference on Pattern Recognition (ICPR)*, pages 706–710.

[Heikkilä et al., 2006] Heikkilä, M., Pietikäinen, M., and Schmid, C. (2006). Description of interest regions with center-symmetric local binary patterns. *Computer Vision, Graphics and Image Processing.*, 2 :58–69.

[Ho et al., 1991] Ho, T. K., Hull, J. J., and Srihari, S. N. (1991). Word Recognition With Multi-Level Contextual Knowledge. In *First International Conference on Document Analysis and Recognitio*, pages 905–915.

[Hoang and Tabbone, 2010] Hoang, T. V. and Tabbone, S. (2010). Text extraction from graphical document images using sparse representation. In *Proceedings of the 8th IAPR International Workshop on Document Analysis Systems (DAS)*, volume 2010, pages 143–150.

[Hol et al., 2008] Hol, M., Kalsbeek, F., and Petkov, N. (2008). Grating cell operator for image processing and computer vision - Explanation of parameters.

[Horowitz and Sahni, 1989] Horowitz, E. and Sahni, S. (1989). *Fundamentals of Computer Algorithms*. Rockville, Ill. : Computer Science Press.

[Hu, 1962] Hu, M.-K. (1962). Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2) :179–187.

[Huang et al., 2004] Huang, X., Li, S. Z., and Wang, Y. (2004). Shape localization based on statistical method using extended local binary pattern. In *Image and Graphics, 2004. Proceedings. Third International Conference on (ICIG)*, pages 184–187.

[Impedovo et al., 1991] Impedovo, S., Ottaviano, L., and Occhinegro, S. (1991). Optical Character Recognition - a Survey. *International Journal of Pattern Recognition and Artificial Intelligence*, 05(6) :1–24.

[Jain and Yu, 1998] Jain, A. and Yu, B. (1998). Automatic text location in images and video frames. In *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference*, volume 2, pages 1497–1499.

[Jain and Farrokhnia, 1991] Jain, A. K. and Farrokhnia, F. (1991). Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12) :1167–1186.

[Jain et al., 1999] Jain, A. K., Prabhakar, S., and Hong, L. (1999). A Multichannel approach to fingerprint classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4) :348–359.

[Jones and Mewhort, 2004] Jones, M. N. and Mewhort, D. J. K. (2004). Case-sensitive letter and bigram frequency counts from large-scale English corpora. *Behavior research methods, instruments, & computers : a journal of the Psychonomic Society, Inc*, 36(3) :388–396.

[Journet et al., 2008] Journet, N., Ramel, J. Y., Mullot, R., and Eglin, V. (2008). Document image characterization using a multiresolution analysis of the texture : Application to old documents. *International Journal on Document Analysis and Recognition*, 11(1) :9–18.

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

[Jung et al., 2002] Jung, K., Kim, K. I., Kurata, T., Kourogi, M., and Han, J. (2002). Text Scanner with Text Detection Technology on Image Sequences. In *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, pages 473–476.

[Karatzas et al., 2015] Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V. R., Lu, S., Shafait, F., Uchida, S., and Valveny, E. (2015). ICDAR 2015 competition on Robust Reading. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1156–1160.

[Karatzas et al., 2013] Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., Bigorda, L. G. I., Mestre, S. R., Mas, J., Mota, D. F., Almazan, J. A., and De Las Heras, L. P. (2013). ICDAR 2013 robust reading competition. In *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, pages 1484–1493.

[Khotanzad and Hong, 1990] Khotanzad, A. and Hong, Y. H. (1990). Invariant Image Recognition by Zernike Moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5) :489–497.

[Kim et al., 2002] Kim, E. Y., Chang, J. S., and Kim, H. J. (2002). Automatic Text Location using Cluster-based Template Matching. In *Proceedings. IEEE 16th International Conference on Pattern Recognition*, volume 3, pages 423–426.

[Kingsbury, 1998] Kingsbury, N. (1998). The Dual-Tree Complex Wavelet Transform : A New Efficient Tool For Image Restoration And Enhancement. *Proc. European Signal Processing Conference, EUSIPCO 98, Rhodes*, pages 319–322.

[Kumar et al., 2013] Kumar, A., Yadav, M., Patnaik, T., and Kumar, B. (2013). A Survey on Touching Character Segmentation. *International Journal of Engineering and Advanced Technology (IJEAT)*, 2(3) :569–574.

[Kumar et al., 2007] Kumar, S., Gupta, R., Khanna, N., Member, S., Chaudhury, S., and Joshi, S. D. (2007). Text Extraction and Document Image Segmentation Using Matched Wavelets and MRF Model. *IEEE Transactions on Image Processing*, 16(8) :2117–2128.

[Lazebnik et al., 2006] Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features : Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178.

[Lebourgeois, 1997] Lebourgeois, F. (1997). Robust Multifont OCR System from Gray Level Images. In *Proceedings of the IEEE 4th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1–5.

[Lee et al., 1996] Lee, S. W., Lee, D. J., and Park, H. S. (1996). A new methodology for gray-scale character segmentation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10) :1045–1050.

[Li et al., 2008] Li, J., Tian, Y., Huang, T., and Gao, W. (2008). Multi-polarity text segmentation using graph theory. In *5th IEEE International Conference on Image Processing (ICIP)*, pages 3008–3011.

[Li et al., 2000] Li, L., Nagy, G., Samal, A., Seth, S., and Xu, Y. (2000). Integrated text and line-art extraction from a topographic map. *International Journal on Document Analysis and Recognition*, 2(4) :177–185.

[Liang et al., 2012] Liang, Y., Fairhurst, M. C., and Guest, R. M. (2012). A synthesised word approach to word retrieval in handwritten documents. *Pattern Recognition*, 45(12) :4225–4236.

[Liao et al., 2009] Liao, S., Law, M. W. K., and Chung, A. C. S. (2009). Dominant local binary patterns for texture classification. *IEEE Transactions on Image Processing*, 18(5) :1107–1118.

[Liao et al., 2007] Liao, S., Zhu, X., Lei, Z., Zhang, L., and Li, S. (2007). Learning Multi-scale Block Local Binary Patterns for Face Recognition. In *In International Conference on Biometrics*, pages 828–837.

[Liu et al., 2008] Liu, F., Peng, X., Wang, T., and Lu, S. (2008). A Density-based Approach for Text Extraction in Images. In *IEEE 19th International Conference Pattern Recognition (ICPR)*, pages 1–4.

[Liu et al., 2006] Liu, Q., Jung, C., and Moon, Y. (2006). Text segmentation based on stroke filter. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 129–132.

[Liu et al., 2012] Liu, W., Kiranyaz, S., and Gabbouj, M. (2012). Robust Scene Classification by GIST with Angular Radial Partitioning. *Communications Control and Signal Processing (ISCCSP)*, (May) :1–6.

[Lopresti and Zhou, 1996] Lopresti, D. and Zhou, J. (1996). Retrieval strategies for noisy text. *Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval*, pages pp. 255–269.

[Louloudis et al., 2009] Louloudis, G., Stamatopoulos, N., and Gatos, B. (2009). A novel two stage evaluation methodology for word segmentation techniques. In *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, pages 686–690.

[Lowe, 1999] Lowe, D. G. (1999). Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 2(8) :1150–1157.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2) :91–110.

[Lu, 1993] Lu, Y. (1993). On the segmentation of touching characters. In *Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on*, pages 440–443.

[Lu, 1998] Lu, Z. (1998). Detection of Text Regions From Digital Engineering Drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4) :431–439.

[Lucas et al., 2003] Lucas, S. M., Panaretos, A., Sosa, L., Tang, A., Wong, S., and Young, R. (2003). ICDAR 2003 robust reading competitions. In *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, pages 682–687.

[Maguluri et al., 2013] Maguluri, H. B., Tian, Q., and Li, B. (2013). Detecting Text in Floop Maps Using Hisogram of Oriented Gradients. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1932–1936.

[Mancas-Thillou and Gosselin, 2006] Mancas-Thillou, C. and Gosselin, B. (2006). Character Segmentation-by-Recognition Using Log-Gabor Filters. In *International Conference on Pattern Recognition*, pages 18–21.

[Mancas-Thillou et al., 2005] Mancas-Thillou, C., Mancas, M., and Gosselin, B. (2005). Camera-based degraded character segmentation into individual components. In *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, volume 2005, pages 755–759.

[Manmatha et al., 1996] Manmatha, R., Han, C., and Riseman, E. (1996). Word Spotting : A New Approach to Indexing Handwriting. In *In Computer Vision and Pattern Recognition, Proceedings (CVPR)*, pages 631–637.

[Matas et al., 2004] Matas, J., Chum, O., Urban, M., and Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10) :761–767.

[Matti and Okun, 2001] Matti, P. and Okun, O. (2001). Edge-Based Method for Text Detection from Complex Document Images. In *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, pages 286–291.

[Mehri et al., 2015] Mehri, M., Pierre, H., Sliti, N., Gomez-kr, P., Essoukri, N., Amara, B., and Mullot, R. (2015). Extraction of Homogeneous Regions in Historical Document Images. In *VISAPP*, volume 3, pages 47–54.

[Meshesha and Jawahar, 2008] Meshesha, M. and Jawahar, C. V. (2008). Matching word images for content-based retrieval from printed document images. *International Journal on Document Analysis and Recognition*, 11(1) :29–38.

[Mukundan and Ramakrishnan, 1995] Mukundan, R. and Ramakrishnan, K. R. (1995). Fast computation of Legendre and Zernike moments. *Pattern Recognition*, 28(9) :1433–1442.

[Muralikrishna and Koti Reddy, 2011] Muralikrishna, M. and Koti Reddy, D. V. R. (2011). An OCR-character segmentation using routing based fast replacement paths in reach algorithm. In *ICIIP 2011 - Proceedings : 2011 International Conference on Image Information Processing*.

[Ojala et al., 2002] Ojala, T., Pietikäinen, M., and Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7) :971–987.

[Pal et al., 2003] Pal, U., Belaid, A., and Choisy, C. (2003). Touching numeral segmentation using water reservoir concept. *Pattern Recognition Letters*, 24(1) :261–272.

[Pal and Chaudhuri, 2004] Pal, U. and Chaudhuri, B. B. (2004). Indian script character recognition : A survey. *Pattern Recognition*, 37(9) :1887–1899.

[Pal et al., 2006] Pal, U., Kimura, F., Roy, K., and Pal, T. (2006). Recognition of english multi-oriented characters. In *Proceedings - International Conference on Pattern Recognition*, volume 2, pages 873–876.

[Pal et al., 2010] Pal, U., Pratim Roy, P., Tripathy, N., and Llads, J. (2010). Multi-oriented Bangla and Devnagari text recognition. *Pattern Recognition*, 43(12) :4124–4136.

[Pan et al., 2007] Pan, W., Bui, T., and Suen, C. (2007). Text Segmentation from Complex Background Using Sparse Representations. In *IEEE 9th International Conference on Document Analysis and Recognition (ICDAR)*, pages 412–416.

[Pezeshk and Tutwiler, 2010a] Pezeshk, A. and Tutwiler, R. L. (2010a). Extended character defect model for recognition of text from maps. In *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*, pages 85–88.

[Pezeshk and Tutwiler, 2010b] Pezeshk, A. and Tutwiler, R. L. (2010b). Improved multi angled parallelism for separation of text from intersecting linear features in scanned topographic maps. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1078–1081.

[Pierrot et al., 1995] Pierrot, M., Le, H., and Stamon, G. (1995). Character string recognition on maps , a rotation-invariant recognition method. *Pattern Recognition Letters*, 16(12) :1297–1310.

[Pintus et al., 2016] Pintus, R., Yang, Y., Gobbetti, E., and Rushmeier, H. (2016). An automatic word-spotting framework for medieval manuscripts. In *2015 Digital Heritage International Congress, Digital Heritage 2015*, pages 5–12.

[Pouderoux et al., 2006] Pouderoux, J., Gonzato, J.-c., Pereira, A., and Guitton, P. (2006). Toponym Recognition in Scanned Color Topographic Maps. In *IEEE 9th International Conference on Document Analysis and Recognition (ICDAR)*, pages 531–535.

[R. Fisher et al., 2003] R. Fisher, Perkins, S., Walker, A., and Wolfart., E. (2003). Spatial Filters - Laplacian of Gaussian.

[Rabiner, 1989] Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2) :257–286.

[Ramana Murthy et al., 2013] Ramana Murthy, O. V., Roy, S., Narang, V., Hanmandlu, M., and Gupta, S. (2013). An approach to divide pre-detected Devanagari words from the scene images into characters. *Signal, Image and Video Processing*, 7(6) :1071–1082.

[Ramel and Vincent, 2003] Ramel, J.-Y. and Vincent, N. (2003). Strategy for Line Drawing Understanding. In *International Workshop on Graphics Recognition*, pages 1–12.

[Ramel et al., 1998] Ramel, J. Y., Vincent, N., and Emptoz, H. (1998). A coarse vectorisation as an initial representation for the understanding of line drawing images. In *In Graphics Recognition - Algorithms and Systems. Lecture Notes in Computer Science*, volume 1389, pages 48–55.

[Ramel et al., 2000] Ramel, J. Y., Vincent, N., and Emptoz, H. (2000). A structural representation for understanding line-drawing images. *International Journal on Document Analysis and Recognition*, 3(2) :58–66.

[Rath and Manmatha, 2003] Rath, T. M. and Manmatha, R. (2003). Features for word spotting in historical manuscripts. In *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, pages 218–222.

[Rätsch et al., 2000] Rätsch, G., Schölkopf, B., Mika, S., and Müller, K. (2000). SVM and boosting : One class. Technical report.

[Rendek et al., 2004] Rendek, J., Masini, G., Dosch, P., and Tombre, K. (2004). The search for genericity in graphics recognition applications : Design issues of the qgar software system. In *In International Workshop on Document Analysis Systems*, pages 366–377.

[Ricard et al., 2005] Ricard, J., Coeurjolly, D., and Baskurt, A. (2005). Generalizations of angular radial transform for 2D and 3D shape retrieval. *Pattern Recognition Letters*, 26(14) :2174–2186.

[Rigaud et al., 2013] Rigaud, C., Karatzas, D., Weijer, J. V. D., Burie, J.-c., and Ogier, J.-m. (2013). Automatic Text Localisation in Scanned Comic Books. In *9th International Conference on Computer Vision Theory and Applications*, pages 814–819.

[Rocha and Pavlidis, 1993] Rocha, J. and Pavlidis, T. (1993). A solution to the problem of touching and broken characters. In *Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR '93)*, pages 602–605.

[Rodríguez-Serrano and Perronnin, 2012] Rodríguez-Serrano, J. A. and Perronnin, F. (2012). A model-based sequence similarity with application to handwritten word spotting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11) :2108–2120.

[Roussopoulos et al., 1995] Roussopoulos, N., Kelley, S., and Vincent, F. (1995). Nearest neighbor queries. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data - SIGMOD '95*, pages 71–79.

[Roy et al., 2009a] Roy, P., Pal, U., and Llados, J. (2009a). Touching Text Character Localization in Graphical Documents Using SIFT. In *International Workshop on Graphics Recognition*, pages 199–211.

[Roy et al., 2008a] Roy, P., Pal, U., Llados, J., and Kimura, F. (2008a). Convex hull based approach for multi-oriented character recognition from graphical documents. In *IEEE 9th International Conference on Pattern Recognition (ICPR)*, pages 1–4.

[Roy et al., 2008b] Roy, P. P., Pal, U., and Lladós, J. (2008b). Recognition of multi-oriented touching characters in graphical documents. In *Proceedings - 6th Indian Conference on Computer Vision, Graphics and Image Processing, ICVGIP 2008*, pages 297–304.

[Roy et al., 2010] Roy, P. P., Pal, U., and Lladós, J. (2010). Query driven word retrieval in graphical documents. In *Proceedings of the 8th IAPR International Workshop on Document Analysis Systems - DAS '10*, pages 191–198.

[Roy et al., 2012] Roy, P. P., Pal, U., and Llados, J. (2012). Text line extraction in graphical documents using background and foreground information. *International Journal on Document Analysis and Recognition*, 15(3) :227–241.

[Roy et al., 2009b] Roy, P. P., Pal, U., Llados, J., and Delalandre, M. (2009b). Multi-oriented and multi-sized touching character segmentation using dynamic programming. In *Proceedings of the IEEE 10th International Conference on Document Analysis and Recognition (ICDAR)*, pages 11–15.

[Roy et al., 2008c] Roy, P. P., Pal, U., Lladós, J., and Kimura, F. (2008c). Multi-Oriented English Text Line Extraction Using Background and Foreground Information. In *The 8th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 315–322.

[Rusiñol et al., 2011] Rusiñol, M., Aldavert, D., Toledo, R., and Lladós, J. (2011). Browsing heterogeneous document collections by a segmentation-free word spotting method. In *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, pages 63–67.

[Rusiñol and Lladós, 2008] Rusiñol, M. and Lladós, J. (2008). Word and symbol spotting using spatial organization of local descriptors. In *DAS 2008 - Proceedings of the 8th IAPR International Workshop on Document Analysis Systems*, pages 489–496.

[Rusiñol and Lladós, 2010] Rusiñol, M. and Lladós, J. (2010). *Symbol spotting in digital libraries : Focused retrieval over graphic-rich document collections.*

[Saba et al., 2010] Saba, T., Sulong, G., and Rehman, A. (2010). A Survey on Methods and Strategies on Touched Characters Segmentation. *International Journal of Research and Reviews in Computer Science (IJRRCS)*, 1(2) :103–114.

[Salton, 1989] Salton, G. (1989). Automatic text processing : the transformation. *Reading : Addison-Wesley.*

[Seo, 2006] Seo, N. (2006). Texture Segmentation using Gabor Filters. Technical report.

[Shen et al., 2007] Shen, L., Bai, L., and Fairhurst, M. (2007). Gabor wavelets and General Discriminant Analysis for face identification and verification. *Image and Vision Computing*, 25(5) :553–563.

[Shivakumara et al., 2011] Shivakumara, P., Phan, T. Q., and Tan, C. L. (2011). A Laplacian approach to multi-oriented text detection in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2) :412–419.

[Smeulders et al., 2000] Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., and Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12) :1349–1380.

[Smith, 2007] Smith, R. (2007). An overview of the tesseract OCR engine. In *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, volume 2, pages 629–633.

[Srihari, 1995] Srihari, R. K. (1995). Automatic Indexing and Content-Based Retrieval of Captioned Images. *Computer*, 28(9) :49–56.

[Strouthopoulos and Nikolaidis, 2008] Strouthopoulos, C. and Nikolaidis, A. (2008). A robust technique for text extraction in mixed-type binary documents. In *19th International Conference on Pattern Recognition (ICPR)*, pages 1–4.

[Su and Cai, 2009] Su, F. and Cai, S. (2009). A character extraction and recognition method for line drawings. In *Proceedings of the 2nd International Congress on Image and Signal Processing (CISP)*, pages 1–5.

[Sural and Das, 1999] Sural, S. and Das, P. (1999). An MLP using Hough transform based fuzzy feature extraction for Bengali script recognition. *Pattern Recognition Letters*, 20(8) :771–782.

[Takasu et al., 1994] Takasu, A., Satoh, S., and Katsura, E. (1994). A document understanding method for database construction of an electronic library. In *In Pattern Recognition, Conference B : Computer Vision & Image Processing., Proceedings of the 12th IAPR International. Conference on*, volume 2, pages 463–466.

[Tan and Ng, 1998] Tan, C. L. and Ng, P. O. . (1998). Text Extraction Using Pyramid. *Pattern Recognition*, 31(1) :63–72.

[Tombre et al., 2002] Tombre, K., Tabbone, S., Pélissier, L., Lamiroy, B., and Dosch, P. (2002). Text/graphics separation revisited. In *International Workshop on Document Analysis Systems (DAS)*, pages 200–211.

[Vapnik, 1995] Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory.*

[Wall and Danielsson, 1984] Wall, K. and Danielsson, P.-E. (1984). A fast sequential method for polygonal approximation of digitized curves. *Computer Vision, Graphics, and Image Processing*, 28(2) :220–227.

[Wang et al., 2012] Wang, T., Wu, D. J., Coates, A., and Ng, A. Y. (2012). End-to-end text recognition with convolutional neural networks. *ICPR, International Conference on Pattern Recognition*, pages 3304–3308.

[Weis, 2009] Weis (2009). Random Noise_ May 2009.

[Wikimedia, 2016] Wikimedia (2016). Digital image processing - Wikipedia, the free encyclopedia.

[Witten et al., 1994] Witten, I. H., Bell, T. C., Emberson, H., Inglis, S., and Moffat, A. (1994). Textual Image Compression : Two-Stage Lossy/Lossless Encoding of Textual Images. *Proceedings of the IEEE*, 82(6) :878–888.

[Wolf and Jolion, 2005] Wolf, C. and Jolion, J.-M. (2005). Object count / Area Graphs for the Evaluation of Object Detection and Segmentation Algorithms. *International Journal of Document Analysis and Recognition (IJDAR)*, 8(4) :280–296.

[Wu et al., 1997] Wu, V., Manmatha, R., Riseman, E. M., Wu, V., Manmatha, R., and Riseman, E. M. (1997). TextFinder : An Automatic System To Detect And Recognize Text In Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11) :1224–1229.

[Wu et al., 2005] Wu, W., Chen, D., and Yang, J. (2005). Integrating Co-training and Recognition for Text Detection. In *IEEE International Conference on Multimedia and Expo*, page 4.

[Xu et al., 2016a] Xu, P., Miao, Q., Chen, X., and Nie, W. (2016a). Graphic-based character grouping in topographic maps. *Neurocomputing*, 189 :160–170.

[Xu et al., 2016b] Xu, P., Miao, Q., Liu, R., Chen, X., and Fan, X. (2016b). Dynamic character grouping based on four consistency constraints in topographic maps. *Neurocomputing*, 212 :96–106.

[Yao et al., 2014] Yao, C., Bai, X., and Liu, W. (2014). A unified framework for multioriented text detection and recognition. *IEEE Transactions on Image Processing*, 23(11) :4737–4749.

[Ye et al., 2001] Ye, X., Cheriet, M., and Suen, C. Y. (2001). Stroke-Model-Based Character Extraction from Gray-Level Document Images. *IEEE Transactions on Image Processing*, 10(8) :1152–1161.

[Zhang et al., 2002] Zhang, J., Tan, T., and Ma, L. (2002). Invariant texture segmentation via circular Gabor filters. *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, 2(2) :901–904.

[Zhang et al., 2013] Zhang, X., Ai, T., Stoter, J., Molenaar, M., and Kraak, M.-j. (2013). Building pattern recognition in topographic data : examples on collinear and curvilinear alignments. *Geoinformatica*, 17 :1–33.

[Zhang et al., 2015a] Zhang, Y., Lai, J., and Yuen, P. C. (2015a). Text string detection for loosely constructed characters with arbitrary orientations. *Neurocomputing*, 168 :970–978.

[Zhang et al., 2012] Zhang, Y., Wang, C., Xiao, B., and Shi, C. (2012). A New Text Extraction Method Incorporating Local Information. In *International Conference on Frontiers in Handwriting Recognition*, pages 252–255.

[Zhang et al., 2015b] Zhang, Y., Wang, S., Sun, P., and Phillips, P. (2015b). Pathological brain detection based on wavelet entropy and Hu moment invariants. *Bio-medical materials and engineering*, 26 :S1283–S1290.

[Zhu and Zanibbi, 2013] Zhu, S. and Zanibbi, R. (2013). Label detection and recognition for uspto images using convolutional k-means feature quantization and ada-boost. In *IEEE 12th International Conference on Document Analysis and Recognition (ICDAR)*, pages 633–637.

Résumé :

Les outils et méthodes d'analyse d'images de documents (DIA) donnent aujourd'hui la possibilité de faire des recherches par mots-clés dans des basesd'images de documentsalors même qu'aucune transcription n'est disponible. Dans ce contexte, beaucoup de travaux ont déjà été réalisés sur les OCR ainsi que sur des systèmes de repérage de mots (*spotting*) dédiés à des documents textuels avec une mise en page simple. En revanche, très peu d'approches ont été étudiées pour faire de la recherche dans des documents contenant du texte multi-orienté et multi-échelle, comme dans les documents graphiques. Par exemple, les images de cartes géographiques peuvent contenir des symboles, des graphiques et du texte ayant des orientations et des tailles différentes. Dans ces documents, les caractères peuvent aussi être connectés entre eux ou bien à des éléments graphiques. Par conséquent, le repérage de mots dans ces documents se révèle être une tâche difficile. Dans cette thèse nousproposons un ensemble d'outils et méthodes dédiés au repérage de mots écrits en caractères bengali ou anglais (script Roman) dans des images de documents géographiques. L'approche proposéerepose sur plusieurs originalités. Premièrement, nous proposons de générer une représentation structurelle de bas niveau du contenu des documents, séparant les éléments textuels des éléments graphiques. L'originalité vient ici du fait que l'information est produite à la fois au niveau pixel (par des méthodes de filtrage et *clustering*) et à un niveau structurel élémentaire (analyse et classification de composantes connexes). Egalement, la détection des éléments textuels issus du filtrage est renforcée par l'usage de classificateursreposant sur des phases d'apprentissage et de reconnaissance itératives(les éléments étant très vraisemblablement du texte sont utilisés pour l'apprentissage initial et celui-ci est renforcé après chaque phase d'identification en intégrant les nouveaux éléments textuels identifiés). Par ailleurs, chaque niveau d'information aboutit à la création de cartes de probabilités au lieu de fournir pour chaque région de l'image une décision stricte (texte ou graphique). Ces cartes de probabilité peuvent être utilisées séparément ou agrégées pour extraire différents types d'information (identification de leurcontenu, repéragede contenu textuel, etc.). Ces différentes approches et niveaux d'information ont été utilisés pour évaluer leur qualité pour une tâche de séparation entre la couche texte et la couche graphique. Ellesont été comparées avec différentes méthodesde la littérature, tant dans le domaine fréquence que dans le domaine spatial. Une fois cette description élémentaire obtenue (séparation texte-graphique), un niveau de description lexical est rajouté au document en séparant les éléments textuels connectés entre eux par une méthode à base de water-*reservoir*afin d'identifier les caractères eux-mêmes. Ici, des descripteurs invariants à l'échelle et à la rotation sont utilisés avec des SVMs. Partant de ce résultat, la méthode de *spotting* permettant la recherche d'un mot-clé procède en plusieurs étapes. L'initialisation s'effectue en recherchant au niveau lexical les éléments (caractères) correspondants à la requête et ayant été reconnus avec un bon taux de confiance par le classificateur. En effet, à cause de la complexité inhérente aux documents (dégradations, liaisons inter-caractères ou liaisons texte-graphique), certains caractères de la requête peuvent ne pas être identifiés clairement ou induire des ambigüités dans la recherche, ce que nous préférons évité en premier abord. En partant de ces éléments textuels stables bien identifiés, et en prenant en compte leur position, taille et orientation, nous estimons les régions candidates correspondant aux parties manquantes de la requête. Afin d'identifier ces éléments manquants, nous utilisons une méthode à base de points d'intérêts (SIFT) pour confirmer leur présence dans les régions candidates. Nous avons effectué des expérimentations à la fois sur des cartes numérisées en anglais (Roman) et enbengali. Les résultats expérimentaux démontrent que la méthode est efficace pour repérer les mots dans des documents graphiques. Le jeu de données et la vérité terrain correspondante sont rendus publics afin que d'autres chercheurs puissent se comparer et faire de nouvelles propositions.

Mots clés : Analyse d'images de documents, repérage de mots (*word spotting*), documents graphiques, recherche d'information, séparation texte-graphique, filtrage, cartes de probabilité, points d'intérêts (SIFT), Bengla

Abstract :

Word spotting in graphical documents is a very challenging task. To address such scenarios this thesis deals with developing a word spotting system dedicated to geographical documents with Bangla and English (Roman) scripts. Inthe proposed system, at first, text-graphics layers are separated using filtering, clustering and self-reinforcement through classifier. Additionally, instead of using binary decision we have used probabilistic measurement to represent thetext components. Subsequently, in the text layer, character segmentation approach is applied using water-reservoir based method to extract individual character from the document. Then recognition of these isolated characters is done using rotation invariant feature, coupled with SVM classifier. Well recognized characters are then grouped based on their sizes. Initial spotting is started to find a query word among those groups of characters. In case if the system could spot a word partially due to any noise, SIFT is applied to identify missing portion of that partial spotting. Experimental results on Roman and Bangla scripts document images show that the method is feasible to spot a location in text labeled graphical documents. Experiments are done on an annotated dataset which was developed for this work. We have made this annotated dataset available publicly for other researchers.

Keywords : Document Image Analysis, Word Spotting, Graphical documents, Information Retrieval, Probability matrix information, 2-D Filter, Water Reservoir Principle, Clustering, SIFT

Final Ackowledgement

This work is partly funded by Indo-French Centre for the Promotion of Advanced Research (IFCPAR) Project (No 4700-IT-1).