ABSTRACT

Title of dissertation: EFFICIENT MACHINE LEARNING METHODS FOR DOCUMENT IMAGE ANALYSIS

Jayant Kumar, Doctor of Philosophy, 2013

Dissertation directed by: Professor Larry Davis
Department of Computer Science

Dr. David Doermann
University of Maryland Institute for
Advanced Computer Studies

With the exponential growth in volume of multimedia content on the internet, there has been an increasing interest for developing more efficient and scalable algorithms to learn directly from data without excessive restrictions on nature of the content. In the context of document images, many large scale digitization projects have called for reliable and scalable triage methods for enhancement, segmentation, grouping and categorization of captured images. Current approaches, however, are typically limited to a specific class of documents such as scanned books, newspapers, journal articles or forms for example, and analysis and processing of more unconstrained and noisy heterogeneous document collections has not been as widely addressed. Additionally, existing machine-learning based approaches for document processing need to be carefully applied to handle the challenges associated with large and imbalanced training data.

In this thesis, we address these challenges in three primary applications of document image analysis - low-level document enhancement, mid-level handwritten line segmentation, and high-level classification and retrieval. We first present a data selection method for training Support Vector Machines (SVM) on large-scale data sets. We apply the proposed approach to pixel-level document image enhancement, and show promising results with a relatively small number of training samples. Second, we present a graph-based method for segmentation of handwritten document images into text-lines which is more efficient and adaptive than previous approaches. Our approach demonstrates that combining results from local and global methods enhances the final performance of text-line segmentation. Third, we present an approach to compute structural similarities between images for classification and retrieval. Results on real-world data sets show that the approach is more effective than earlier approaches when the labeled data is limited. We extend our classification approach to a completely unsupervised setting, where both the number of classes and representative samples from each class is assumed to be unknown. We present a method for computing similarities based on learned structural patterns and correlations from the given data. Experiments with four different data sets show that our approach can estimate number of classes in large document collections and group structurally similar images with a high-accuracy.

EFFICIENT MACHINE METHODS FOR DOCUMENT IMAGE ANALYSIS

by

Jayant Kumar

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2013

Advisory Committee:
Professor Larry Davis, Chair
Dr. David Doermann, Co-chair
Professor David Jacobs
Professor Ramani Duraiswami
Professor Douglas W. Oard

Acknowledgments

I would like to thank my advisor, Dr. David Doermann for his guidance, inspiration, and support during my graduate studies. I have learned a great deal from him about research, topics in document image understanding, and other important academic lessons. I deeply cherish the time I have spent working with him.

I would also like to thank my committee chair, Professor Larry Davis for taking interest in my research and for providing useful suggestions during my presentations. I wish to thank Professor David Jacobs with whom I took computer vision courses in the beginning of my graduate studies, and which helped me in getting started with my research. Thanks are due to Professor Ramani Duraiswami and Professor Doug Oard for agreeing to serve on my thesis committee and for sparing their invaluable time reviewing the manuscript.

I have also benefited greatly from knowing and working with Wael Abd-Almageed, Zhuolin Jiang, Jaishanker Pillai, and Ming-Yu Liu while at University of Maryland, and with Francine Chen, Raja Bala, and Huaigu Cao during my internship at FXPAL, Xerox Research and BBN Technologies respectively. I would also like to thank my lab mates Guangyu Zhu, Xu Liu, Mudit Agrawal, Radu Dondera, Rajiv Jain, Xianzhi Du and Chung-Ta Ku who were very supportive. Special thanks to my lab mates Peng Ye and Le Kang who were always available to discuss and review my work, and provided many constructive comments on my work.

I owe my deepest thanks to my family - my parents, brother Abinash and wife Shaily who have always encouraged me and guided me through my career. Words

cannot express the gratitude I owe them.

I would like to acknowledge financial support for different projects discussed herein from DARPA through BBN and US government through NSF.

It is impossible to remember all, and I apologize to those I have inadvertently left out. Thanks everyone!

Table of Contents

List of Tables

List of Figures

List of Abbreviations

DARPA	Defense Advanced Research Projects Agency
FCC	Federal Communications Commission
GALE	Global Autonomous Language Exploitation
MADCAT	Multilingual Automatic Document Classification Analysis and Translation
NSF	National Science Foundation
NIST	National Institute of Standards and Technology
UMIACS	University of Maryland Institute for Advanced Computer Studies

Chapter 1: Introduction

Over the past several decades, the document analysis community has been focused primarily on developing specific solutions to deal with rather narrow ranges of document content. As the number of large scale digitization projects involving heterogenous content continues to grow, there is a compelling need for reliable and scalable triage methods for enhancement, segmentation, classification and categorization of document images.

In this thesis our goal is to address the challenges of processing large collections by applying learning approaches that can efficiently handle large amounts of data for enhancement, segmentation, classification and categorization. While the final goal is typically optical character recognition for indexing and retrieval, the ability to "organize" the collection apriori is essential. Because current methods are typically limited to specific classes of documents such as scanned books, newspapers, journal articles, forms, memos etc., some initial processing is required if we do not know the class ahead of time. Furthermore, analysis and processing of more unconstrained and noisy document collections have not been as widely addressed by existing methods.

To address these challenges, we can first turn to the machine learning community. The last two decades have seen a tremendous growth in machine learning algorithms applied to problems in computer vision and natural language processing. The ability of a method to learn directly from data, and adapt to different settings of input signals has been very successful. For many document image analysis and processing problems, approaches based on *learning* have given the state-of-the-art results on narrow problems. For example, Artificial Neural networks (ANN) have been used extensively in many pre-processing tasks including binarization, noise reduction, skew detection and character thinning [3]. In [4], Le Cun *et al.* provides an interesting overview on different ANN models for recognition of handwritten words. A wide variety of methods exist for detecting tables, and analyzing their structures [5]. Many recent character recognition systems use Support Vector Machines (SVMs) with one-versus-all strategy [6] and Hidden Markov Model (HMMs) have been applied to handwriting recognition [7]. Forensic document image analysis problems like *signature verification* and *writer identification* have also taken advantages of existing machine learning models [8].

However, when it comes to handling the image data at the pixel level, the first problem faced by most machine learning techniques is the large number of training instances. When the dimensionality of feature space is also high, it makes it more difficult to train a learning method in a reasonable amount of time. Moreover, when the number of classes to be labeled is large (for example, in pixel labeling), many learning algorithms initially designed as binary classifiers (for example, SVM) need to be extended or applied multiple times to get the multi-class classification results. Often these strategies become infeasible as the number of classes grow. In many scenarios, obtaining enough labeled data is costly and time consuming. In light of

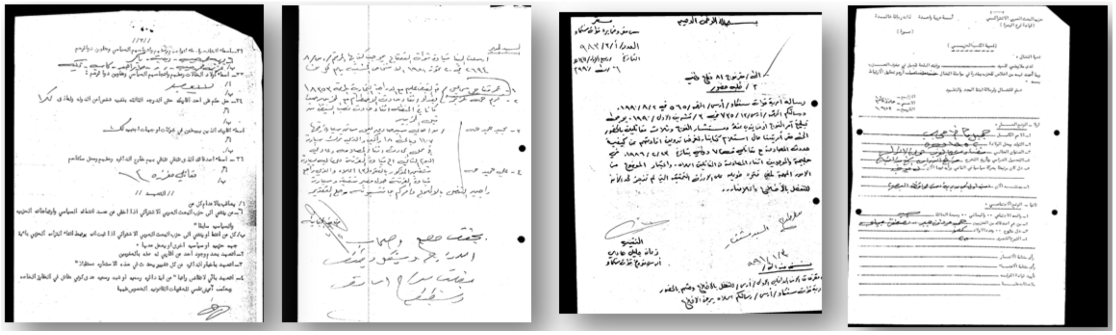


Figure 1.1: Sample document images from MADCAT program

these problems, the recent focus of many vision researchers have been to develop variants of learning approaches which are scalable, require minimal training samples, and work well with high-dimensional data.

In this dissertation, we focus on problems associated with the processing of *unconstrained* and *noisy* document collection with mixed content much of which comes via the MADCAT program (see Figure 1.1). We first define and formulate the problem in the context of a learning based approach, and then discuss challenges associated with scaling and time-efficiency. We present methods which advocate the use of little or no training data, either by relying on strong representations or by selecting strategies which require minimal supervision. More specifically, we focus on large-scale support vector learning for enhancement, and efficient approaches for classification and segmentation of document images. We then discuss the contributions of our work for each of the problems.

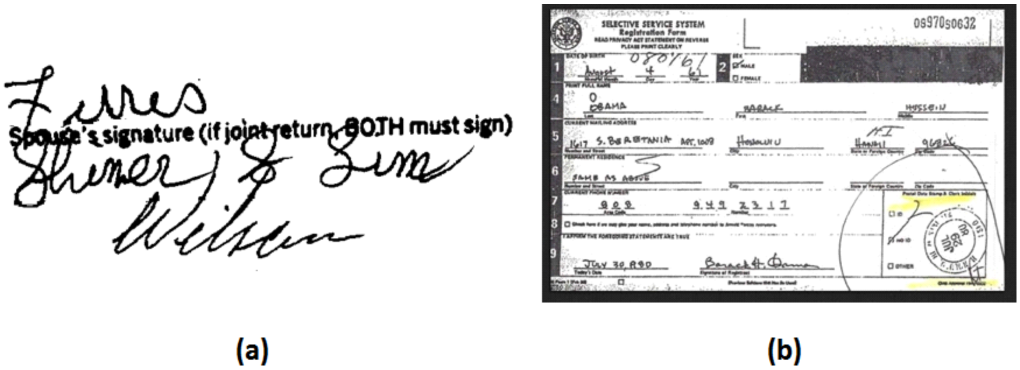

Spouse's signature (if joint return, BOTH must sign)
SELECTIVE SERVICE SYSTEM
Registration Form
(a)
(b)

Figure 1.2: Sample document image content demonstrating a class of problems which require processing at a pixel-level. (a) Handwritten signature overlaps with the printed name. Pixel-level labeling is required in such cases. (b) The noise and human annotations interfere with main content at a pixel-level.

1.1 Data Selection for SVMs amd Applications

There are many problems in document image analysis which require processing of low-level content in images such as separation of handwritten/machine-print text, noise removal and rule-line removal, to name a few. In many scenarios, handwritten text is written so close to the printed content that strokes overlap with the printed text (Figure 1.2(a)). Since the processing of handwritten and printed text require different pipelines, it is necessary to accurately separate both types of text. Similarly, noise and degradations in document images often occur at a very-low level, and require processing at a pixel-level (Figure 1.2(b)). The general philosophy has traditionally been to abstract the pixels by considering small windows or connected components, but there are many applications in document analysis that would benefit from pixel level classification [9, 10, 11, 12, 13].

1.1.1 Problem Definition and Challenges

Although at a very basic level, above separation problems are pixel classification problems, until recently researchers have avoided modeling the problem in this way. Previous approaches had to carefully consider the choice of classifiers and data structures due to the large scale nature of these problems [10, 11]. Assuming a standard size of document, the amount of data at pixel-level becomes huge (N images x 300dpi x 8.5" x 11") and even more in gray and color space. The main bottleneck of taking a pixel-based classification approach is the feature computation and classification time for each pixel. Traditional feature extraction and supervised learning approaches can be very time-consuming. The support vector machine (SVM) is one of the most popular method for supervised classification but due to its quadratic time ($O(n^2)$) dependency on the size of data, SVM cannot handle large training sets. To train 10 million 100-dimensional points, it takes 24 hours with just one set of parameters. In the formulation of SVM, however, the final decision surface depends on only a small subset of training data called Support vectors (SVs) [14]. Hence, for large datasets it becomes important to first select points which are likely to be SVs and then solve a much smaller quadratic programming problem.

We consider the problem of selecting the *most informative* points from a large set of training samples for effectively reducing the size of training set for SVM learning. Our motivation is that in large scale datasets, points in the close neighborhood (in feature space) of an already selected point might be redundant, and do not contribute to the decision surface learned during the training. For example, pixels which are spatially close might have similar characteristics in most cases, and these redundancies can be used to effectively reduce large-scale datasets. The challenge is to select these points efficiently so as to reduce the overall computational cost. At the same time, it is important to assess how many points should be selected so that the accuracy of classification is not affected.

As an application of our data selection method we consider the problem of rule-line removal in document images. It is very common to find rule-lines in handwritten documents whose primary purpose is to keep the base line of written content straight. But the interaction of text with pre-printed lines changes its characteristics to an extent that segmentation and recognition becomes difficult. Many character recognition and segmentation systems are developed for clean and *processed* content, and when images with overlapping rule-lines and text are processed through these systems, the lines can affect performance drastically [15, 16, 17]. Other documents processing modules such as text-line extraction [18] and page segmentation [19] methods which are based on the assumption that characters and/or words form separate components fail to work in these scenarios too. For example, in forms, horizontal rule-lines tend to connect the characters/words making connected-component methods unreliable. If the page has vertical lines for its margin then all the text-lines may become connected making the whole foreground content a single connected-component.

Rule-line removal is still considered a challenging task due to the fluctuation in thickness of rule-lines, text-line interactions and the large variation in shape of handwritten characters. Existing approaches often fail when the lines are severely

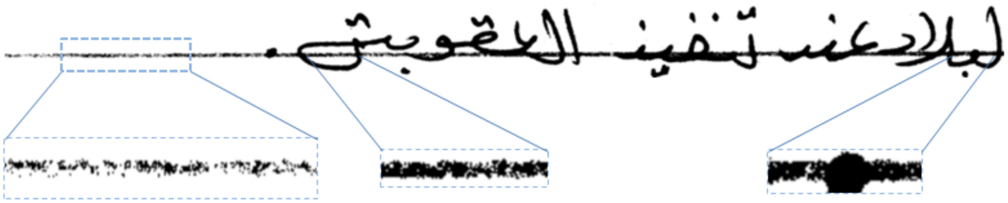


Figure 1.3: Three different parts of the same rule-line having different width and level of degradation.

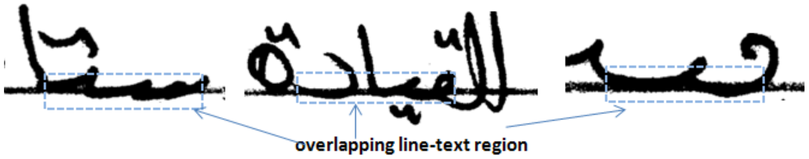


Figure 1.4: Example of scenarios where Arabic characters exhibit long baseline. The overlap with similar width rule-line makes the separation very difficult.

broken, not straight, and/or when the rule-lines interact significantly with the text [16]. As shown in Figure 1.3, different parts of the same rule-line exhibit different characteristics with respect to level of degradation, thickness and radius of curvature. These variations are introduced during the collection, scanning and binarization of the documents. Features extracted from the rule-lines and the regular baselines have very similar distributions which may lead to ambiguity in segmentation and recognition of words and characters, in particular in Arabic. Figure 1.4 shows some example scenarios where the baseline of Arabic characters completely overlap the rule-line. It is even difficult for humans, to mark the exact boundaries of baselines in Arabic scripts [16].

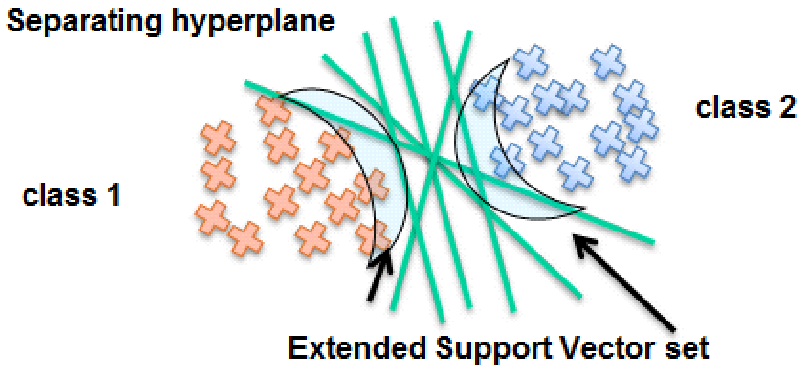


Figure 1.5: Points in the extended support vector (ESV) set are more likely to become support vectors than points in the Convex-hull.

1.1.2 Contributions

We present a method to select probable support-vector points from a large training set to do fast SVM training. While previous methods have focussed on sampling from the Convex-hull of the datasets, we target the extended support vector (ESV) set which has relatively fewer redundant points than Convex-hull (Figure 1.5). We use random subspaces to select points in each iteration. We show that the points are being selected from the ESV set. Our experiments show that our method can effectively reduce a large training set for SVM training, thereby reducing the overall training time.

For rule-line removal problem, we first present features based on an integral-image representation [20] which are not only discriminative for text/rule-line classification but are also very fast to compute. Once the integral-image is computed, feature computation for each pixel is just few subtraction operations. We express the computation of *Horizontal Projection Profile* (HPP) and *Vertical Projection profile*

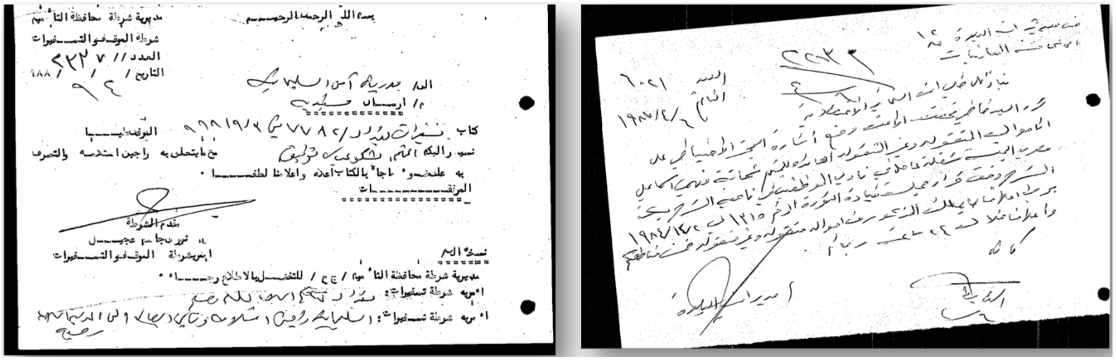


Figure 1.6: Sample document images with hand-written text-lines from Anfal dataset.

(VPP) around each pixel in terms of integral image, thereby making the computation very fast. In this thesis, we also introduce a novel scheme for evaluating noise removal algorithms using a constructed data and we use it to assess the quality of our rule-line removal algorithm [21]. We test our approach on both constructed and real-world handwritten Arabic document images which contain pre-printed horizontal and vertical rule-lines. Our experiments show that our approach is effective and computationally feasible even for high-resolution (600 dpi) document images. Details of this work are presented in Chapter 2.

1.2 Handwritten Text-line Segmentation

Text-line segmentation is an important step for many document processing tasks such as character and word recognition [15], layout-analysis [22, 23] and skew estimation [24, 25]. Many popular OCR systems are based on HMMs, and use the sequence of features extracted from vertical slice of text-line [15]. These systems require as input horizontal de-skewed text-lines which must be segmented with *high* *accuracy.* Errors in segmentation step usually result in a significant drop in the recognition accuracy.

1.2.1 Problem Definition and Challenges

For handwritten text, the text-line segmentation is even more difficult due to its free style nature, character size variations and non-uniform spacings between components. Moreover, the touching of characters across lines and overlapping spatial envelopes of text-lines make the problem more challenging. Methods previously developed for segmentation of printed text-lines do not adapt well to these scenarios. Recent work has focused on addressing each of these issues individually but a unified framework to take into account all the challenges associated with handwriting is still desired. For example, methods based on level-sets [26] are effective but computationally slow, methods based on connected-components(CCs) are fast but are challenged by touching components and overlapping lines [27]. Similarly, projection-based methods [28] cannot handle overlapping lines or touching, and perform poorly when there is large variation in character or word dimensions.

For Arabic, in particular, the presence of diacritical components significantly complicates the task. Figures 1.6 and 1.7 show some sample Arabic document images from the Anfal and GALE datasets respectively. Many existing approaches developed for Latin script do not adapt well to Arabic due to high variation in character dimensions and presence of diacritic/accent components. Figures 1.8 and 1.9 demonstrate some challenges associated with text-line segmentation for Arabic

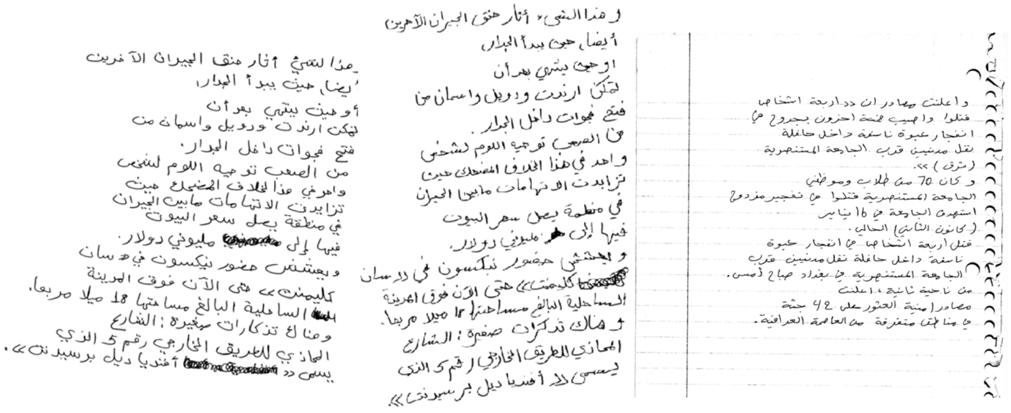


Figure 1.7: Sample images from Arabic GALE data (MADCAT project).

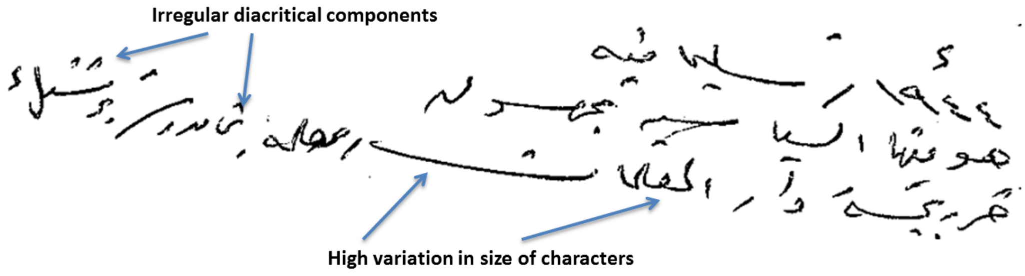
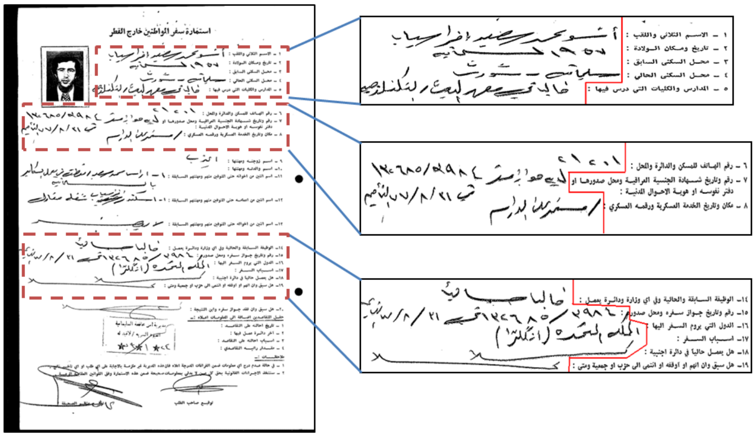

Irregular diacritical components
High variation in size of characters

Figure 1.8: A sample image demonstrating (a) High-variation in character sizes (b) Over-lapping spatial envelopes of text-lines (c) Non-uniform skew in text-lines (d) Presence of diacritical components



١ ـ الاسم الثلاثي واللقب :
٢ ـ تاريخ ومكان الولادة :
٣ ـ محل السكنى السابق :
٤ ـ محل السكنى الحالي :
٥ ـ المدارس والكليات التي درس فيها :

٦ ـ رقم الهاتف للمسكن والدائرة والمحل :
٧ ـ رقم وتاريخ شهادة الجنسية العراقية ومحل صدورها او
دفتر نفوسه او هوية الاحوال المدنية :
٨ ـ مكان وتاريخ الخدمة العسكرية ورقمه العسكري :

١٤ـ الوظيفة السابقة والحالية وفي اي وزارة ودائرة يعمل :
١٥ـ رقم وتاريخ جواز سفره ومحل صدوره :
١٦ـ الدول التي يروم السفر اليها :
١٧ـ اسباب السفر :
١٨ـ هل يعمل حالياً في دائرة اجنبية :
١٩ـ هل سبق وان اتهم او اوقفه او انتمى الى حزب او جمعية ومتى :

Figure 1.9: A sample document from Anfal dataset demonstrating the challenges involved in text-line segmentation. Both printed and handwritten text-lines are present whose properties and characteristics differ substantially.

Anfal images.

1.2.2 Contributions

We have developed a graph-based method for identifying unconstrained handwritten text-lines in document images. The challenge is to detect text-lines in presence of touching components, and text-lines having overlapping spatial envelopes. One obvious solution is to detect and correct such touching errors at the component level before applying any text-line segmentation, but this may be computationally expensive as the number of components in a document image may be large. We take another approach in which the detection and correction of such errors are delayed until an initial estimate of text-line segmentation is obtained (Figure 1.10). Each line is then checked for touching and proximity errors. This is computationally more efficient as the number of lines detected are far less than the number of components in a typical document image.

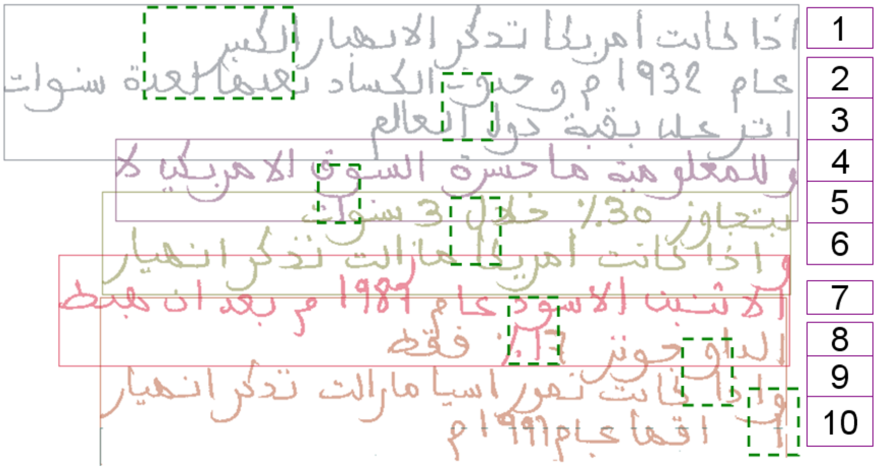


Figure 1.10: Text-line segmentation errors due to touching components across different lines. Line 1,2,3 are grouped as one segment due to touching components. Text-lines are color coded by the algorithm. Dotted boxes show the various touching component.

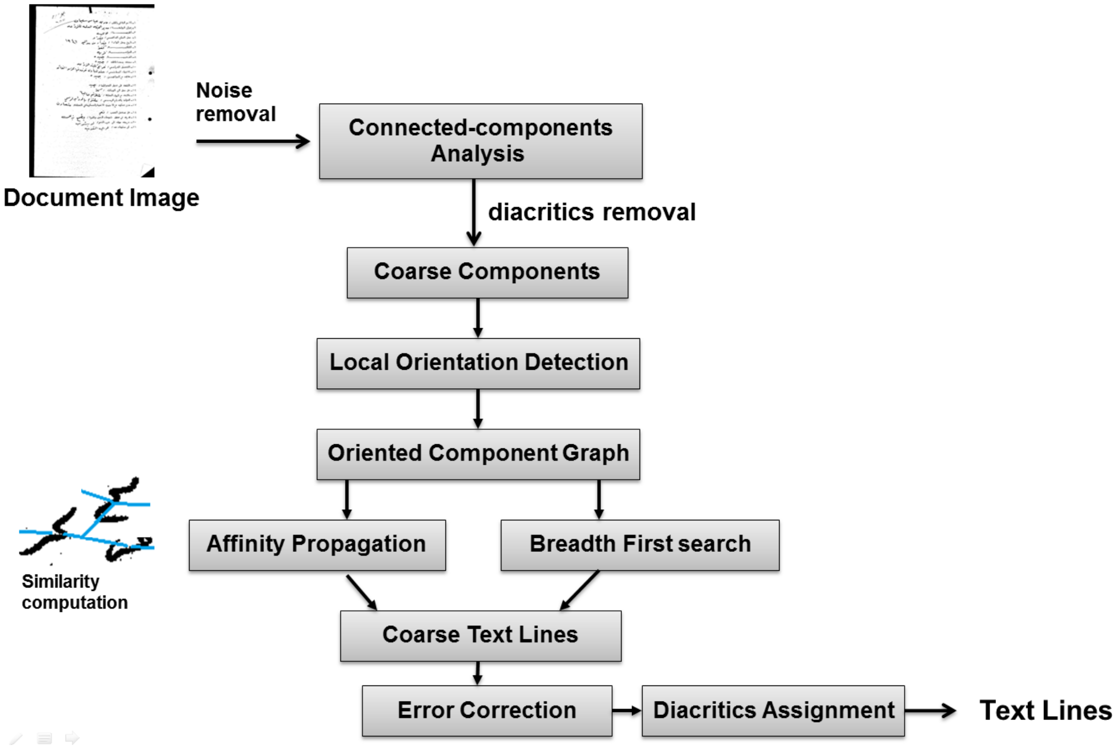


Figure 1.11: Block-diagram of our text-line segmentation method.

The block diagram of our text-line extraction method is shown in Figure 1.11. Our first contribution is a *local-orientation* and *shortest-path* based similarity measure between the components which provides an initial estimate of text-lines using a combination of Affinity-propagation [29] and Breadth-first-search method [30]. Our approach is adaptive to the dimensions of handwritten text and does not require any initial estimate of number of text-lines. Further, we present an iterative graph-based error detection and correction method to get the final estimate of text-lines. Unlike many previous approaches, our method is very general and allows any clustering/segmentation approach to be used for grouping text-components of any script such as Arabic, English, French, German or Greek, for example. Our error detection and correction method can be used as a post-processing step in any connected-component based method which gives an initial estimate of text-lines. We achieve a high-accuracy on associating a diacritic/accent component to a text-line which is crucial for processing Arabic documents. Additionally, our approach is faster than many previous methods.

1.3 Structural Similarity for Classification and Retrieval

Classification of document images into known categories is often a preliminary step towards recognition, understanding and information extraction [31]. Queries related to the information in document image databases can be greatly simplified if we know a priori the *genre* and *layout-type* of documents.

Many document classification approaches exist in literature which vary in their choice of features, representation and learning mechanisms. Structural-based similarity features have been shown to enhance the capabilities of content-based matching, and often provide an effective way to reduce the set of candidate documents for matching. However, mining layout structure (e.g., location and extent of components, spatial relationships among the components) in unconstrained and noisy documents has been difficult due to variation in content, translation, rotation and scale of components (Figure 1.13). Moreover, the meaningful information in a document's layout is often implicit in the global structure of the page.

The problem of retrieving similar document images from a large heterogenous collection has been of interest for many years [32, 33, 34, 35, 36, 37]. A large number of retrieval techniques have been developed using a query by example paradigm where features are extracted and indexed from document images off-line [34, 35, 36, 38]. A query image (e.g., words, logos, signatures) is provided, and features are extracted and matched against the indexed database of features (Figure 1.12). Documents which result in a number of matches above a certain threshold are considered relevant and can be geometrically verified [35, 38]. All these works emphasize the importance of using robust and scale-invariant descriptors for matching.

1.3.1 Problem Definition and Challenges

One of the the most important factors in developing a good classification method is related to defining the similarity between two images. There are numerous applications in office automation, litigation support and general document image search which could benefit from an efficient and effective method for computing similarity between images. Although methods have been developed for layout-specific or content-specific document image matching, a general approach which can detect *salient structures* and *co-occurrence relationships* among different regions of a document image automatically, is still being researched. Content-based approaches are highly dependent on and sensitive to the quality of optical character recognition (OCR) or component labeling classifiers. In cases of handwritten documents, these approaches may not be applicable since OCR for unconstrained handwritten documents is still a difficult problem.

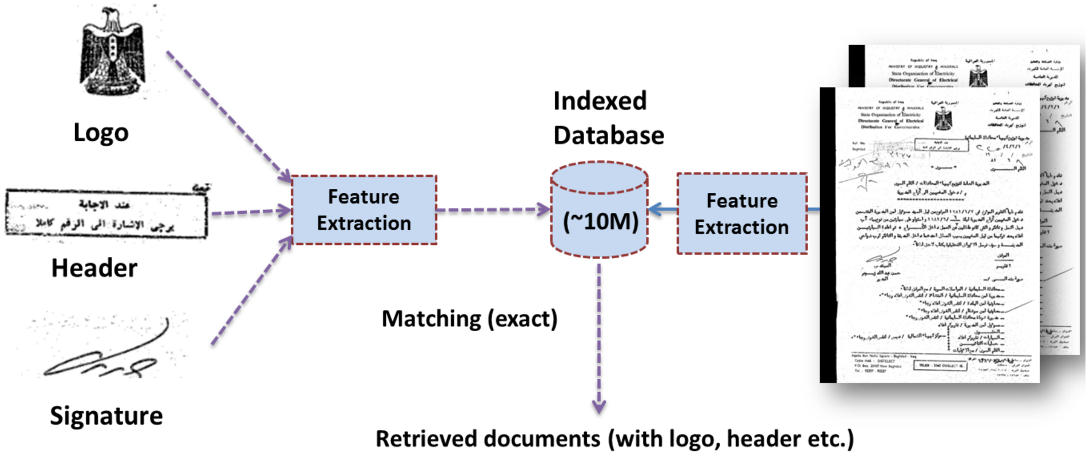
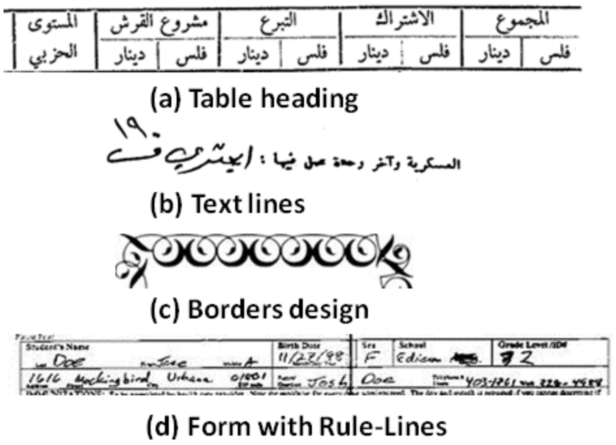


Figure 1.12: An example of retrieval setting in which logo, header, signature is used as query for retrieving documents.

Another challenge in formulating the classification and retrieval problems for document images in this way is the imbalance in training data. Often, the number of relevant documents provided for retrieval is much lower than the number of irrelevant documents causing an *imbalance* in the training data. Many learning methods suffer


(a) Table heading

(b) Text lines

(c) Borders design

(d) Form with Rule-Lines

Figure 1.13: Examples of document objects

from an *imbalanced* data problem [39].

Our approach defines *similarity* based on the layout/structure of whole document, and uses general features (not class-specific) extracted from a small set of user-provided set of *examples* to retrieve documents. Figure 1.14 shows an example of retrieval setting in which our approach can be applied. Our goal is to develop an approach which uses very few examples (typically fewer than 5) provided by user for learning a model for classifying document images in large databases.

1.3.2 Contributions

We present a method for the retrieval of document images with chosen *layout characteristics*. It is often not clear what features are best suited for monochromatic document images. To address this issue, we explore unsupervised feature learning and use raw-image patches and speeded-up robust features (SURF [40]) to construct a codebook representative of basic structural elements in document images. Our features are general (i.e. not specific to a class) and are based on statistics of

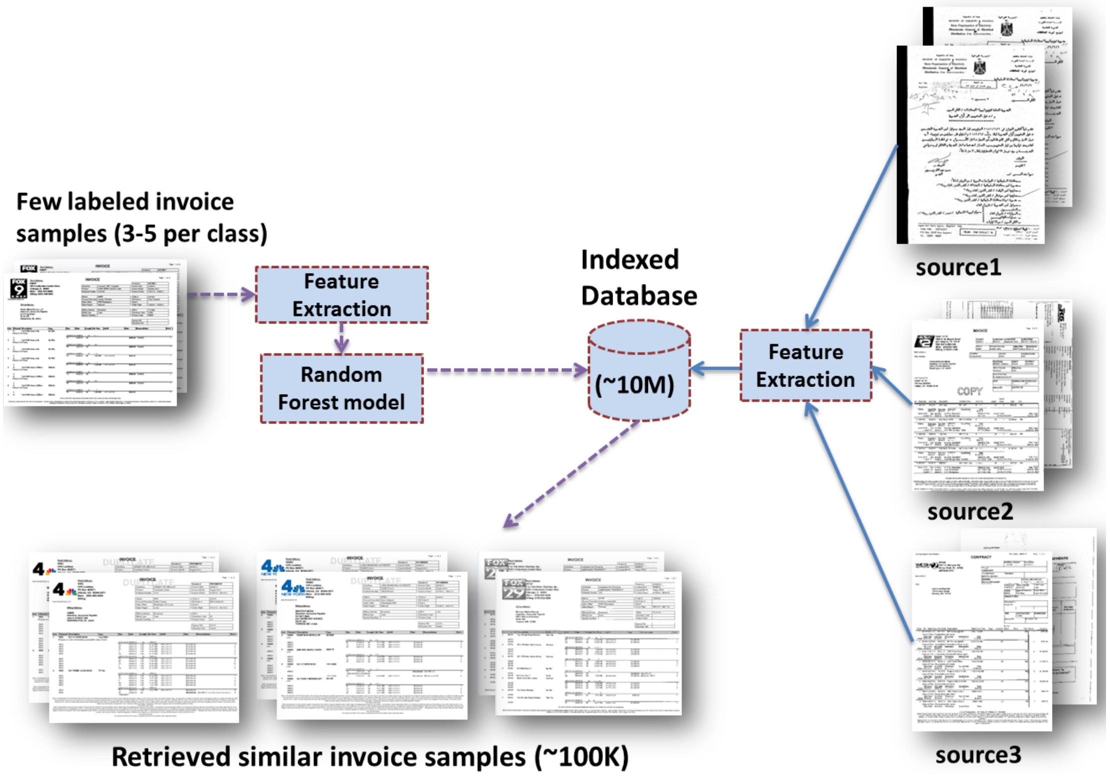


Figure 1.14: Example of a retrieval setting in which user provided examples are used to learn a model for retrieving/classifying documents.

quantized SURF descriptors over different regions of the image. To model the spatial relationships between codewords, the image is recursively partitioned horizontally and vertically, and a histogram of codewords is computed in each partition. The resulting set of features gives high precision and recall for the retrieval of hand-drawn and machine-print table-documents, and unconstrained mixed form-type documents, when trained using a Random forest (RF) classifier. Previous approaches have used non-overlapping partitions for modeling contextual information. In contrast, we allow overlapping partitions and learn important structures in document images using the *importance plots* obtained from RF. We compare our method to the *spatial-pyramid* scheme, and show that our approach for learning layout characteristics is more effective for document images when the labeled data is limited to a few examples.

Our approach differs from previous approaches in several ways: (1) we apply unsupervised feature learning to obtain a dictionary of representative structural units of document elements, (2) we use a horizontal-vertical partitioning scheme for learning spatial relationships, and (3) using the *importance* estimates of variables in a particular region, we learn *unimportant* partitions, and do not compute features over those regions. This results in computational efficiency, and in some cases, better performance. We compare RF with SVM and show that it is competitive for this problem even when the data is imbalanced.

We further extend our approach to a completely unsupervised setting, for grouping structurally similar document images (Figure 1.15). For this work, we extract SURF descriptors from a smaller set of representative images to construct a dictionary. Then, the image is recursively divided into vertical and horizontal partitions and histograms of dictionary atoms are computed for each partition as in previous case. To learn different structural patterns and correlation among features in data, we train a random forest classifier against the randomly-sampled auxiliary data. The learned trees in RF are then used to compute similarities between images. Our experimental results show that our approach for similarity provides an effective way to estimate the number of classes and group structurally similar images. Using four real-world datasets we show the effectiveness of our document clustering method.

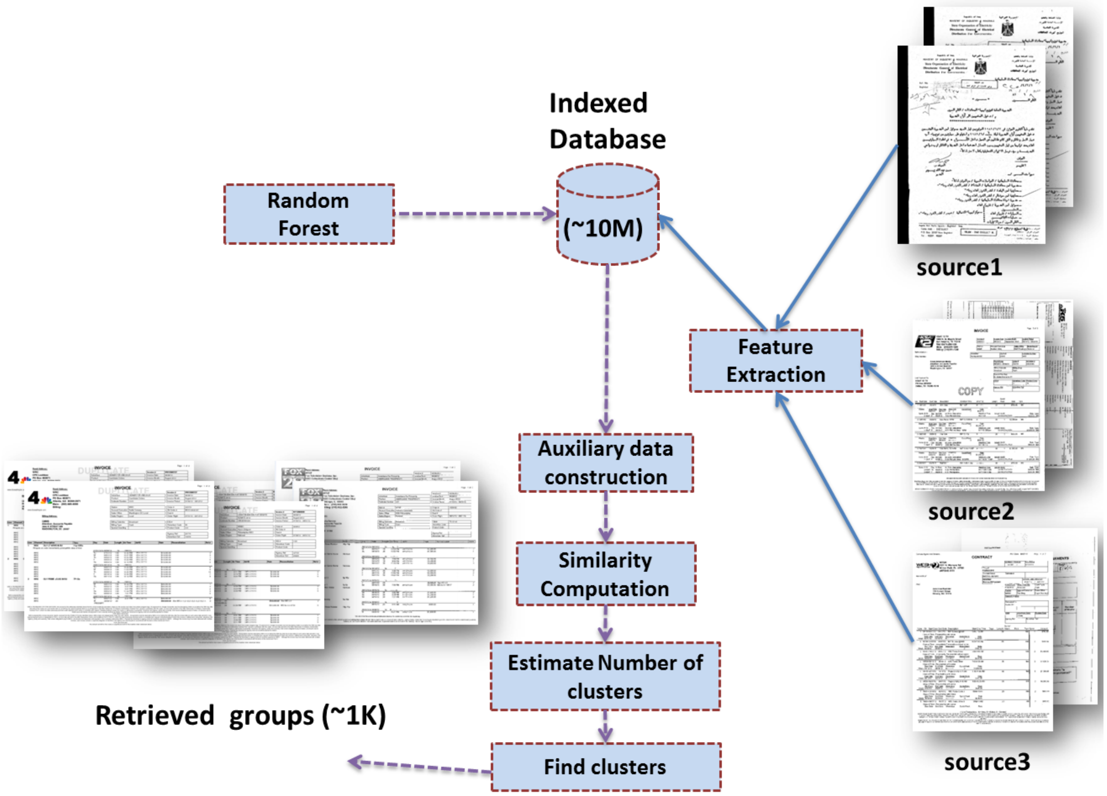


Figure 1.15: Example of a setting in which documents in a large database is grouped and classified without any labeled data (completely unsupervised). The number of clusters is estimated prior to applying clustering.

1.4 Organization

The thesis is organized as follows. Each module is described in one chapter along with its evaluation and experimental results. Chapter 2 describes the data-selection approach for SVMs and its application to the problem of rule-line removal in handwritten documents. Our handwritten text-line detection, segmentation and error-correction work is described in Chapter 3. This is followed by structural similarity based retrieval and classification approach in Chapter 4 and Chapter 5. A thesis summary, its contributions and possible future work in these areas are discussed in Chapter 6.

Chapter 2: Data selection for SVMs and Applications

In this Chapter, we assess the computational feasibility of selecting a set of most informative points prior to SVM training in order to reduce the overall training time. Even with a very few document images for training, the number of pixel level samples can be on the order of millions (100 images x 300 dpi x 8.5" x 11"). Support Vector machine (SVM)[14], due to its quadratic time ($O(n^2)$) dependency on the size of data, cannot handle such large training sets. In the formulation of SVMs, however, the decision function is fully determined by a small subset of the training data, called Support Vectors (SVs) [14] and hence for large data sets, one alternative is to first select points which are likely to be SVs, reducing the size of training set, and then train the SVM using only selected points.

One important property known to the researchers in this area is the relationship between SVs and the convex hulls of datasets containing the classes [41]. More specifically, in the separable case, if we only use the points in the convex hulls of different classes for training SVM, the solution obtained will be exactly the same as the one obtained by using the whole dataset [42]. The challenge is identifying these points. Many existing methods use the relationship with convex hull for obtaining a much smaller subset of training data. Since the complexity of finding the convex

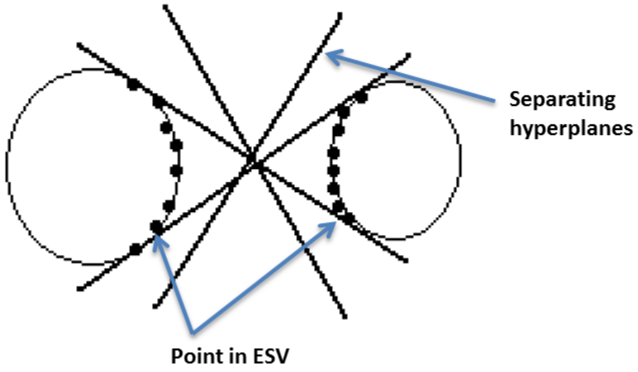
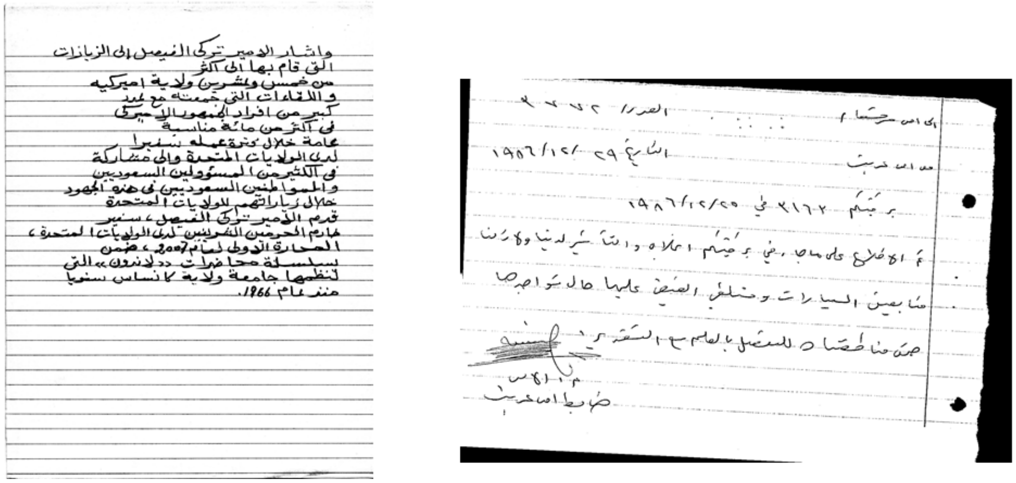


Figure 2.1: An illustration of Extended Support Vectors (ESV) in black dots.

hull has an exponential dependence $((\theta(n^{\lfloor d/2 \rfloor})))$ on the dimensionality (d) of the feature space, the methods based on finding the exact convex hull are computationally infeasible.

For the binary classification problem, however, the convex hulls of the two classes contain many redundant points. The final set of SVs belongs to a much smaller subset of the convex hull, known as the Extended Support Vector (ESV) set [43]. By ESV, we refer to the minimal subset of training instances that determine the space of separating hyperplanes, which cannot be ruled out as possible candidates on the basis of the examples processed at any stage in the learning process (Figure 2.1). It is well understood that the points which do not belong to the ESV set can never become SVs for binary classification. Hence, as a novel extension of previous approaches, we present a method to select points from the ESV set in order to do large scale SVM training. One major advantage of doing this is that the number of redundant points in the obtained set is minimized efficiently and each point is a good candidate for the final SV. This reduces the training size very effectively as



واشار الامير تركي الفيصل الى الزيارات
التي قام بها الى اكثر
من خمس وعشرين ولاية اميركيه
واللقاءات التي جمعته مع عدد
كبير من افراد الجمهور الامريكي
في اكثر من مائة مناسبة
عامة خلال فترة عمله سفيرا
لدى الولايات المتحدة والى مشاركة
في الكثير من المسؤولين السعوديين
والمواطنين السعوديين في هذه الجهود
خلال زياراتهم للولايات المتحدة
قدم الامير تركي الفيصل، سفير
خادم الحرمين الشريفين لدى الولايات المتحدة،
المحاضرة الدولية لعام 2007، ضمن
سلسلة محاضرات «لاندون» التي
تنظمها جامعة ولاية كانساس سنويا
منذ عام 1966.

الى امن [illegible] العدد / [illegible]
التاريخ ٢٩/١٢/١٩٨٦
مدير امن غريب
برقيتكم ٢١٦٢ في ٢٥/١٢/١٩٨٦
تم الاطلاع على ما جاء في برقيتكم اعلاه واننا نشير لدينا ولا زلنا
نتابع السيارات ومطلوب القبض عليها حال تواجدها
ضمن مناطقنا للعمل بالعلم مع التقدير
ضابط امن غريب

Figure 2.2: Sample document images with rule-lines

compared to the previous approaches based only on convex hull approximation.

As an application of our data selection approach we consider the pixel-level rule-line removal problem in document images. The rule-lines significantly reduce the accuracy of subsequent document processing tasks, especially if the task is to be performed on a monochromatic document image where there is no significant contrast between foreground text and underlying rule-lines [16, 44, 45, 46]. For example, in OCR systems, the background rule lines interferes with the foreground pixels and therefore cause inaccuracies during segmentation and/or feature extraction processes, which leads to poor recognition performance. Most of the previous work on this problem avoided pixel-level classification due to a high computational cost, and took heuristics based approach for rule-line removal [16, 17, 45, 46].

In this Chapter, we present a fast and effective method for removing preprinted rule-lines in handwritten document images (Figure 2.2). We use an *integral-image* representation which allows fast computation of features and apply our approach for large scale Support Vector learning using a data selection strategy to sample a small subset of training data. We express the computation of *Horizontal Projection Profile* (HPP) and *Vertical Projection profile* (VPP) around each pixel in terms of integral image, thereby making the subsequent computations very fast. We test our rule-line removal algorithm using constructed and real-world handwritten Arabic document images which contain pre-printed horizontal and vertical rule-lines. Our experiments show that the proposed approach is effective and computationally feasible even for high-resolution document images. The idea presented in this work is generic and can be applied to any problem where pixel-level feature computations and large scale SVM training are required.

2.1 Related Work

2.1.1 Data Selection for SVM

The typical approach taken by SVMs is to solve a quadratic optimization problem with linear constraints [47]. To solve the high-dimensional quadratic programming (QP) problems for the SVM, special algorithms have been developed by exploiting the sparsity of the SVM solution and the Karush-Kuhn-Tucker optimality conditions. These methods solve smaller QP problems with a selected subset of the data and iteratively add examples which violate the optimality conditions. One of the recent and most popular is Platt's SMO [48]. One major drawback of this approach is that it is still necessary to solve QP problems multiple times with an increasing number of Support Vectors (SVs). Furthermore, these training methods make use of the whole training set. This results in a very long training time for large datasets (which can go up to an order of days for 100 million points).

Zhang and King [49] proposed a β-skeleton algorithm to identify support vectors which also used the convex hull property to connect SVs and β-skeleton. Due to $O(n^3)$ time complexity their algorithm is not applicable to large training sets. Abe and Inoue [50] estimate boundary points using Mahalanobis distance. They express the decision boundary of classifier using Mahalanobis distance by a quadratic polynomial thereby approximating the boundary data. The time and space complexity of their approach is quadratic in training data size due the computation of covariance matrix. In [51], Lee and Mangasarian randomly select a subset of the data that is typically 10% or less of the original dataset to obtain a nonlinear separating surface for the *ReducedSVM* training. Huang and Lee [52], showed that uniform random sampling is the optimal robust selection scheme in terms of several statistical criteria. Wang et al. [53] proposed methods based on a statistical confidence measure and Hausdorff distance which is motivated by the geometrical interpretation of SVMs based on the reduced convex hulls.

2.1.2 Rule-line Removal

Existing approaches for rule-line removal can be broadly classified as *heuristic-based* [16, 17, 45, 46] or *model-based* [44, 54]. In *heuristic-based* approaches, rule-lines are detected and removed using *Projection profiles* [16, 17], *Hough-transform* [45], *Run-lengths* or *Morphological operators* [46].

For *Projection profile* based methods, a horizontal histogram is used to locate the center locations of horizontal rule-lines [16, 17]. These methods are very sensitive to the skew of the document image and often have difficulty in estimating the accurate thickness of line. To overcome these problems, Cao *et al.* partitioned the image into vertical zones and projection profiles were computed for each zone [16]. But finding the optimal width of vertical zone is difficult and empirical.

Hough-transform based methods detect lines based on peaks in parameter-space also known as Hough-space. They can detect broken-lines but may fail to accurately localize rule-lines with varying thickness. To address this problem, Chen and Lee proposed the *strip projection* method motivated by the fact that lines are more likely to form peaks in a small region [45]. But one of the main drawbacks of *Hough-based* methods are that they are computationally slow. *Morphological operator* based methods use a structuring element to remove rule-lines by dilation and erosion operations [46]. The design of accurate structuring elements often depends on the width of the rule-line and the strokes of characters. These methods are incapable of removing rule-lines with large variation in thickness. Shi *et al.* used directional local profiling followed by adaptive vertical run-length search to remove rule-lines in Arabic documents [55]. Although the method seems to work reasonably well, it is susceptible to remove text pixels in overlapping regions, thereby degrading the quality of text. Abd-Almageed *et al.* proposed a *linear-subspace* based method to detect rule-line pixels in binary images [44]. The computation of central-moment features used in their approach is very time-consuming for all the foreground pixels and makes the method infeasible for high-resolution document images.

2.2 Approach

In this Section, we first discuss our data selection method for training SVMs. We then present an application of our data selection on pixel-level rule-line removal in document images. We present our integral image based formulation of computing HPP and VPP features followed by text/ruleline classification approach in subsections.

2.2.1 Data Selection for Support Vector Learning

We first explain our randomized method for selecting points from the convex hull of data points. This method can be used for training a one-class SVM, where labels are known only for one class [56]. We then present our second method which uses incremental subspace learning for selecting points from the ESV set of two classes.

2.2.1.1 Data Selection from the Convex Hull

Let $P \in \mathbb{R}^{DxN}$ denote the set of N training points in a D dimensional space. We initialize a subspace S_d of dimension $d = 0$ with a normalized vector $[v_k / \parallel v_k \parallel]$ using a randomly selected point $v_k \in P$. We use *incremental subspace learning*, which has been extensively used in pattern recognition and computer vision [57].

$$S_0 = \left[\frac{\mathbf{v_k}}{\parallel \mathbf{v_k} \parallel} \right] \tag{2.1}$$

The idea is that one starts with a subspace of dimension zero with a single point($\mathbf{S_0}$) and incrementally adds points from the training set using the following strategy. If the newly selected point lies in the current subspace, then we do not do anything and find another point. If it does not, we create a new subspace which has dimension one higher than the previous subspace ($\mathbf{S_{d+1}}$), to accommodate the new point. For this we first compute the projection $\mathbf{p}$ of the new vector $\mathbf{v_i}$ on current subspace S_d:

$$\mathbf{p} \leftarrow {S_d}^T \mathbf{v_i} \tag{2.2}$$

where the superscript (T) represents to the transpose of matrix S_d. In the next step we compute the reconstructed ($\mathbf{r_v}$) and the residual vector ($\mathbf{v_{res}}$) as follows :

$$\mathbf{r_v} \leftarrow S_d \mathbf{p} \tag{2.3}$$

$$\mathbf{v_{res}} \leftarrow \mathbf{r_v} - \mathbf{v_i} \tag{2.4}$$

$$S_{d+1} = \begin{bmatrix} S_d & \dfrac{\mathbf{v_{res}}}{\parallel \mathbf{v_{res}} \parallel} \end{bmatrix} \tag{2.5}$$

Since the subspace $S_{d'-1}$ divides the next higher subspace $S_{d'}$ into two halves, we select two points for $S_{d'-1}$, one from each side. This is done so as to guarantee that every point in the convex hull has a nonzero probability of getting selected, as explained in Claim 2 below. Finally, we obtain a subspace containing all the points.

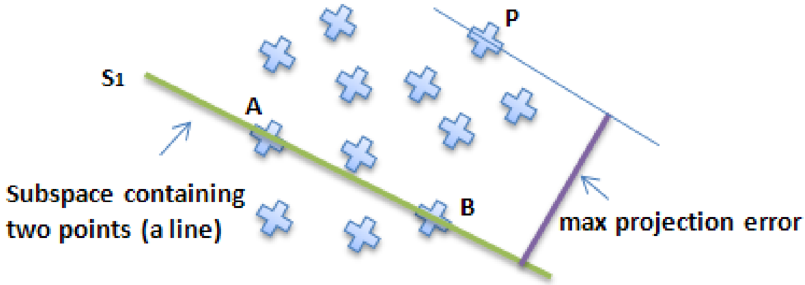


Figure 2.3: S_1 is a subspace containing points A and B. The point with maximum projection error (P) lies on the Convex-hull of points.

Our method uses this strategy to select points, since for every new subspace, the point not contained in the subspace and having the maximum projection error (ϵ_i) will lie on the Convex-hull of training points of each class (Figure 2.3).

$$\epsilon_i = d(\mathbf{r_v}, \mathbf{v_i}) = \sqrt{\sum |r_v - v_i|^2} \tag{2.6}$$

$$\mathbf{v_p} = \{\mathbf{v_i} : \epsilon_i > \epsilon_k \quad \forall k\} \tag{2.7}$$

where $d(\mathbf{r_v}, \mathbf{v_i})$ represents the Euclidian distance between vectors $\mathbf{r_v}$ and $\mathbf{v_i}$, and $\mathbf{v_p}$ is the point with maximum projection error. For the non-linearly separable case we replace the dot-products by kernel function and use slack-variables to sample points in feature space. Algorithm 1 in the Appendix A provides the pseudo-code for implementation.

Claim 1. *Let $P \in \mathbb{R}^{DxN}$ be the given set of N training points in the feature space of dimension D. Let CH_P represent the actual Convex Hull of the set of points in P and CHS be the set of points returned by Algorithm 1. Then, $CHS \subseteq CH_P$.*

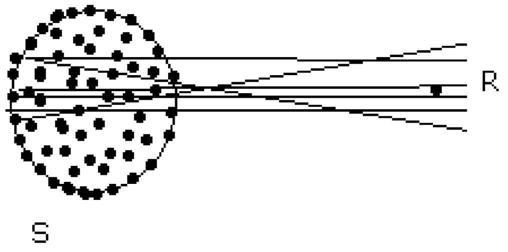


Figure 2.4: R is more important than the other points in the Convex-Hull because if we miss R, the approximation error of computing the convex hull will be much higher than the corresponding errors obtained from missing the other points.

Proof. See Appendix A. □

Claim 2. *Let T be any arbitrary point in CH_P i.e. the actual convex hull of the points. Then, there exists a non-zero probability of choosing T by Algorithm 1.*

Proof. See Appendix A. □

Our strategy of selecting points from the convex hull has two main advantages. The first advantage is that the *more important* points in the convex hull have a higher probability of getting selected than the *less important* points. Here, we refer to *more important* points as those points which preserve the shape of the original convex hull better than the other points. For example, consider the case when the data is spatially skewed and one point, namely R, lies very far from the remaining points in the set S (Figure 2.4). R is more important than the other points in the CH because if we miss R, the approximation error of computing the convex hull will be much higher than the corresponding errors obtained from missing the other points. Now, based on the distance between two points, the probability of selecting R is very high because at the first step, the probability of selecting a point in S is high and for all the points in S, the farthest point in whole dataset is R. Similarly, consider creating random lines (d=1) by choosing two points randomly from the entire dataset. Again, the probability of obtaining a line for which R is selected is high. Another advantage is that as we incrementally build high dimensional subspaces, the number of points for which the projection error needs to be computed decreases rapidly. This is due to the fact that higher order subspaces of data would contain most of the points. This makes the method computationally more feasible.

Time complexity: If we consider the basic operation to be the computation of the projection error, then in the worst case, we may have to compute this $O(N)$ times for all d' subspaces, resulting in a total running time of $O(d'N)$ for d' points. Hence, to obtain K points, we get $O(KN)$ computations as d' gets canceled. K is asymptotically $O(N)$ which makes the overall complexity quadratic. Hence, for large N, we first apply a fast clustering approach (fast K-means [58], Filtering method [59]) to obtain K_1 exemplars, and then hierarchically find the points with maximum projection error. This reduces the time complexity to O(K_1) + O(K_2), where K_2 is the average number of points in each cluster. Also, in general, K would be much smaller than N (10%-30% of N), leading to a smaller number of actual computations than $O(N^2)$. Since we compute the projection errors only for those points that are not in the subspace, the N in the second term decreases rapidly as d grows. This effectively makes the number of computations required much smaller. Another important property of the method is that the K iterations can be parallelized. This gives a linear parallel time complexity for smaller values of K. The space complexity of our method is linear in training data size.

2.2.1.2 Data Selection from the Extended Support Vector set

In this section, we extend the idea of random subspaces to sample data points from the ESV set for binary classification. In this case, we select points using only subspaces $S_{d'-1}$ of dimension $d'-1$, where $d' \leq D$ is the underlying subspace of the training points belonging to a particular class. Algorithm 2 in Appendix A provides the pseudo implementation details (ESVsample).

Linearly Separable Case: We denote the training sets of class 1 and class 2 by $P \in \mathbb{R}^{DxN1}$ and $Q \in \mathbb{R}^{DxN2}$ respectively. It is known that any subspace of dimension one less than the current subspace (S_d) divides it into two half-spaces. We denote such half-space of P towards Q by P_Q and half-space of Q towards P by Q_P. We first incrementally build subspace $S_{d'-1}$ of P and find the point $v_P \in P_Q$ with the maximum projection error e_{maxP} to $S_{d'-1}$. We compute the projection errors to $S_{d'-1}$ from all the points in Q to find the point v_Q with the minimum projection error (e_{minQ}).

We then check if e_{minQ} is at least as big as e_{maxP}. If this condition is satisfied, then the selected points $v_P \in P$ and $v_Q \in Q$ must belong to the ESV set. This is explained more formally in Claim 3. Figure 1(c) in the Appendix A shows this condition pictorially. Algorithm 2 summarizes the main steps required for implementation. The given pseudo code demonstrates the selection of points using random subspaces of P. However, the same idea can be applied to select points using the subspaces of Q. Like Algorithm 1, we allow reconstruction error up to a threshold(C).

Time and Space Complexity: Analyzing the time complexity reveals that even for small values of K, the number of possible d-dimensional subspaces is $\binom{N}{d}$ and N is a very large number. Hence, to reduce this combinatorial complexity, we first randomly select smaller set of samples and then use this as input to the second method. It reduces the time complexity to $\binom{K_1}{d}$. This can be done due to the fact that ESV is a subset of convex hull. Another alternative is to run a fast clustering algorithm ([59]) and use the cluster centers first instead of all points. In practice, random sampling to reduce the large dataset before selection works as well as clustering based reduction.

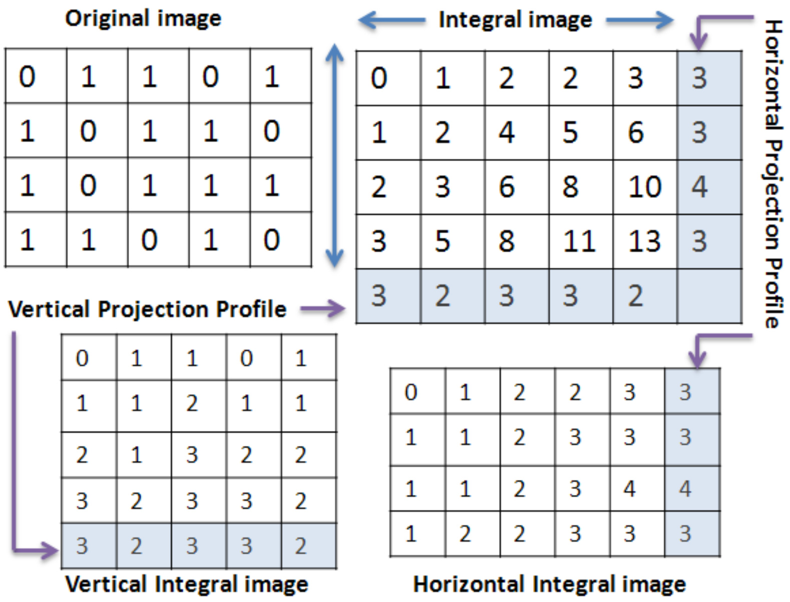
Non-linearly Separable Case: A non-linear SVM gives a decision function $f(\mathbf{x}) = sign(g(\mathbf{x}))$ for an input vector $\mathbf{x}$ where $g(\mathbf{x})$ is given by Equation 2.8:

$$g(\mathbf{x}) = \sum_{i=1}^{l} w_i K(\mathbf{x}, \mathbf{z}_i) + b \tag{2.8}$$

where $\mathbf{z}_i$ are representatives of training examples called support vectors. $K(\mathbf{x}, \mathbf{z}_i)$ is a kernel that implicitly maps vectors to a higher dimensional space. We use the kernel trick to replace the dot products in Algorithm 2 by kernel function $K(\mathbf{x}, \mathbf{y})$ given by Equation 2.9 for non-linearly separable data. This leads to subspace sampling in kernel space instead of input space.

$$K(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle \tag{2.9}$$

Claim 3. *Let $S_{d'-1}$ be any subspace of dimension one less than the maximum subspace dimension $d' \leq D$ of data points P to be sampled. Define $e_{maxP} =$*


Original image
Integral image
Horizontal Projection Profile
Vertical Projection Profile
Vertical Integral image
Horizontal Integral image

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

0	1	2	2	3	3
1	2	4	5	6	3
2	3	6	8	10	4
3	5	8	11	13	3
3	2	3	3	2	

0	1	1	0	1
1	1	2	1	1
2	1	3	2	2
3	2	3	3	2
3	2	3	3	2

0	1	2	2	3	3
1	1	2	3	3	3
1	1	2	3	4	4
1	2	2	3	3	3

Figure 2.5: The value at any point (x,y) of integral image is the sum of all pixel values above and to the left. If we compute horizontal and vertical integral image then any row and column sum can be computed in two array references.

$\{e_{iP} | e_{iP} \geq e_{jP} \quad \forall \quad v_{jP} \in P_Q\}$ *and* $e_{minQ} = \{e_{iQ} | e_{iQ} \geq e_{jQ} \quad \forall \quad v_{jQ} \in Q\}$ *where* e_j *is the projection error of* v_j *to subspace* $S_{d'-1}$ *and* $P_Q \subseteq P$ *is the set of points in* P *which lie towards points in* Q *with respect to* $S_{d'-1}$*. If* $e_{maxP} \leq e_{minQ}$ *then the corresponding points* $\{v_{iP}, v_{iQ}\} \in ESV$*.*

Proof. See Appendix A. □

2.2.2 Application: Rule-line Removal

Our first contribution in making a rule-line removal efficient is the integral-image based feature computation. We present the formulation of traditional features using an integral image in Section 2.2.2.1. We then present our approach for SVM based text/rule-line classification in Section 2.2.2.2.

2.2.2.1 Integral Image Features

Horizontal projection profiles and Vertical projection profiles can be computed very rapidly using an intermediate representation of the image known as an integral-image [20]. The value of integral image (ii) at the location (x,y) is the sum of pixel values above and to the left of (x,y) as demonstrated in Figure 2.5.

$$ii(x, y) = \sum_{x\prime \leq x, y\prime \leq y} i(x\prime, y\prime) \tag{2.10}$$

An integral-image can be computed in one pass over the original image. Using the integral image any rectangular sum can be computed in four array references. Moreover, any row sum or column sum can be computed in just two array references if we compute both the vertical (V_{ii}) and horizontal (H_{ii}) integral image (Figure 2.5).

$$H_{ii}(x, y) = \sum_{x\prime \leq x} i(x\prime, y), \quad V_{ii}(x, y) = \sum_{y\prime \leq y} i(x, y\prime) \tag{2.11}$$

$$HPP_k(c_1, c_2) = H_{ii}(k, c_2) - H_{ii}(k, c_1) \tag{2.12}$$

$$VPP_k(r_1, r_2) = V_{ii}(k, r_2) - V_{ii}(k, r_1) \tag{2.13}$$

where $HPP_k(c_1, c_2)$ represents the sum of k^{th} row between columns c_1 and c_2 and $VPP_k(r_1, r_2)$ represents the sum of k^{th} column between rows r_1 and r_2. As observed in Equation 2.12 and Equation 2.13 the computation of each sum takes one

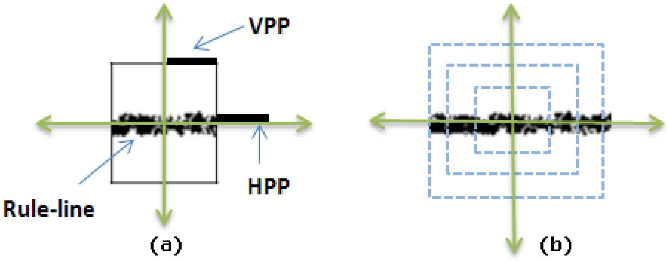


Figure 2.6: An illustration of feature computation. (a) Horizontal Projection Profile (HPP) and Vertical Projection Profile (VPP) in the first quadrant. Features from all the four quadrants are concatenated to form a feature vector at one scale. (b) Three different scales for computing features

subtraction and two array accesses.

We use the HPP and the VPP of the four quadrants around the pixel as features to train a two-class SVM. We concatenate these features at different scales to capture more context around each pixel. Figure 2.6(a) shows an illustrative example for HPP and VPP in the first quadrant of a rule-line image. Figure 2.6(b) shows the different scales considered for feature computation. If we consider a quadrant of size 16x16 then the computation of HPP and VPP requires 64 = (16+16)*2 array accesses instead of 256 = 16*16 accesses without integral-image.

2.2.2.2 SVM based Text-Ruleline Classifier

Using the features described in Section 2.2.1, we train a nu-SVM [60] classifier using the selected training points. The classification time for SVM depends on the number of SVs in the trained model. As more and more points lie near the boundary of separation of two-classes, the number of SVs also increases. To expedite the rule-line removal we train two SVM classifiers. The first classifier is trained to remove

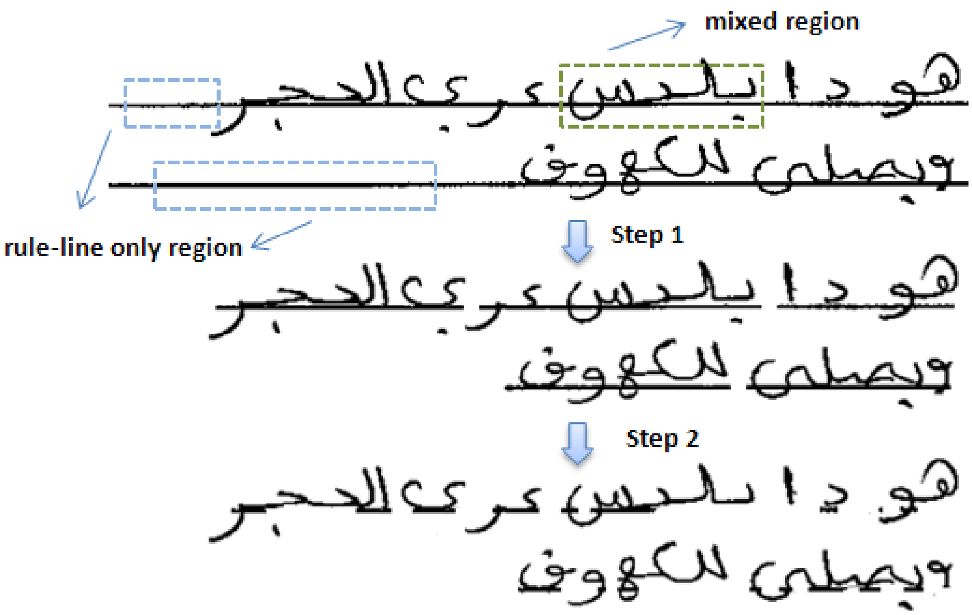


Figure 2.7: Rule-line removal is done in two steps. Two different classifiers are used to remove the rule-line only region and the mixed region in different steps.

rule-line pixels from *rule-line only* regions as shown in Figure 2.7. We only use the features extracted at one scale (4x4 quadrants) to train this classifier. The second classifier is used in second step to remove rule-lines from the *mixed-region* (Figure 2.7). In this stage we compute features from three windows at different scales (4x4, 6x6, 8x8 quadrants). This is computationally more efficient because the first classifier is less computationally intense with much fewer SVs. Another advantage of detecting the rule-line pixels in first pass is that using those pixels we can estimate the parameters of each rule-line and use it to restrict the number of pixels to be classified in second pass.

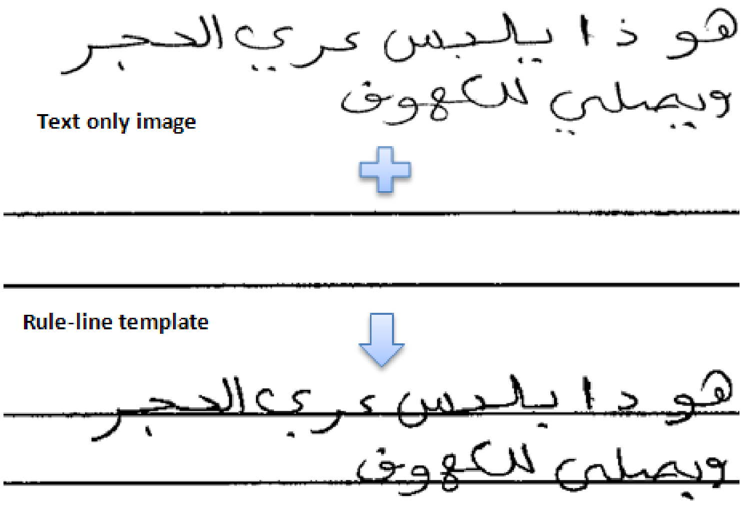


Figure 2.8: Creation of constructed data from a text-only image and a rule-line template.

2.3 Experiments

We conduct three sets of experiments to validate our data selection and rule-line removal approach. In the first experiment, our objective to show that our data sampling approach can effectively reduce pixel-level large-scale datasets without sacrificing much accuracy. In the second experiment, the objective is to compare our data selection method with two previous approaches on two standard datasets from UCI repository [61]. In the third, the objective is to demonstrate the ability to perform efficient pixel-level rule-line removal in high-resolution document images.

2.3.1 Competing Approaches

In our first experiment on large-scale pixel-level dataset, we compared our approach with a Reduced-SVM [51] method available in LibSVM package. Reduced-

SVM [51] uses a randomly selected subset of the data to obtain a nonlinear separating surface for classification. We call our approach *ESVsample* and denote Reduced SVM by ReducedSVM. For rule-line removal we compare results of our approach with results obtained using a subspace based approach [44]. We also report results based on data points selected using ReducedSVM.

For evaluating the effectiveness our data selection approach, we compare with two previous approaches based on a *Confidence measure* and a *Hausdorff-distance* [53]. In the first approach, the confidence measure of a training point is estimated by the number of training examples that are contained in the largest sphere centered at that training example without covering an example of a different class. Second approach determines whether a training example is likely to be a support vector by computing the distance to the convex hull of the training examples of the opposite class. We used the results reported in their work to compare our approach against the approaches. For each data selection method an increasing number of training points are selected from the data and error rates are computed for comparison.

For time performance, we compare our approach with the same subspace-based approach [44] mentioned above. The moment based features used in their work do not use integral-images, and hence pixel-level feature computation is time intensive. We also compute performance of our method without using integral-image features to show the computational advantages of our features.

2.3.2 Datasets

Evaluating methods for pixel-level content separation requires pixel-level annotation of images. Such annotation is difficult to obtain manually for large collections of document images. We therefore evaluate our approach using a constructed dataset proposed in [44] and a small set of real handwritten document images. Images in the first dataset were created by combining templates of rule-line images with the images containing handwritten text (Figure 2.8). It contains a total of 50 images each having a resolution of 600 dpi. Each image has approximately 2 million foreground (text/rule-line) pixels. Since the total number of foreground pixels in the 35 images of training set is too large for SVM training, we randomly sampled 100K points for applying data selection approaches. This serve as our dataset for first experiment. We also report results on a dataset of 10 real handwritten images with pre-printed rule-lines. The ground-truth for these images was created by removing rule-line pixels manually. Both the datasets are available for download at [62].

For comparison with other data selection approaches we used *Breast cancer* and *Pima Indians* datasets from UCI repository. These datasets were used in [53] for demonstrating the data selection approaches. The Breast cancer dataset consists of 683 examples from two classes [61]. Each samples has eight attributes. The size of the training set in each iteration is 547 and the size of the test set is 136. The Pima Indians dataset consists of 768 examples with each example having eight attributes [61]. The sizes of the training and test sets in each iteration are 615 and 153, respectively. Although, these datasets are not large, comparisons with previous approaches on these datasets show the quality of selected points for different methods.

2.3.3 Evaluation Protocol

We compute *recall* and *precision* values along with their harmonic mean (F_1 score) to evaluate our rule-line removal method. If a rule-line pixel detected by our method is also a rule-line pixel in ground-truth then it is counted as *true positive* (TP). Similarly, if a rule-line pixel detected by our method is not a rule-line pixel in ground-truth then it contributes to *false positive* (FP). Regions where rule-lines and text overlap are considered text. *False negatives* (FN) are those rule-line pixels which are missed by our algorithm. Using these values we compute *precision*, *recall* and F_1 score as follows:

$$Precision = \frac{TP}{TP + FP} \tag{2.14}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.15}$$

$$F_1 score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{2.16}$$

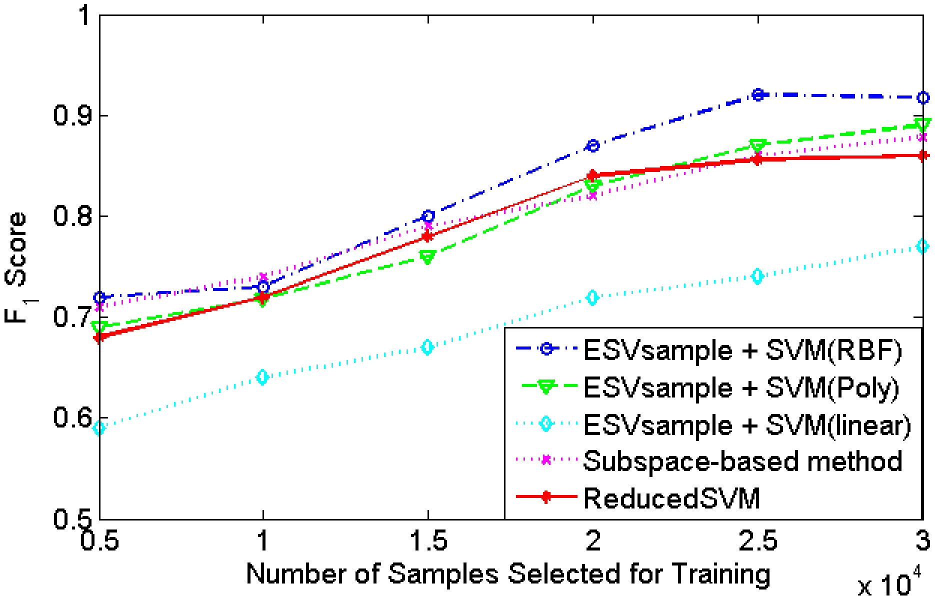


Figure 2.9: Plot of F1-scores on constructed data

2.3.4 Results and Discussion

In our first set of experiments a total of 15 test images were used to obtain the plots shown in Figure 2.9. An increasing number of points are selected from our first dataset (100K samples) using $ESVsample$ and $ReducedSVM$, and F_1 is computed on the test data. Since the methods involve randomization, we repeated our experiments five times and report the mean accuracies. For the two stages of rule-line classification, namely, rule-line only region and mixed region, data was selected to train two SVM classifiers. Using the selected points for both stages, SVM with a radial-basis kernel (RBF) achieved the best accuracy of 0.914 on test images. We did not observe any significant improvement in accuracy after selecting 30,000 points. The mean F_1 obtained using the complete set of 100K points for training was 0.922. The poor accuracy of the linear-kernel confirms that the two classes are

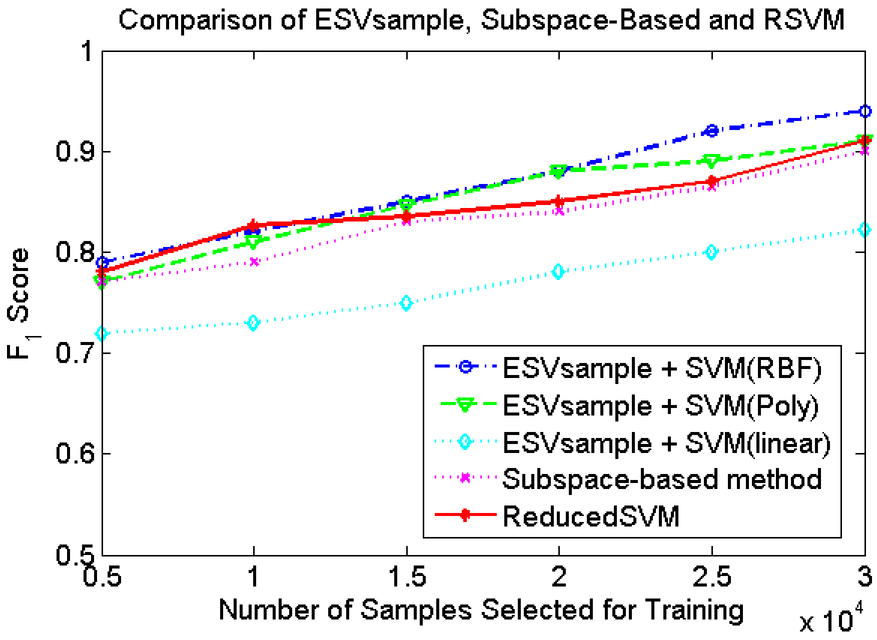


Figure 2.10: Plot of F1-scores on real-data

not linearly separable.

In order to compare our data selection strategy for large scale SVM learning, we used a Reduced-SVM [51] for training and obtained a similar plot of F_1 scores on test data. We selected the same number of training points using ReducedSVM as our method and obtained the F_1 scores. Our approach outperforms $ReducedSVM$ based data selection method showing that our selected set of points are more effective than those selected by $ReducedSVM$.

We also computed the F_1 scores using the same set of selected points for subspace-based method [44]. The better accuracy of $ESVsample + SVM(RBF)$ compared to subspace based approach justifies the use SVM compared to subspace learning based approach for rule-line removal.

We also evaluated our method on real handwritten images. A similar plot of

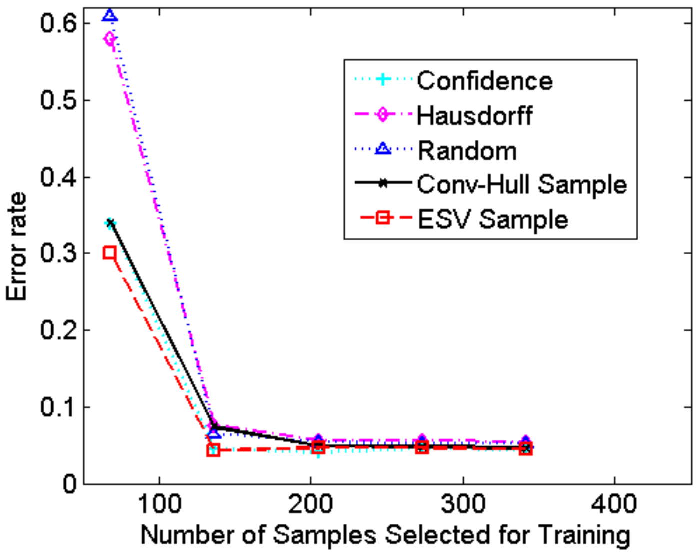


Figure 2.11: Comparison of data selection approaches on UCI Breast cancer dataset.

F_1 scores with increasing number of selected data points is shown in Figure 2.10. We used a test set of three images and a train set of seven images. Slightly better accuracy (94%) for the second dataset may be due to the fact that in real scenarios, the possible interactions between rule-lines and text is more structured and limited than random interactions in the constructed data. Misclassified pixels in our results are mainly from the mixed-regions where a clear-cut boundary between the rule-line and text is ambiguous.

Figure 2.11 and Figure 2.12 shows the comparisons with two additional data selection approaches based on *confidence* and *Hausdorff-distance* metrics [53]. Additionally, we compare with the random selection approach for reducing the training data. For each of the methods, a fixed set of data was selected from the full-set of available samples. The selected samples were trained and tested on the test data to obtain the error rates in plots. The mean error in classification using all the samples

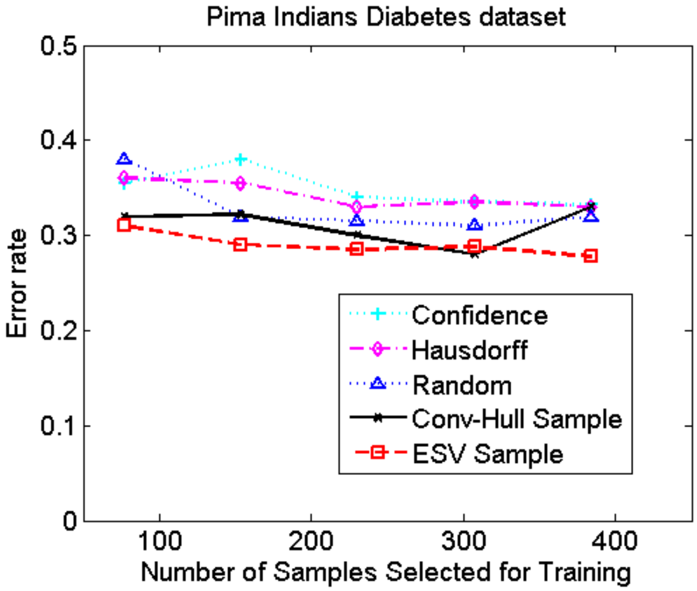


Figure 2.12: Comparison of data selection approaches on UCI Pima Indian Diabetes dataset.

in training set (547) for breast-cancer data is 0.035, while our method archives an error rate of 0.045 using just 140 samples. Similarly, a competitive error-rate is observed for Pima Indian diabetes dataset.

Time-performance: In Figure 2.13, we show the plots of time-taken for rule-line removal using MATLAB. As seen, there is a significant reduction in time using the integral-image features. All the experiments were conducted on a P4 machine with 3GB RAM. Table 2.1 provides the time taken by the data selection method (for 10K points) and SVM on the whole rule-line data set of 100K points. We are reporting the time for our first SVM model used for rule-line only regions (32 dimensional features). Our data selection method is implemented in MATLAB and the LibSVM executable used was a windows executable. We expect a further reduction time with C++ implementation of our approach. As observed, once the

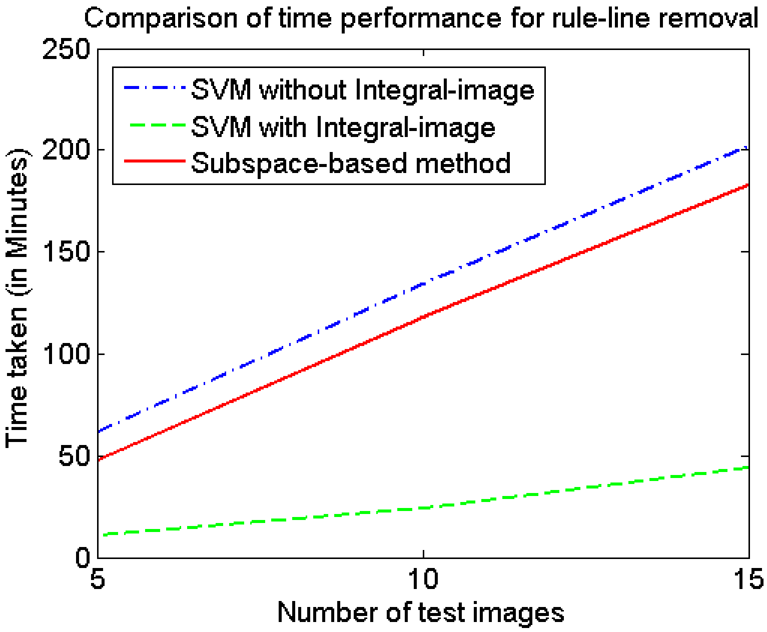


Figure 2.13: Comparison of time-performance on test data

Table 2.1: First term shows the time taken (in seconds) for data selection (ESVsample) and the second term is the time taken by LibSVM solver.

Data set	ESVsample + LibSVM (10K points)	LibSVM (100K points)
RuleLine	1084 + 72	3174

data set is reduced to a smaller set of important points the SVM solver takes much less time.

Chapter 3: Handwritten Text-line Segmentation

Text line segmentation is a critical preprocessing step in document analysis and is especially difficult for handwritten material. Text lines are crucial for analyzing the document layout, assessing the skew or orientation of a document and indexing/retrieval based on word and character recognition [63]. Although text line segmentation for machine printed documents is often seen as a solved problem, free style handwritten text lines still present a significant challenge [64, 65]. This is because handwritten text lines are often non-uniformly skewed and curved, have nonuniform space between lines and have spatial envelopes that may overlap. Irregular layout, variable character size originated from different writing styles, existence of touching lines and the lack of a well defined baseline also contribute to making handwritten document analysis more difficult [66].

Although projection based methods [67] have been successfully applied for machine-printed documents [68], variation in baseline and skew make them less effective for handwritten lines. Hough-based methods can handle documents with variation in the skew angle between text lines, but their performance also degrades rapidly when the baseline is not straight. Grouping based approaches use connected components(CC) to handle complex layouts, but due to proximity or touching char-

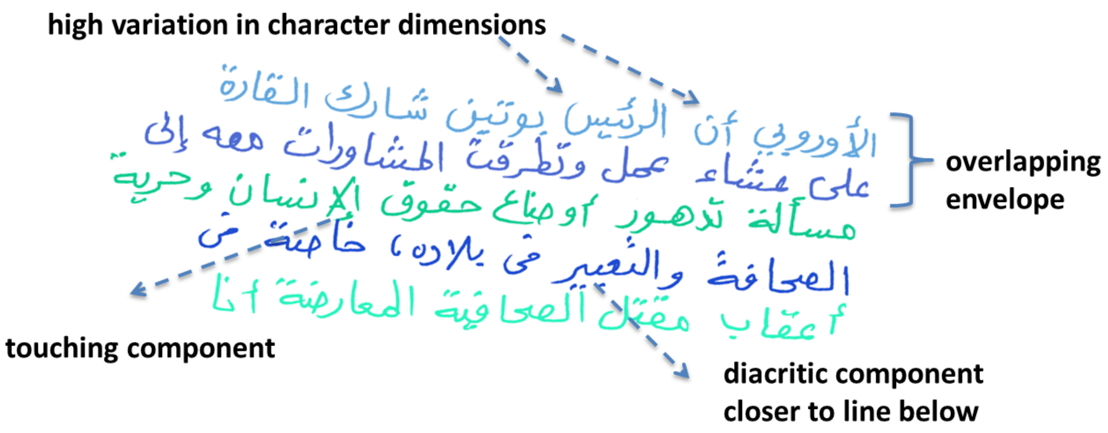

high variation in character dimensions
الأوروبي أن الرئيس بوتين شارك القارة
على هشاء عمل وتطرقت المشاورات معه إلى
مسألة تدهور أوضاع حقوق الإنسان وحرية
الصحافة والتعبير في بلاده، خاصة في
أعقاب مقتل الصحافية المعارضة أنا
overlapping envelope
touching component
diacritic component closer to line below

Figure 3.1: Challenges associated with handwritten text-line segmentation (sample taken from GALE data). High variation in character dimensions makes the segmentation more difficult. The accent components associated with a text-line may be visually closer to another line.

acters across and within textlines, this method is also inadequate. The presence of diacritical/accent components in some scripts make the problem even more complicated (Figure 3.1 shows for Arabic).

To address these problems, we adopt an approach which combines advantages of both local and global methods. We model the problem of text-line extraction as a clustering problem, and present a novel and fast way of obtaining text-lines in this Chapter. In Section 3.1, we present an overview of previous work related to handwritten text line extraction. We explain our method in detail in Section 3.2, and results of our experiments are presented in Section 3.3.

3.1 Related Work

Plamondon and Srihari [65] provide a brief survey of different text-line segmentation methods for on-line and off-line handwritten documents. Existing text-line extraction techniques can be broadly categorized as *projection based*, *component-grouping based* or *hybrid methods* [64, 66, 69]. *Projection based* methods typically divide the document image into vertical strips, compute horizontal projection profiles to extract components and group them based on few heuristics to extract text-lines. In [63], components are grouped by modeling the text-lines as bivariate Gaussian densities. Another method [66] initially over-segments the zones into text and gap regions, and uses a Hidden Markov Model to find the optimal assignment of text and gap areas in each zone. The width of the zone is selected to maximize the amount of text and minimize the effect of skew in each zone. Due to large variations in width of Arabic characters this criterion may not always be satisfied and the method may give suboptimal performance.

In [28], Pal *et al.* presents a text-line extraction method for handwritten Bangla document images. They use horizontal histograms of the vertical strips and the relationship of minimal values to obtain handwritten text-lines. The piecewise projection based line computation used in their method may not work well if the lines are closely spaced and the orientation variation within each line is high. *Connected-component based* methods merge neighboring connected components using rules based on the geometric relationship between neighboring blocks, such as distance and size compatibility. Alaei *et al.* [70] employed a painting technique to smear the foreground content of the document image. Their intuition is to enhance the separability between the foreground and background portions before applying text-line detection. They dilate the foreground portion of the painted image to obtain a single component for each text-line. The starting and ending points of the candidate line separators along with distances among them was used to obtain segmented text-lines. In [71], Louloudis *et al.* present a block-based Hough transform approach to detect handwritten text-lines. In a post-processing step false alarms are rectified using a merging method.

A very effective method based on curve evolution and level-sets was proposed in [64], but the method is slow for high resolution document images. A similar method based on the Mumford-Shah model was presented in [72]. Both algorithms are script independent but the main bottleneck is the computation time, which limits their application to large-scale document processing.

For handwritten Arabic documents, Zahour *et al.* used a partial contour following based method to find the separating lines[73]. They proposed a new segmentation method suited for Arabic historical manuscripts to segment the document images into three classes: text, graphics and background. But these approaches seem to be effective only when the components are not touching and the text-lines do not have overlapping envelopes. Nicolaou and Gatos [74] proposed a technique based on shredding the surface of text-lines with local minima tracers. In their approach, they make a topological assumption that for each text line, there exists a path from one side of the image to the other that traverses only one text-line. They first blur the document image and then follow the foreground and background paths from left to right as well as from right to left using a tracer in order to shred the image into text-line areas.

Historical documents, printed or handwritten, differ substantially from the modern-day documents discussed above since the layout formatting requirements were very loose. In many cases, it is more difficult to extract the physical structure of historical documents. Additionally, these documents are of low quality, due to aging or degradations. Characters and words may have unusual and varying shapes, depending on the writer, the period and the place of document creation. Likforman-Sulem *et al.* [75] surveys different approaches for text-line segmentation in historical documents.

Closer to our approach is the work of Yin and Liu [76]. They propose a text-line segmentation method based on minimal spanning tree (MST) clustering with distance metric learning. In their approach, the connected components (CCs) of a document image are first grouped into a tree structure. Text lines are then extracted by dynamically cutting the edges using their hypervolume reduction criterion and a straightness measure. By learning the distance metric in supervised setting on a dataset of pairs of CCs, their algorithm achieves robustness to handle multi-skewed and curved text-lines. Their method obtained a high accuracy on a dataset of Chinese documents.

In contrast to previous methods, our approach allows combining both soft-assignment and hard-assignment based estimates to obtain the final text-lines. We formulate the problem as graph-partitioning problem, and use both Affinity propagation and Breadth-first-search on locally computed estimates to obtain text-lines. Further, our framework allows line-level error correction and detection for obtaining more accurate estimates and is computationally faster than many previous approaches.

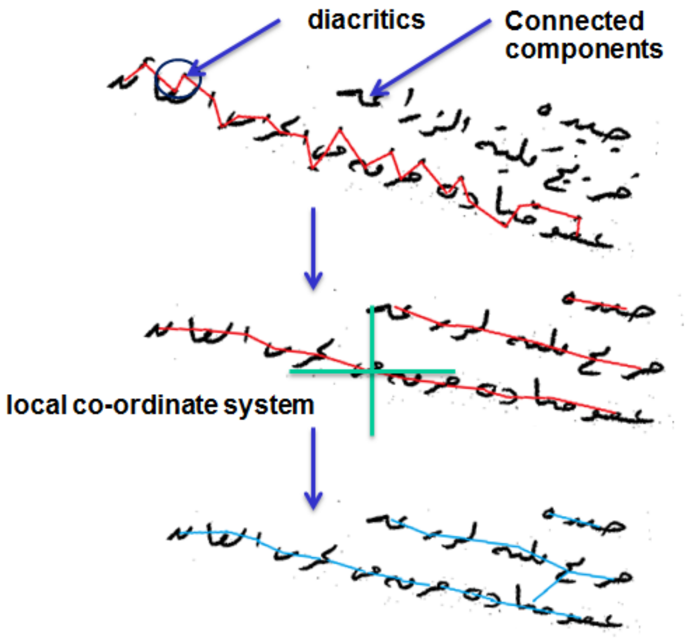


Figure 3.2: Flow diagram of our method showing diacritic/accent component removal, local orientation detection using coarse-components and orientation graph construction.

3.2 Approach

Our method consists of four steps: *Coarse text-line estimation*, *error detection and correction*, *touching component localization and separation*, and *diacritic/accent component assignment*. In the following subsections we explain each of these steps in detail.

3.2.1 Coarse Text Line Estimation

We first filter the probable accent and diacritic components based on the mass and size characteristics of components to obtain a set of coarse components (CCs) (Figure 3.2). Removing such small components gives us two advantages. First,

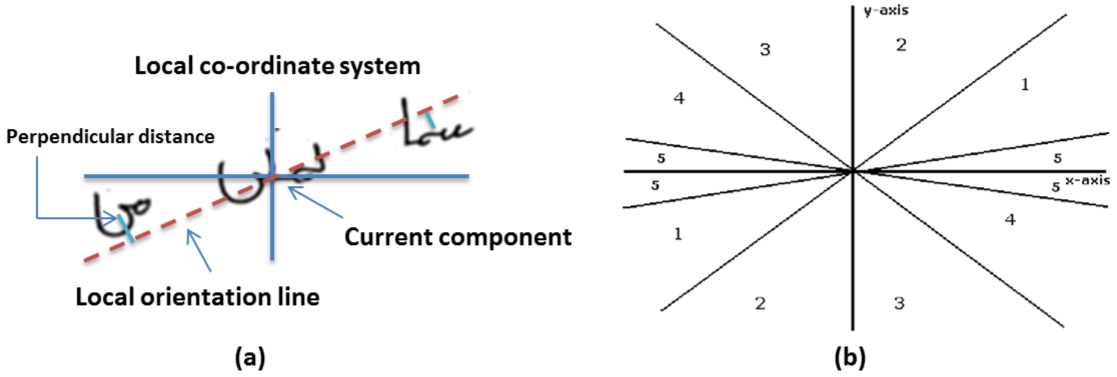


Figure 3.3: (a) A local cartesian co-ordinate system with the origin at the centroid of component. (b) Local orientation of text-line is quantized in to five directions

the reduction in number of components makes the graph-search and the Affinity-propagation method used in the next step fast and second, the coarse text-lines can be assumed to have smoothly varying orientation which can be estimated locally.

3.2.1.1 Local Orientation Detection

At each coarse component we estimate the direction of text-line by defining a local rectangular coordinate system with the origin at the centroid of the component (Figure 3.3). Let the set of all coarse components be denoted by S. For each coarse component $C_i \in S$, we define a local coordinate system centered at the centroid of the component and denote neighbors of C_i as $N(C_i)$ as given by Equation 3.1:

$$N(C_i) = \{C_j : j \neq i, D_{ij}(x) < R_x, D_{ij}(y) < R_y\} \tag{3.1}$$

where $D_{ij}(x)$ and $D_{ij}(y)$ represents the horizontal and vertical distances between C_j and C_i as given by Equation 3.2:

$$D_{ij}(x) = \|C_j(x) - C_i(x)\|, \quad D_{ij}(y) = \|C_j(y) - C_i(y)\| \tag{3.2}$$

where $C_i(x)$ and $C_i(y)$ denotes the x and y coordinates of centroid of C_i. R_x and R_y are adaptive and determined from the statistics of components in S and the size of current component C_i. Let H_{med} and W_{med} be the median height and width of bounding-boxes of all components in S and W_{cur} be the width of current component. The initial R_x and R_y is given by Equation 3.3:

$$R_x = W_{cur} + t_1 * W_{med}, \quad R_y = t_2 * H_{med} \tag{3.3}$$

where t_1 and t_2 are parameters whose value depends on the average character width and gap between the characters of a script. If the number of neighboring components is not sufficient for local orientation estimation, we increase the dimension of region in steps by a factor (f_1,f_2) as given by Equation 3.4:

$$R_x{}^{new} = f_1 * R_x{}^{old}, \quad R_y{}^{new} = f_2 * R_y{}^{old} \tag{3.4}$$

We divide each of the four quadrants to get eight regions around the origin as shown in Figure 3.3(b). Each pair of the diagonally adjacent region is grouped together to quantize the orientation of text line at C_i. Since the orientation can also be approximately horizontal, and the centroid of neighboring components may lie close to the x-axis in any of these regions, we define a new region which is the union of regions at an angle of 10 degrees with x-axis in each of the four quadrants. We consider all the neighboring components defined by Equation 3.1 which lie in the rectangular region centered at the centroid of current CC. We obtain the count of neighboring components in each of the five regions and find the region ${R_i}^{max}$ given by Equation 3.5 with maximum components.

$$ {R_i}^{max} = \max_j \{Count(R_j, C_i)\} \tag{3.5} $$

where $Count(R_j, C_i)$ denotes the count of components in region R_j. We estimate the direction by obtaining the least square estimate of a line passing through C_i using the centroid of components in region ${R_i}^{max}$. The orientation of line determines local orientation at C_i. We then find the distance between the centroid of each neighboring component in region ${R_i}^{max}$ and the estimated line to compute the similarity between the current component C_i and neighboring components as defined in Equation 3.6 and 3.7.

$$ Dist(C_j, C_i) = Dist(C_j, L_i) = \frac{|y_j - mx_j - b|}{\sqrt{m^2+1}} \tag{3.6} $$

$$ S(C_j, C_i) = \exp\left(-Dist(C_j, L_i)\right) \tag{3.7} $$

where m is the slope of line and b is the y-intercept of estimated line L_i at component C_i. In parallel, we build a similarity graph in which we put an edge between current component and other components in the region ${R_i}^{max}$ with similarity value given by Equation 3.7. When $Dist(C_i, C_j)$ is not equal to $Dist(C_j, C_i)$, we retain the

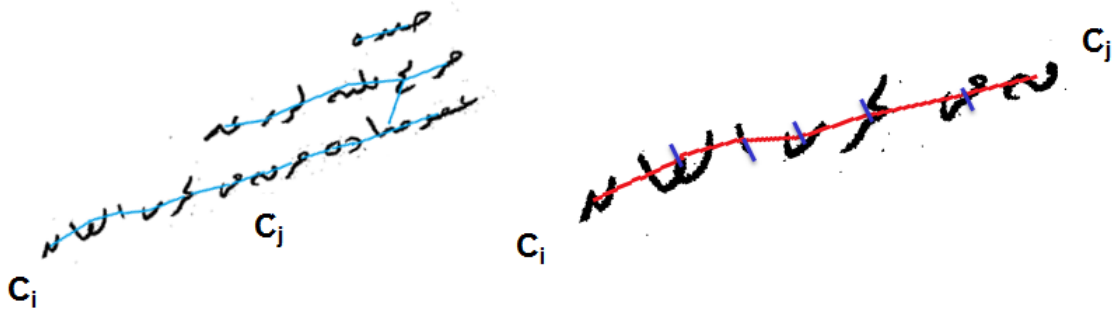


Figure 3.4: An illustration of shortest-path based similarity computation in our method. Similarity between two non-neighboring components C_i and C_j is computed based on the distance along the shortest path in local orientation graph. Short solid lines represent the distance to the orientation line estimated locally. These distances are summed along the shortest path for two non-neighboring components.

minimum of two so that the similarity matrix is symmetric. This will frequently happen when local orientation estimation is done at C_j and $C_i \in {R_j}^{max}$.

Once all the components are processed, a graph with nodes corresponding to the coarse components is obtained. The weights on edges between the current CC and the neighboring CCs are given by the distance to the estimated orientation line (Figure 3.3). A *shortest-path algorithm* [30] is used to compute the distances between non-neighboring components (Figure 3.4).

We obtain two estimates of text-lines based on this graph. First estimate uses Breadth-First-Search (BFS) [30] to find the different connected-components of the graph which represent potential text-lines if the local estimates are correct. But due to overlapping envelopes, touching components and character size variations, local estimates may not always be correct. Hence, we also compute soft similarities with all the other components and use Affinity-propagation clustering method [29] to obtain a global estimate of text-lines.

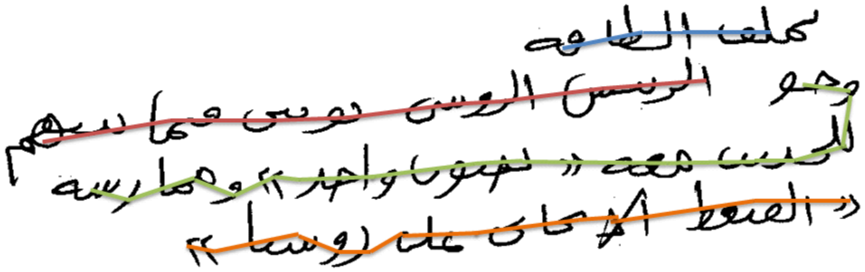


Figure 3.5: An illustration of scenario where BFS based approach could miss a coarse component resulting in a split error (second line from top).

3.2.1.2 Text-Line Estimation using Breadth-first Search

We find all disjoint subsets of vertices of the similarity graph for which there exist a path from each element to every other element in the set. These subsets represent *connected-components of the similarity graph* which are obtained using Breadth-First search (BFS) [30]. Ideally, if all the local estimations are correct then each connected component represents a text line. In practice, the variation in the size of characters and proximity between lines causes errors in the orientation estimation. For example, while computing the locally oriented neighbors of a component, if any component C_i of another line is also present in that region, then it will also have an edge to the current component as demonstrated in Figure 3.4. Another scenario where BFS based estimate may result in error is when a coarse component is missed in local estimation as shown in Figure 3.5. The running time of BFS is $O(|V|+|E|)$, where $|V|$ is the number of nodes and $|E|$ is the number of edges in the graph. Since the graph in our case is sparse, running time is almost linear ($O(|V|)$) in the number of coarse components.

3.2.1.3 Text-Line Estimation using Affinity Propagation

We first find similarities between all components in the graph as follows. For each node in the local orientation graph we find the shortest path to every other node using Dijkstra's shortest path algorithm [30]. The distance of two nodes is then based on sum of the distances in shortest path as given by Equation 3.8:

$$Dist_{SP}(C_i, C_j) = \sum_{C_l, C_k \in SP} Dist(C_l, C_k) \tag{3.8}$$

where SP is the set of nodes in shortest path from C_i to C_j. Once the distance is obtained, the similarity between node i and node j is computed by Equation 3.7. The running time of this algorithm is $O(|V|^2 + |E|)$, but for sparse graphs it can be implemented more efficiently in $O(|E| + |V|log|V|)$ using an adjacency list and Fibonacci heap as a data structure. We also assign a similarity to each component and its neighbors $N(C_i) \notin R_i^{max}$, by finding the distance to the estimated line at C_i as given by Equation 3.9, where distance $Dist(C_j, L_i)$ is given by Equation 3.6. $\alpha > 1$ is a penalizing factor for the component since it does not belong to ${R_i}^{max}$. Hence, in this way we have a non-zero probability for each component $C_j \in N(C_i)$ not detected locally, to become associated with the same text line as C_i on basis of AP.

$$S(C_j, C_i) = \exp\left(-\alpha * Dist(C_j, L_i)\right) \tag{3.9}$$

Affinity Propagation does not require an initial estimate of number of clusters.

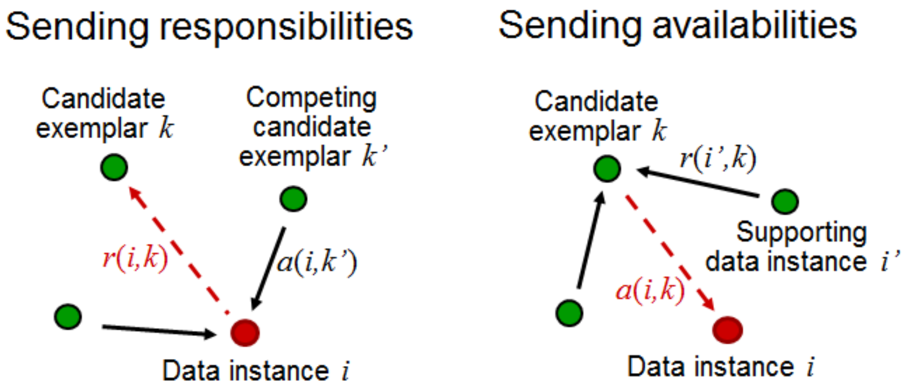


Figure 3.6: Two types of message exchanges in Affinity propagation.

It takes as input, a similarity matrix of real-valued entries, where the similarity *s(i,k)* indicates how well the data point with index k is suited to be the exemplar for data point i [29]. The diagonal entries of similarity matrix representing self-similarity is called the *preference*. This value for each data point k encodes the likelihood of that point to be selected as an exemplar. The number of clusters depends on both the values of the preferences, and the message-passing procedure. There are two kinds of messages exchanged between data points (Figure 3.6). The *responsibility r(i,k)* given by Equation 3.10, sent from data point i to point k, represents the evidence for how well-suited point k is to serve as the exemplar for point i. The *availability a(i,k)* given by Equation 3.11, sent from candidate exemplar point k to point i, shows the accumulated evidence for how appropriate it would be for point i to choose point k as its exemplar. The self-availability is updated in a different way as given by Equation 3.12.

$$r(i,k) \leftarrow s(i,k) - \max_{k',k' \neq k} \{a(i,k') + s(i,k')\} \tag{3.10}$$

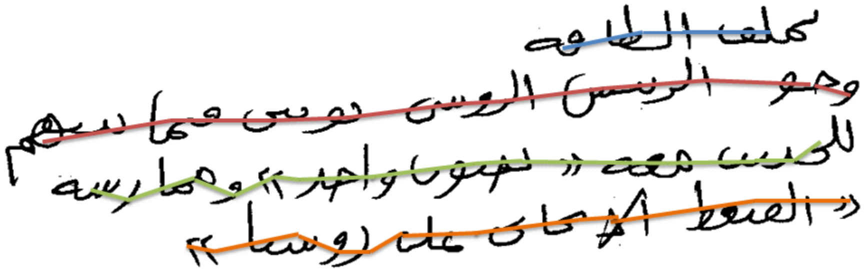


Figure 3.7: An illustration where Affinity propagation produced a different segmentation result as compared to BFS.

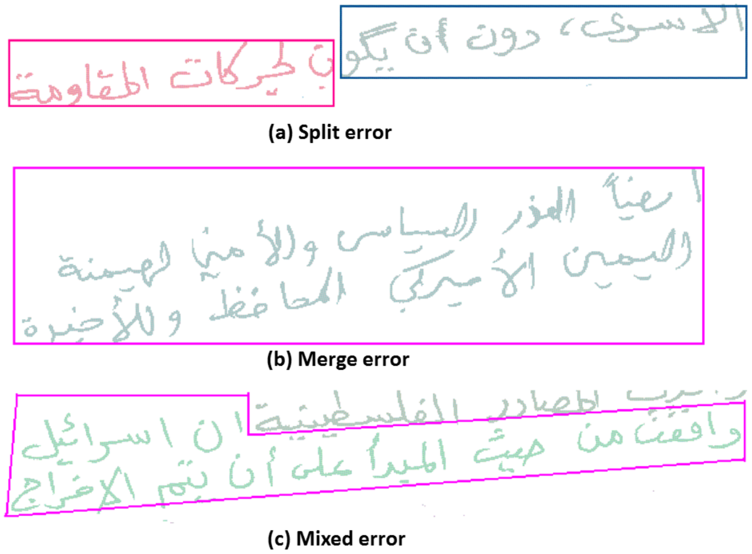
$$a(i,k) \leftarrow \min\{0, r(k,k) - \sum_{i', i' \notin \{k,i\}} \max\{0, r(i',k)\}\} \qquad (3.11)$$

$$a(k,k) \leftarrow \max\{0, \sum_{i', i' \neq k} r(i',k)\} \qquad (3.12)$$

Figure 3.7 shows an example from our GALE data where compared to BFS, Affinity propagation produced a different result using similarities computed based on shortest path in the graph. Finally, the two estimates are combined based on the definition of a valid text-line and a final set of text-lines are obtained.

3.2.2 Error Detection and Correction

Errors in text-line results obtained in previous step are of mainly three types: *split* errors, *merge* errors and *mixed* errors (Figure 3.8). When two or more neighboring text-lines in the result correspond to a single line in ground-truth then we refer to it as a *split error*. This usually happens when the local orientation detection fails to identify adjacent neighboring component. This error is detected and corrected by


الاسرى، دون أن يكون
تحركات المقاومة
أيضاً الدور السياسي والأمني لهيمنة
اليمين الأمريكي المحافظ والأخيرة
المصادر الفلسطينية أن اسرائيل
وافقت من حيث المبدأ على أن يتم الإفراج

(a) Split error

(b) Merge error

(c) Mixed error

Figure 3.8: Three types of error in text line segmentation results

checking against Affinity propagation based results and analyzing inter-component distances of each lines. Since the neighboring components will assume a smaller distance to the orientation line of a component, these adjacent components will have high similarities and hence are likely to get clustered together using AP. One good thing to notice about this error is that it preserves the reading order of text lines and hence for evaluations which consider reading order of text lines, correction of this error may not be required. Due to the same reason, it is easy to correct this kind of error. We correlate each text line in results obtained from AP to detect for disjoint multiple regions from BFS estimate, to detect such errors. The latter occurs if the best estimate of orientation involves some component of another text line, then both text lines may grouped together to form a single text line in result. Due to our adaptive region size computation at different scales for local orientation estimation this error is minimized to a good extent as demonstrated by our results on proximity data sets. In the next section, we explain an approach to correct errors due touching of components of two different lines.

3.2.2.1 Graph Based Merge Error Detection

Merge errors occur when multiple text-lines are grouped as one segment in our results. For each such segment obtained we find the *All-pair shortest path* distances [30] between the components. We then compare this graph-based Euclidian distance and the direct Euclidian distance between each pair of components to detect the merge errors. If the difference in the distances along the shortest path in the graph and the Euclidian distances of components is greater than some pre-defined threshold, then the detected line is declared an error segment. As shown in Figure 3.9, shortest path in the local orientation graph between components A and B is much greater than the direct Euclidian distance on the image. For a valid text-line the proposed scheme works because as we move from one end to another, direct Euclidian distance grows in proportion to the graph-based distance. For a segment with multiple text-lines, coarse components have much longer graph distances than Euclidian distances. We find the pair of components having the maximum difference in both the distances and use it to detect merge errors.

3.2.2.2 E-M Based Merge Error Correction

We iteratively apply E-M algorithm [77] to split the error detected segment into two segments in each iteration. This is done until the segments obtained in

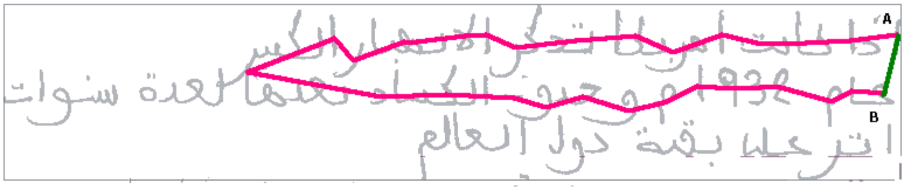


Figure 3.9: Two components labeled A and B have graph based Euclidian distance (red) along the shortest path much greater than the direct distance (green).

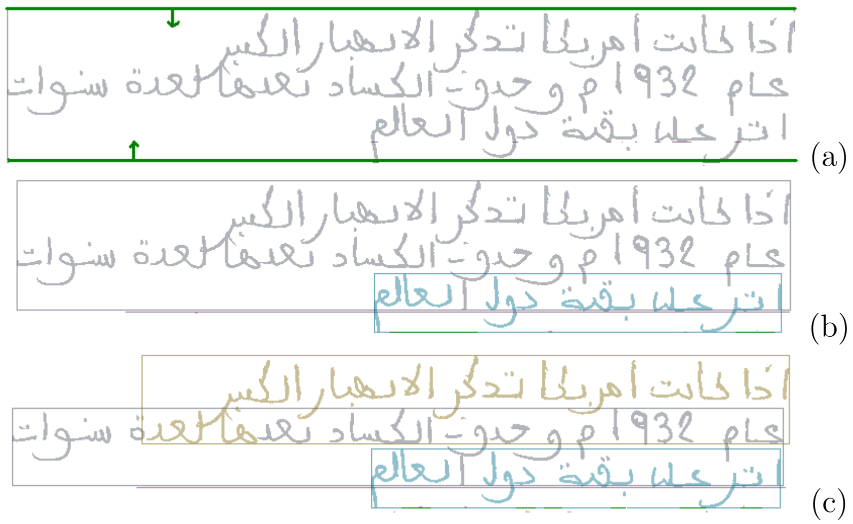


Figure 3.10: (a) Solid horizontal lines (in green) show the initial lines for EM (b) Two segments obtained using EM. Lower segment has no error but the upper segment has error (c) Results after applying EM on upper segment obtained in (b)

each iteration have no detected errors. We initialize two lines with slope m_k and y-intercept c_k based on majority of local orientations in each segment (as shown in Figure 3.10(a)) and update the parameters in each EM step. In the Expectation step, we compute the likelihood of each component being assigned to each line. For this, we find the residual (Res_{ik}) and the weights (w_{ik}) for each component as follows:

$$Res_{ik} = |m_k \cdot x_i + c_k - y_i| \tag{3.13}$$

$$w_{ik} = \frac{e^{-\frac{{Res_{ik}}^2}{\sigma^2}}}{\sum_k e^{-\frac{{Res_{ik}}^2}{\sigma^2}}} \tag{3.14}$$

where Res_{ik} represents the residual of the centroid of the i^{th} component with respect to the k^{th} line, $(x_i,\ y_i)$ represents the centroid of the component, and the free parameter σ corresponds to the amount of residual expected in the data.

In the Maximization step, we find the parameters that maximizes the likelihood of data points. We find the weighted least square estimate for each line. Equation 3.15 is solved twice (k = 1,2) one for each model using the weights obtained from the Expectation step.

$$\begin{pmatrix} \sum_i w_i {x_i}^2 & \sum_i w_i x_i \\ \sum_i w_i x_i & \sum_i w_i 1 \end{pmatrix} \begin{bmatrix} m \\ c \end{bmatrix} = \begin{bmatrix} \sum_i w_i x_i y_i \\ \sum_i w_i y_i \end{bmatrix} \tag{3.15}$$

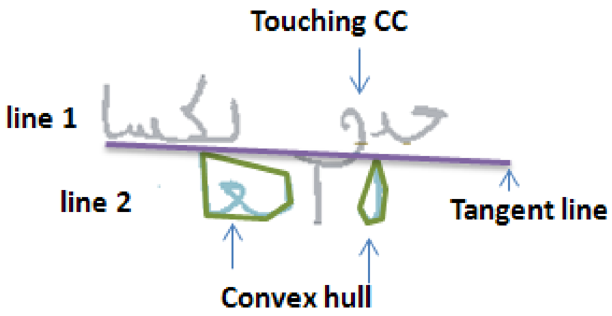


Figure 3.11: Common tangent to the Convex hulls of consecutive components is used for the error component localization.

3.2.3 Touching Component Localization and Separation

Once all the correct segments of an error detected segment are obtained, we localize the touching components in each segment. For this we find the common upper and lower tangent of the Convex-hull of consecutive components in neighboring lines (as demonstrated in Figure 3.11). The upper and lower tangent to Convex hull of the components give a good approximation of the extent of text-line locally in that region. If the ratio of the length of component below or above this tangent to the total height of the component is more than a certain threshold (determined empirically) then the component is considered to be a touching component. The accurate separation of such error detected component requires the knowledge of how character's shape changes when they touch and interact with each other during the writing. In this work we take a very simple approach and cut the component at the junction nearest to the centroid of component. Le *et al.* [78, 79] provides a more sophisticated approach based on contour based shape decomposition for accurate segmentation of touching characters.

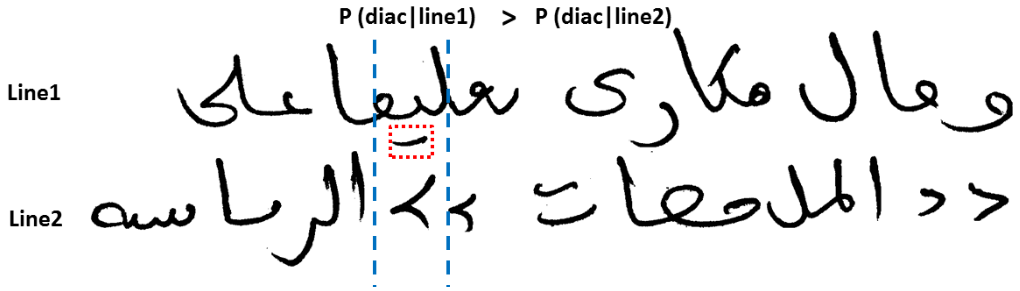


Figure 3.12: For each diacritic/accent component that was removed in first step, we compute the probability of associating it with candidate text-lines. The diacritic is assigned to the line which has highest probability.

3.2.4 Assignment of Diacritic and Accent Components

In this final step we assign all the components which were removed as probable diacritic and accent components to the text-lines. Each such component is given the label of best matched coarse text-line. We take a learning based approach in which we first learn different types of association using a set of training data. We hypothesize that the shape characteristics of a diacritic along with a rectangular region around its centroid provides enough context to find the correct association. We first find the candidate text-lines based on the distance to the centroid of component. We then compute the probability of assigning a particular component to each candidate text-line, and select the assignment with highest probability (Figure 3.12).

We learn an SVM model using two different types of features:

1. **Diacritic features:** We use geometric characteristics such as eccentricity, ratio of major-axis and minor-axis, component orientation, solidity to describe the diacritic/accent components. Table 3.1 lists different features used along

Table 3.1: Features used for modeling text-line and diacritic/accent component association.

Feature type	#features
Orientation histogram	36
Pixel bin histogram	100
Eccentricity (minimum bounding rectangle)	1
Aspect ratio (major-axis/minor-axis)	1
Component orientation	1
Solidity (extent to which convex/concave)	1
Relative position (above/below)	1

with their dimension.

2. **Context features:** We extract several features from the rectangular region around a diacritic component. We compute orientation at each foreground pixel in the region and quantize it into 36 bins to obtain histogram features. We compute pixel-density in the two-dimensional spatial grids obtained by dividing the region into m by n grids. Table 3.1 shows the listing of feature types and their numbers. We train a SVM classifier for obtaining the probability of removed component association with candidate text-lines.

3.3 Experiments

We conduct four sets of experiments to validate our graph-based text-line segmentation method. In the first experiment, the objective is to show the robustness of our approach when text-lines overlap and touch other lines. In the second, the objective is to demonstrate and compare results on more challenging Anfal dataset. In the third, the objective is to show the effectiveness of our approach on scripts

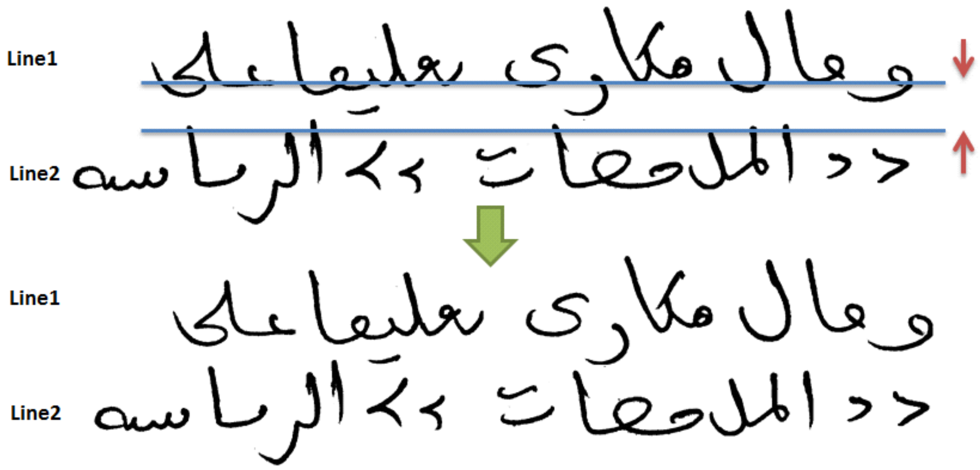


Figure 3.13: An illustration of construction of our relative proximity datasets. We computed average distance between two lines and moved each line closer to the line above it, in steps of some fixed fraction of average distance.

other than Arabic.

3.3.1 Competing Approaches

We compared our approach against the top eight methods in ICDAR segmentation competition [1]. On Anfal dataset, we compared our approach against the top performing method of Shi *et al.* [2] and with a projection profile approach of Arivazhagan *et al.* [63]. On another subset of Anfal dataset, we compared our approach with a learning based text-line segmentation approach of Le *et al.* [80].

3.3.2 Datasets

Our first dataset consists of a set of 123 Arabic document images with 1974 handwritten text-lines. We generated a set of proximity datasets using these images to test the robustness of our approach. We moved each line closer to the line above

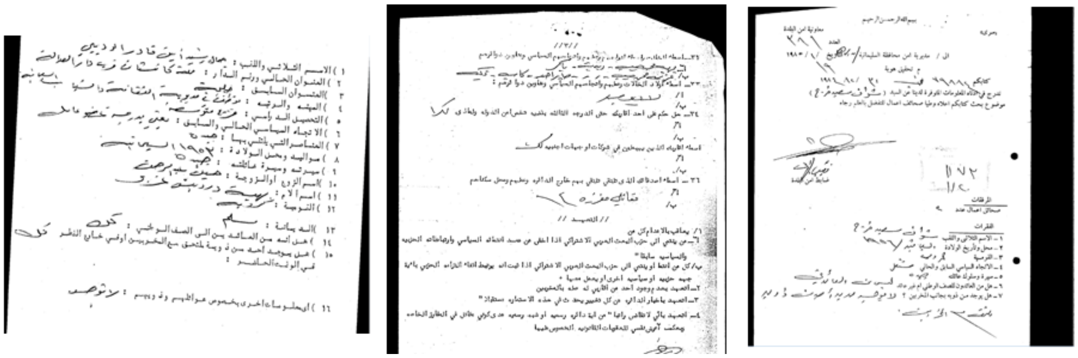


Figure 3.14: Sample document images from Arabic Anfal dataset (MADCAT project).

it, in steps of some fixed fraction of average distance between the lines, to generate a series of datasets (Figure 3.13). We call this the *Relative* proximity dataset [81] which has 19740 text-lines.

Our third and fourth datasets consist of 2677 and 487 images of Arabic documents obtained from a highly unconstrained and noisy field data. Figure 3.14 shows some sample images from Anfal field dataset. We used this dataset to compare our approach against the state of the art text-line segmentation methods [2, 80]. Our fifth dataset is ICDAR 2009 segmentation competition dataset (200 images). Results on this dataset will demonstrate that our approach adapts well to other scripts like French, Greek, English etc. Table 3.2 shows the different datasets used in our experiments.

3.3.3 Evaluation Protocol

We evaluated our results using a pixel-based matching-score(MS) criterion which is computed as follows:

Table 3.2: Datasets used in experiments. Second column is the number of images. The third column specifies the number of text-lines.

Dataset	*#images*(train/test)	*#textlines*
GALE (Arabic)	123	1974
Relative proximity (RP0-RP9)	1230	19740
Anfal1 (Arabic)	2477/200	3352 (test)
Anfal2 (Arabic)	250/237	13904
ICDAR 2009 segmentation competition	200	4034

$$MS(r_i, g_j) = \frac{T(P(r_i) \cap P(g_j))}{T(P(r_i) \cup P(g_j))} \tag{3.16}$$

where $MS(r_i, g_j)$ is a real number between 0 and 1 and represents the matching score between the result zone r_i and the ground truth zone g_j. P represents the foreground and T is an operator that counts the number of pixels in the zone. We obtain the matching-scores between all the result zones and the ground-truth zones. If the score is found above a pre-defined threshold then the result zone is counted as a True positive (TP). Result zones which are not matched to any ground truth zones are False positives (FP) and the ground-truth zones which are left unmatched are False-negatives (FN). We compute precision, recall and the F_1-score as follows:

$$Precision = \frac{TP}{TP + FP} \tag{3.17}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.18}$$

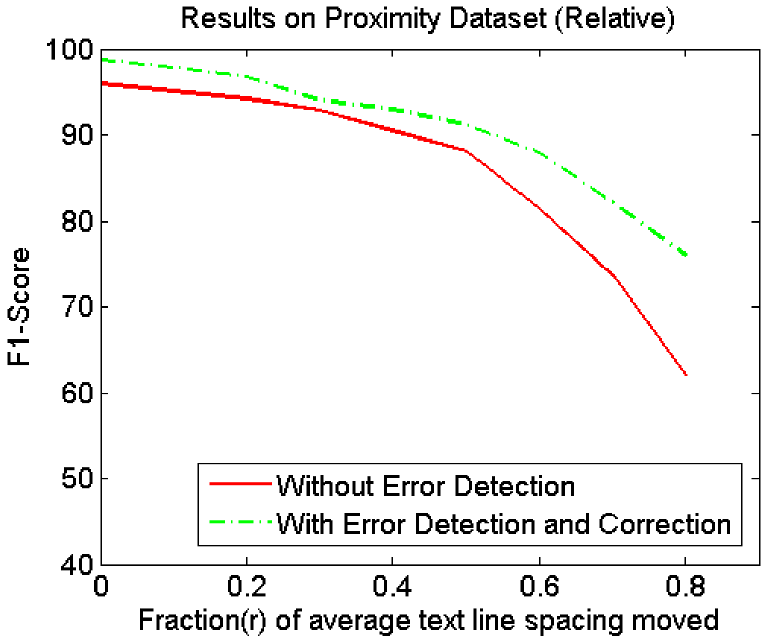


Figure 3.15: Plots of F-1 scores obtained using our method and a previous method which do not apply any touching error detection and correction method.

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3.19}$$

3.3.4 Results and Discussion

Figure 3.15 shows the F_1 scores obtained using our method on the relative-proximity data at MS threshold of 90. We obtained an F_1 score of 98.8% on the original GALE data (Fraction r = 0, when lines are not moved). To show the effectiveness of our error detection and correction method, we also plot the F_1 score of our approach without applying touching error detection and correction. As text-lines are moved closer, the performance of our method does not degrade as rapidly as the other method. We observe an improvement of 2.8% in the accuracy on the

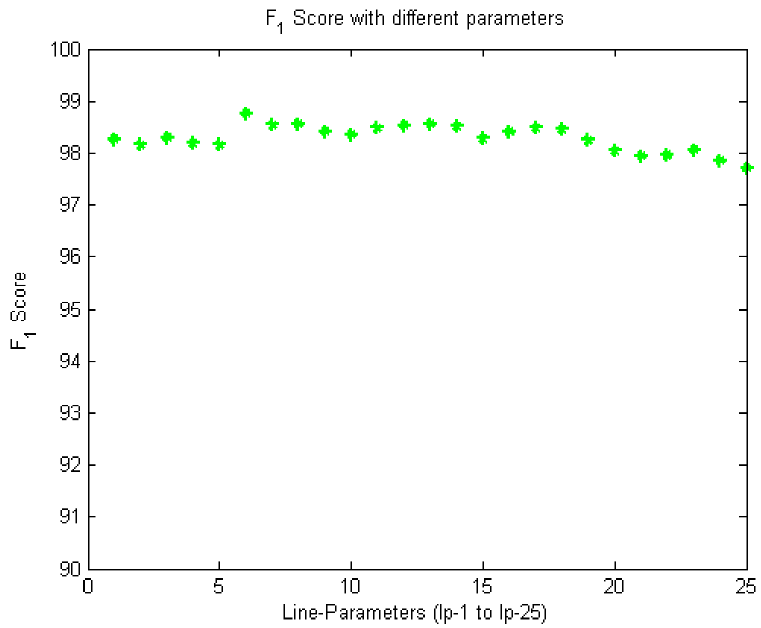


Figure 3.16: Plot of F-1 scores with different values of parameters.

original GALE data and 14% on the proximity data at r = 0.8. Total number of touching components in the the proximity data at r = 0.8 is 604. If the two lines overlap with each other completely then our method breaks down and the accuracy falls substantially. But in real documents we rarely see all the components of two text-lines touching.

Figure 3.16, shows the F_1 scores for different sets of parameters. We varied the parameters $HBand_{thres}$ and $VBand_{thres}$ defining the dimension of rectangular region for local orientation computation. Values from 5 to 3 in steps of 0.5 for $HBand_{thres}$ and 0.8 to 0.4 in steps of 0.1 for $VBand_{thres}$ are used to create 25 sets of parameters (lp-1 to lp-25). As shown, our method is robust to these two parameters.

Table 3.3 shows the accuracies of three methods on Anfal1 dataset. Our approach out-performed the steerable filter based method in [2] and projection profile

Table 3.3: Comparison of our method on a test set (200 images) from Anfal field data set (Anfal1). The steerable filter method is the top-performing method on ICDAR 2009 line segmentation dataset.

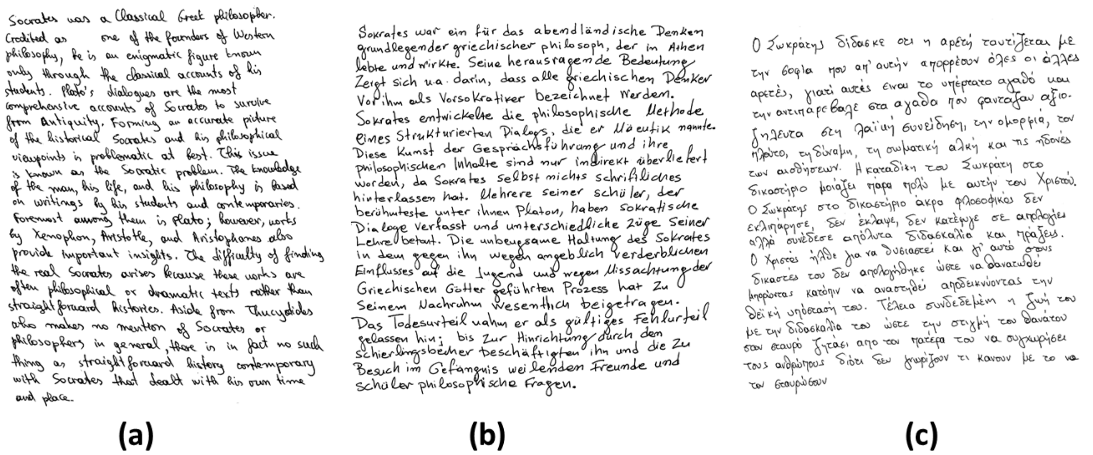
Method	Precision	Recall	F_1 score
Projection profile [63]	0.52	0.61	0.561
Steerable filter [2]	0.50	0.64	0.561
Graph based	**0.60**	0.53	**0.563**

Table 3.4: Comparison of our method on a test set (237 images) from Anfal field data set (Anfal2). Note that the other method is a learning-based approach and uses a training set as opposed to our approach which is unsupervised.

Method	F_1
Le *et al.* [80]	64.9
Graph-based	64.0

approach of [63] as reported in [82]. We also compared our method with a learning based approach for text-line segmentation in [80] on a test set of 200 images from Anfal2 data. The learning based method used 150 images for training and another set of 100 images for validation. Table 3.4 shows the accuracies of both methods with matching threshold of 0.75. Although our approach used only a small amount of development data (10 images) for tuning parameters, the accuracy is comparable to the learning based approach.

We also evaluated our method on ICDAR 2009 segmentation competition dataset (200 images) [1] and F-Measures(F_1) of top eight methods in the competition along with our method is given in the Table 3.5. Figure 3.17 shows some sample document images from ICDAR 2009 segmentation competition dataset. Although our method was developed for Arabic, it adapts well to other scripts like English,



Socrates was a Classical Greek philosopher.
Credited as one of the founders of Western
philosophy, he is an enigmatic figure known
only through the classical accounts of his
students. Plato's dialogues are the most
comprehensive accounts of Socrates to survive
from Antiquity. Forming an accurate picture
of the historical Socrates and his philosophical
viewpoints is problematic at best. This issue
is known as the Socratic problem. The knowledge
of the man, his life, and his philosophy is based
on writings by his students and contemporaries.
Foremost among them is Plato; however, works
by Xenophon, Aristotle, and Aristophanes also
provide important insights. The difficulty of finding
the real Socrates arises because these works are
often philosophical or dramatic texts rather than
straightforward histories. Aside from Thucydides
who makes no mention of Socrates or
philosophers in general, there is in fact no such
thing as straightforward history contemporary
with Socrates that dealt with his own time
and place.

Sokrates war ein für das abendländische Denken
grundlegender griechischer philosoph, der in Athen
lebte und wirkte. Seine herausragende Bedeutung
Zeigt sich u.a. darin, dass alle griechischen Denker
Vor ihm als Vorsokratiker bezeichnet Werden.
Sokrates entwickelte die philosophische Methode
eines Strukturierten Dialogs, die er Mäeutik nannte.
Diese Kunst der Gesprächsführung und ihre
philosophischen Inhalte sind nur indirekt überliefert
worden, da Sokrates selbst nichts schriftliches
hinterlassen hat. Mehrere seiner schüler, der
berühmteste unter ihnen Platon, haben sokratische
Dialoge verfasst und unterschiedliche Züge seiner
Lehre betont. Die unbeugsame Haltung des Sokrates
in dem gegen ihn wegen angeblich verderblichen
Einflusses auf die Jugend und wegen Missachtung der
Griechischen Götter geführten Prozess hat Zu
Seinem Nachruhm wesentlich beigetragen.
Das Todesurteil nahm er als gültiges Fehlurteil
gelassen hin; bis zur Hinrichtung durch den
Schierlingsbecher beschäftigten ihn und die zu
Besuch im Gefängnis weilenden Freunde und
schüler philosophische Fragen.

Ο Σωκράτης δίδασκε ότι η αρετή ταυτίζεται με
την σοφία που απ' αυτήν απορρέουν όλες οι άλλες
αρετές, γιατί αυτές είναι το υπέρτατο αγαθό και
την αντιπαρέβαλε στα αγαθα που φανταζαν αξιο-
ζηλευτα στη λαϊκή συνείδηση, την ομορφιά, τον
πλούτο, τη δύναμη, τη σωματική αλκή και τις ηδονές
των αισθήσεων. Η καταδίκη του Σωκράτη στο
δικαστήριο μοιάζει πάρα πολύ με αυτήν του Χριστού.
Ο Σωκράτης στο δικαστήριο άκρα φιλοσοφικός δεν
εκλιπάρησε, δεν έκλαψε, δεν κατέφυγε σε απολογίες
αλλά συνέδεσε απόλυτα διδασκαλία και πράξεις.
Ο Χριστός ήλθε για να θυσιαστεί και γι' αυτό στους
δικαστές του δεν απολογήθηκε ώστε να θανατωθεί
μπορώντας κατόπιν να αναστηθεί αποδεικνύοντας την
θεϊκή υπόστασή του. Τέλεια συνδεδεμένη η ζωή του
με την διδασκαλία του ώστε την στιγμή του θανάτου
στον σταυρό ζητάει απο τον πατέρα του να συγχωρήσει
τους ανθρώπους διότι δεν γνωρίζουν τι κανουν με το να
τον σταυρώσουν

Figure 3.17: Sample document images from ICDAR 2009 segmentation competition dataset which includes scripts like English, French, German and Greek.

Table 3.5: Comparison of our method with the top eight methods in ICDAR 2009 segmentation competition. Details of each participating method can be found at [1]. Our graph-based method is abbreviated as GB. The top-performing method (CUBS) is the work of Shi et al. [2].

Method	CUBS	ILSP	PAIS	CMM	CASIA	P-Univ	PPSL	LRDE	GB
F_1	99.5	99	98.5	98.4	95.6	94.5	93.4	92	97.8

French, German and Greek used in the dataset. The average time taken for the processing of a single image is 2.2 seconds for the proximity data and 3.2 seconds for the ICDAR competition data on a P4 machine with 3BG RAM.

For diacritic/accent component assignment we obtained five-fold cross validation accuracies for 10 sets in relative-proximity data (Figure 3.18). As lines get closer and start overlapping, the accuracy do not drop sharply and our features are effective (88.6% accuracy at RP9) when lines overlap significantly. Figure 3.19 shows examples of high-probability associations learned using our method from proximity-data RP5. Figure 3.20 shows an example scenario where our feature based approach

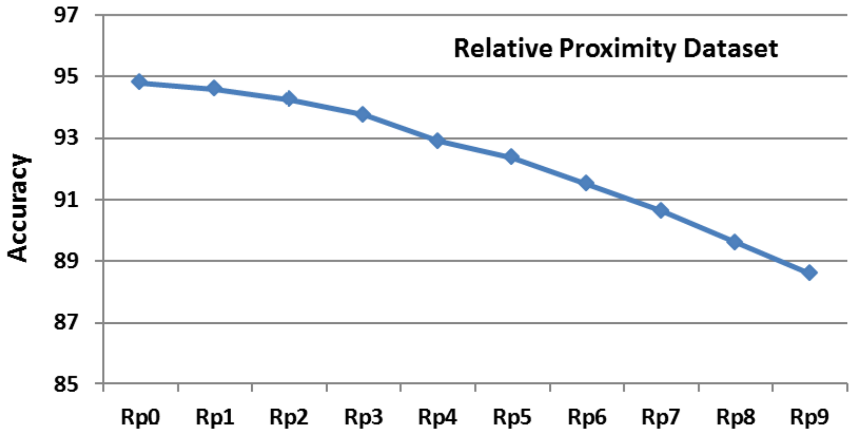


Figure 3.18: Plot of five-fold cross validation accuracy on relative proximity dataset (123 images in each set Rp0 to Rp9).

assigns accent components correctly to a text-line, even when the components are closer to other line distance-wise. In the distance based assignment, Euclidian distance of centroid to the average distance of text-line is used for assignment. In the shown scenario distance based approach will assign components to wrong line.

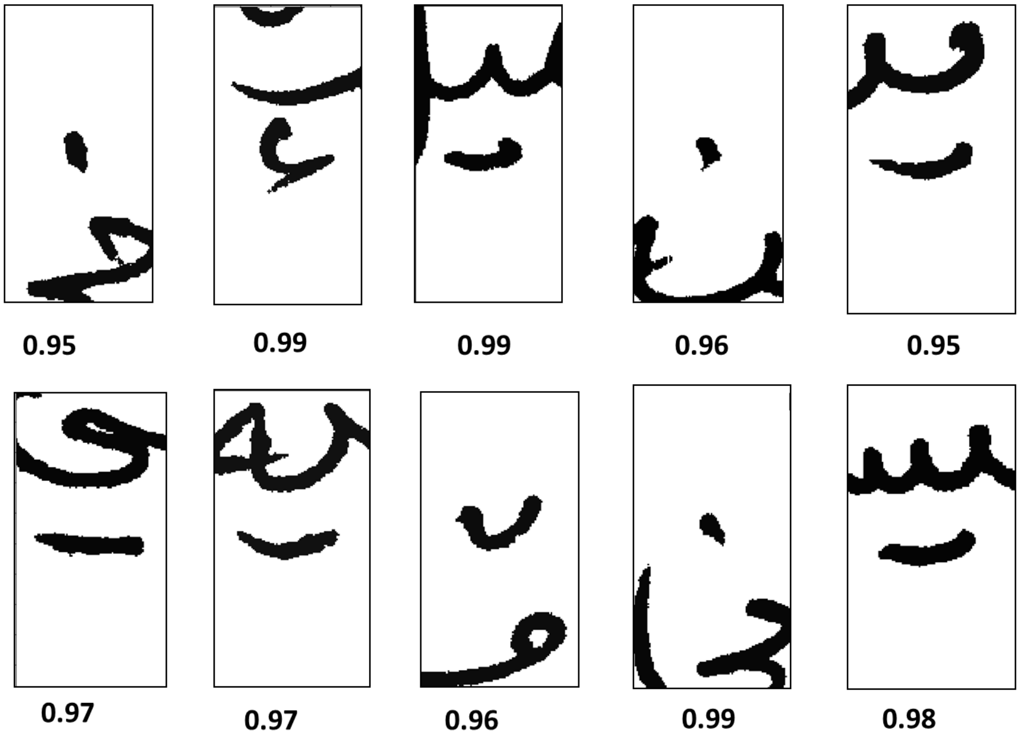


Figure 3.19: Some examples of high probability (> 0.95) diacritical/accent component associations learned in the proximity dataset Rp5.

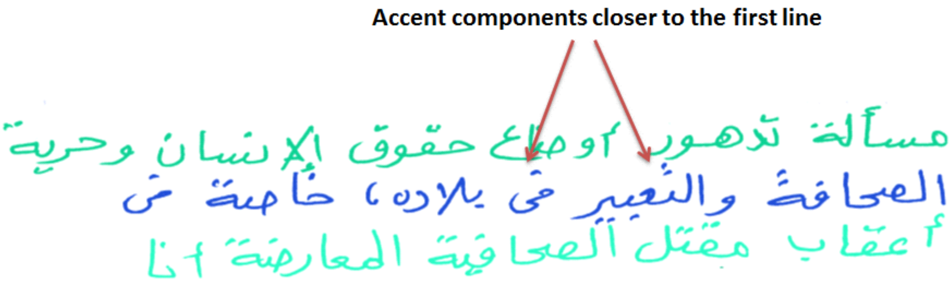


Figure 3.20: Example scenarios where our feature based approach assigns accent components correctly even when the components are closer to other text-lines. In the distance based assignment, Euclidian distance of centroid to the average distance of text-line is used for assignment. In the shown scenario distance based approach will assign components to wrong line.

Chapter 4: Structural Similarity for Retrieval and Classification

Finding structurally similar images in large heterogenous document image collections has been of interest for many years [32, 34, 37]. While there are numerous applications in office automation, litigation support and general document image search which depend on efficient and effective methods for computing similarity, previous approaches have focused on content-specific features or layout-specific structures [32, 33, 83]. Approaches based on content are highly dependent on, and sensitive to, the quality of optical character recognition (OCR), graphics recognition or component labeling. Since the OCR for unconstrained handwritten documents is still a difficult problem, content based approaches are typically limited to more structured machine printed documents [34]. Furthermore, layout based approaches tend to be tailored to fixed layouts, and model known classes of documents such as articles or forms. There is however an emerging need for effective methods for unconstrained document images for which OCR cannot be performed.

Recent work has focused on developing general methods capable of handling less constrained handwritten documents and datasets with highly variable layout [35, 84, 85]. Moreover, approaches which move beyond fixed partitions and compute similarity at different levels can adapt by allowing the user to specify the degree of similarity. For example, a range from 0.0 (no match) to 0.5 (conceptual match) to 1.0 (exact match).

Structural similarity therefore becomes important when users want to supplement search for images using visual content like logos, signatures, and tables etc., with search for *layout characteristics*. In such cases, users may or may not fully understand the layout or structural characteristics they are interested in, so they can either provide a sketch (or explanation) or provide some *representative* documents as examples. So they do not have to make these characteristics explicit, it becomes important to capture similarity at various levels, from the low-level content to high-level structure. Approaches developed for content-based matching and retrieval alone cannot be directly applied as they lack a high-level representation.

One effective way to define layout similarity for matching is based on structural features [32, 83, 86]. However, hand-crafting structure-based features (e.g., spatial relationships among the components) in unconstrained and noisy documents is difficult due to variation in content, translation, rotation and scale of components. Furthermore, as previously mentioned, a majority of the work published on defining and applying structural similarity is specific to a particular document type, such as business letters [31, 87]. The problem is made even more difficult when the number of relevant images for training is limited [39].

In this Chapter, we present a method for the classification and retrieval of structurally similar document images which can be applied to a broad class of documents. By *structural* similarity we mean primarily the layout and spatial organization of document content, including text, signatures, lines, logos, table-elements etc.

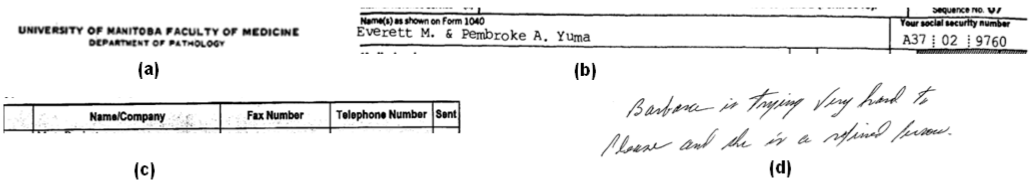

UNIVERSITY OF MANITOBA FACULTY OF MEDICINE
DEPARTMENT OF PATHOLOGY
(a)
Sequence No. 07
Name(s) as shown on Form 1040
Everett M. & Pembroke A. Yuma
Your social security number
A37 02 9760
(b)
Name/Company
Fax Number
Telephone Number
Sent
(c)
Barbara is trying very hard to
please and she is a refined person.
(d)

Figure 4.1: Document objects from Tobacco database showing horizontal bias

in documents. It is useful to map structural similarity to a scale from 0 to 1 where higher values indicates more precise match between document objects. Of course the similarity above which two documents are considered in same class depends on a specific application. A tax-form and a bank-form for example are structurally similar if we are interested in form retrieval, but are dissimilar if we are interested in retrieving a specific instance of a form.

Our approach is based on statistics of robust local features in different partitions of an image. The structure and layout of document objects such as text-lines, margins in text-blocks, lines in tables and border-designs typically run across both horizontal and vertical directions (Figure 4.1). To capture spatial relationships and correlations, we recursively divide the image horizontally and vertically, and compute histograms of *learned* codewords in these regions. We show that this strategy of modeling spatial relationships results in increased accuracy using the random forest (RF) classifier, even when only a few labeled samples are used for training.

We first explore an unsupervised feature learning method, using raw-image patches, to construct a codebook representation of basic structural elements in document images [85]. Since raw-image patches are not scale-invariant and are less robust to noise present in the monochromatic images, we find that it requires a large codebook to achieve a good performance. We then refine our approach by using SURF features as a basic unit of local content. SURF descriptors are more robust to noise and are scale-invariant. Second, we show that the approach is effective for *in-class* table and tax-form discrimination requiring very few labeled samples for training, and present classification results on 53 classes of hand-drawn table images and 20 classes of tax-form images. We compare our approach with the spatial-pyramid method [88] and show that our method gives superior performance on many document retrieval and classification tasks.

The remainder of this Chapter is organized as follows. In Section 4.1 we present related work on the retrieval of structurally similar document images. We discuss the details of our approach in Section 4.2 and present our experimental results in Section 4.3.

4.1 Related Work

There are a number of paradigms in which document image retrieval can be performed. For text-content based retrieval, scanned document images are typically converted to electronic (Unicode) text through optical character recognition (OCR) [89]. More recent retrieval approaches have focused on image-based representations allowing a focus on visual representation. When considering layout, the representation of documents using image-based features is often more intuitive and useful because it preserves the physical structure and access to non-text components such as embedded graphics [34].

An alternative approach defines *similarity* based on the *model* trained using features (possibly class specific) extracted from a user-provided set of *example* documents. Shin and Doermann [32] defined *visual similarity* of layout structures and applied supervised classification for each specific type. They used image features such as the percentage of text and non-text (graphics, images, tables, and rulings) in content regions, column structures, relative point sizes of fonts, density of content area, and statistics of features of connected components. They used a decision tree classifier and self-organizing maps for classification. The main drawback of their approach is that the features were designed for specific document classes (e.g., forms, letters, articles). Additionally, due to a large number of different feature types the approach is computationally slow for large scale document exploration.

Collins-Thompson and Nickolov [83] proposed a model for estimating the inter-page similarity in ordered collections of document images. They used features based on a combination of text and layout features, document structure, and topic concepts to discriminate between related and unrelated pages. Since the text from OCR may contain errors, especially for handwritten documents, the approach is limited to well-structured printed documents. Joutel et al. [86] presented an approach for the retrieval of handwritten historical documents at page level based on the *curvelet transform* to compose a unique signature for each page. The approach is effective when local shapes are important for classification but the approach is likely to miss any higher level of structural saliency. In many cases, the desired similarity is embedded in global structure and relationships among different objects in document images. In our approach, similarity is computed at two levels: first, a local match is performed using SURF based codewords and second, statistics of different codewords in different partitions are considered for higher level structure match.

Chen and Blostein [90] provides a detailed survey on document classification based on three components: the problem statement, the classifier architecture, and the performance evaluation. Kochi and Saitoh [91] proposed a system for identifying the type of a semi-formatted document based on important textual elements extraction and by using a flexible matching strategy for easy model generation. Bagdanov and Worring [92] approached the general problem of genre classification of printed document images using attributed relational graphs (ARGs). They used ARGs to represent the layout structure of document instances, and the first order random graphs (FORGs) to represent document genres. They reported a high-accuracy on a small dataset of 130 documents consisting of 10 genres. It is not clear whether their approach can handle high variation within same class and scale/rotation variation of same layout. Reddy *et al.* [93] address the form classification problem with a classifier based on the k-means algorithm. They use low-level pixel density features and adaptive boosting to classify NIST tax forms.

Approaches based on bag-of-words (BOW) models have shown promising results on many computer vision tasks such as image classification [94], scene understanding [95], and document image categorization [96, 84]. However, initial formulations for computing similarity typically disregard the spatial relationships between codewords, and only consider the occurrences of each codeword in an image. This results in a limited descriptive ability and performance degrades in presence of noise, background clutter, variation of layout and content in images. Subsequently, meth-

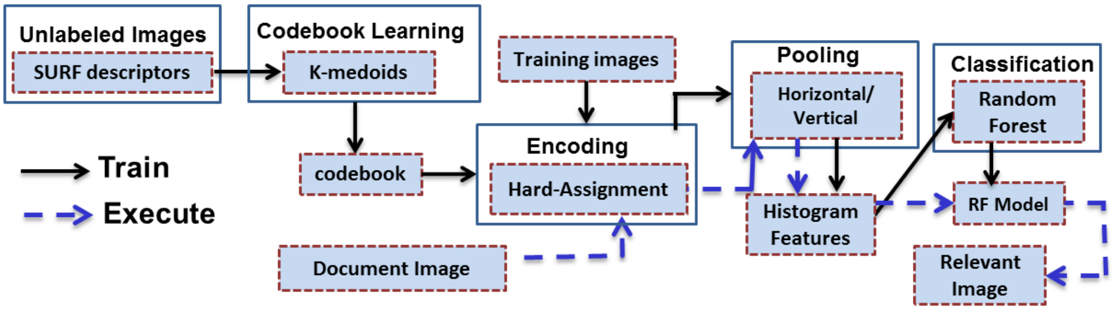


Figure 4.2: Block-diagram of our approach for structural-similarity based retrieval and classification of document images.

ods which extend the BOW approach to incorporate spatial relationships between visual codewords have been proposed. One of the early methods proposes the creation of *spatial-pyramid* features by partitioning the image into increasingly finer grids and computing the weighted histogram based kernel in each region [88]. Subsequently, there has been focus on selecting the optimal feature combination strategy and efficient ways to learn these local statistics, and a number of methods have been proposed [85, 97, 98]. In this work, we present a recursive horizontal-vertical partitioning scheme to learn spatial relationships in document images based on the observation that document objects have a horizontal and vertical bias (Figure 4.1).

4.2 Approach

Figure 5.1 shows a block-diagram of key components in our method. We describe each of these components in detail in the following sub-sections.

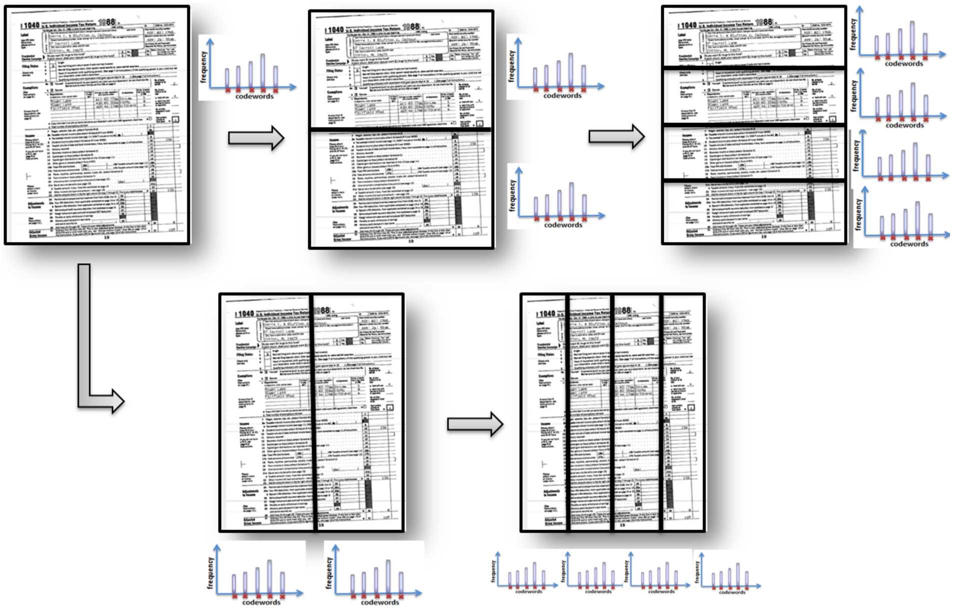


Figure 4.3: Recursive partitioning in vertical and horizontal directions to compute features for modeling spatial relationships in our approach. We compute a normalized histogram for each partition. Our feature set consist of all such histograms.

4.2.1 Codebook construction

In order to capture the local information of document components we use SURF descriptors [40] extracted from key-point locations in the image. Although raw image-patch based features are fast and effective [85], they are not invariant to scale or robust to noise. Compared to SIFT [99], the SURF descriptors are several times faster and more robust to noise, which often occurs during binarization [40]. We select a small set of representative document images for extracting 64 dimensional SURF descriptors. Using the K-medoids method, a set of exemplary codewords which represent the basic structural elements in the document database is obtained.

4.2.2 Horizontal-vertical pooling based features

In the next step we compute features for each document image. We find the nearest (L1-norm) codeword in the codebook for each SURF descriptor extracted from an image. We then compute a normalized histogram of codewords in each region obtained by recursively partitioning the image horizontally and vertically (Figure 4.3). The number of features (N) using our approach is:

$$N \quad = \quad \sum_{l=0}^{H} \sum_{k=1}^{2^l} |C| \quad + \quad \sum_{l=0}^{V} \sum_{k=1}^{2^l} |C| \tag{4.1}$$

where $|C|$ is the number of codewords, H and V represents the level of partition in the horizontal and vertical dimension respectively. If images have dimension (h, w), and $h \geq 1.2 \times w$, we perform an additional partition for h. A similar approach to capture spatial dependencies creates partitions using a *spatial-pyramid* (SP) scheme in which the image is recursively partitioned into four parts irrespective of its dimensions. In contrast, our approach partitions are based on the dimensions of image. Since the number of features per level in the SP method grows faster $(O(4^l))$ than our method, we have the same number of features even with one additional level of partitioning.

4.2.3 Random Forest Classifier

Using the recursive partitioning scheme to model spatial relationships, the number of features obtained in previous step is large (on the order of thousands).

We use a random forest (RF) [100] classifier for classification which has been shown to work well when many features are available. RF constructs a set of tree-based classifiers in the training phase, and then classifies new data points by taking a majority vote on the predictions of each classifier. We selected RF over other supervised learning methods such as SVM for several reasons. First, it increases diversity among the tree classifiers by re-sampling the training data, and by changing the feature sets over the different classifiers. This helps to avoid the *over-fitting* problem which often occurs with the increase in the number of features. Second, random selection of features to split each node make it more robust to noisy data. Third, the importance of a feature variable can be estimated by looking at how the classification accuracy changes when out-of-bag (OOB) data for that variable is permuted while all other variables are left unchanged [100]. Using *variable importance* plots, we find salient *partitions* for classification. We then skip *unimportant* partitions and retrain RF with features from *important* ones. This provides us computational efficiency, and in many cases, better performance.

4.3 Experiments

We conduct four sets of experiments to validate our horizontal-vertical partitioning/random forest based method (**HVP-RF**). In the first experiment, the objective is to show the effectiveness of retrieval with a limited number of labeled images. In the second, the objective is to demonstrate the ability to perform *in-class* discrimination on 53 classes of tables. In the third, the objective is to demonstrate

Table 4.1: Datasets used in our experiments. The second column specifies the number of classes in the dataset.

	#images	*categories*	*Usedfor*
Dataset1	5326 (824 Table + 1020 non-table Anfal + 3482 Tobacco)	53 classes of tables, Non-table	Retrieval, in-class table classification
Dataset2	9072 (5590 Tax-form + 3482 Tobacco)	20 classes of Tax form, Non-tax forms	Retrieval, in-class tax-form classification
Dataset3	2413 (FCC)	12 classes	Genre classification
Dataset4	3482 (Tobacco)	10 classes	Genre classification

effectiveness of our approach for Genre classification. In the fourth, we compare our spatial features with raw-image-patch based features [85] and show the advantage of SURF codewords.

4.3.1 Competing Approaches

As a basis of comparisons we have implemented comparable approaches based on *Bag-of-words* and *Spatial-pyramid.*

Global BOW model: We compute a global histogram of codewords for the whole image without partitioning. The accuracy against this baseline demonstrates the advantage of our partitioning scheme for computing structural similarity at different levels. We used with the global BOW features random forest for classification (**BOW-RF**).

Spatial-pyramid with Support vector machine : Spatial-pyramid matching was proposed by [88]. With spatial-pyramid features we compute accuracies for the random-forest classifier (**SP-RF**), SVM classifier with a linear kernel (**SP-LSVM**) and SVM classifier with a radial-basis kernel (**SP-RSVM**). For SVM, we used the implementation provided with LibSVM package ([60]).

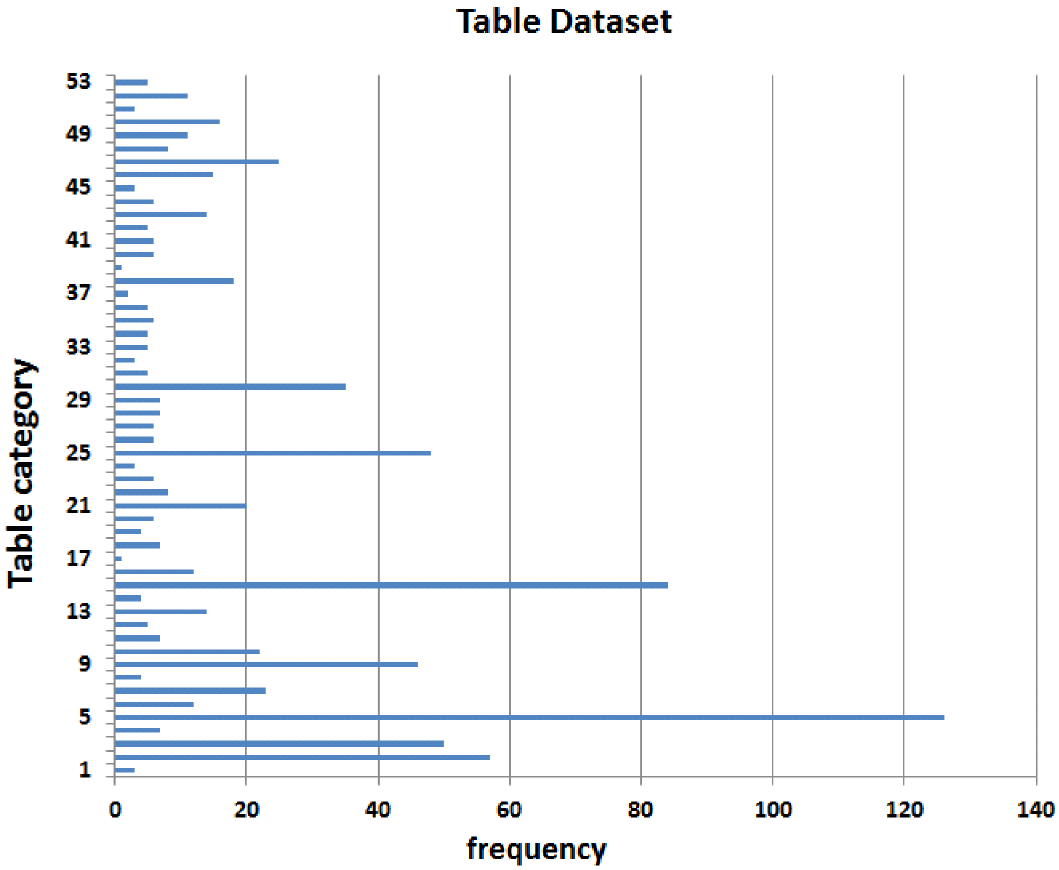


Figure 4.4: Distribution of images in different categories of Table dataset. A total of 824 images were manually labled into 53 classes.

4.3.2 Datasets

In our experiments, we use data from four collections: (1) a collection of Arabic document images collected in a field operation (Anfal)[84, 82] (Figure 4.4), (2) the Tobacco litigation dataset [101] (Figure 4.5), and (3) a collection of 5590

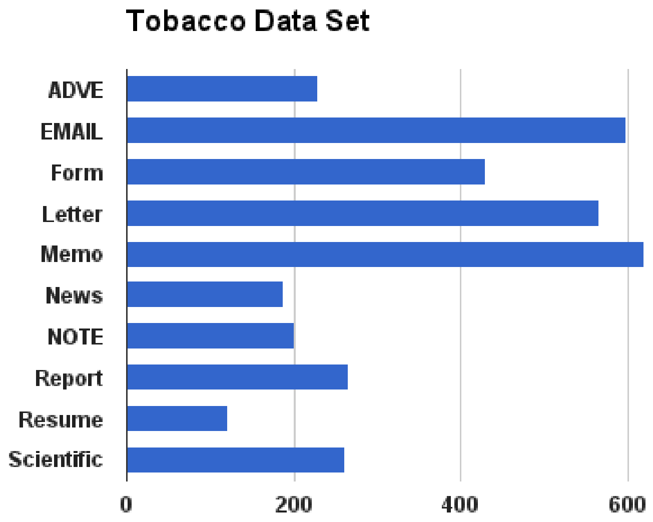


Figure 4.5: Number of images in each category of Tobacco dataset. A small subset of 3482 images was randomly sampled from a large collection and manually categorized into 10 classes.

tax-form images obtained from National Institute of Standards and Technology [102] (Figure 4.6) (4) a large database of documents stored at Federal Communications Commission (FCC) which includes various document related to filings that broadcast TV stations are required to submit to the FCC (Figure 4.7).

For the first set of experiments on table retrieval the dataset consists of 824 table images from the Anfal dataset (Figure 4.4), 1020 non-table images from the Anfal dataset and Tobacco dataset (3482 images). This is $Dataset1$ in Table 5.1. By adding images from different collections our goal is to test the performance of our retrieval approach when images with similar components but different layout are present. The table images were categorized into 53 classes manually, and used for *in-class* table discrimination.

The second dataset consists of 5590 tax-form images from NIST and 3482 non

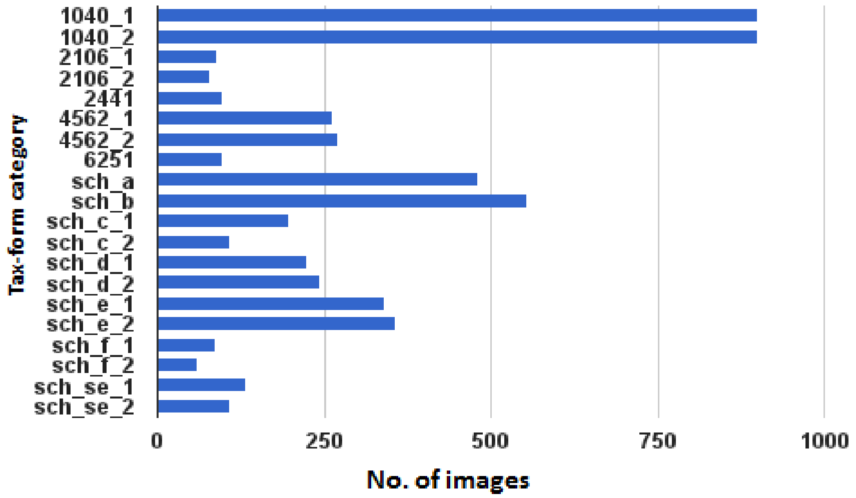


Figure 4.6: Number of images in each category of Tax dataset. A total of 5590 images in 20 classes of tax-forms constitute this dataset.

tax-form images from the Tobacco dataset consisting of 10 genres in total including *Memo*, *E-mail*, *Resume*, *Letter*, *Report*, *Forms*, *Advertisement*, *Scientific*, *Note*, *Letter*. This is *Dataset*2 in Table 5.1. Our motivation in constructing this dataset is to assess if our approach confuses tax-forms with different layouts and genres.

In our third dataset, we include 2413 documents from the FCC database consisting of four genres including orders, contract, invoice, agreement and a total of twelve sub-categories. Figure 4.7 shows the number of images in each of the twelve classes in the FCC dataset. Using this dataset we test whether our classification approach is robust to logical grouping of sub-categories into different genres.

Our fourth dataset consists of only the various genre of Tobacco images, and is used to compare different approaches for *genre classification*. A set of 3482 images from a large collection was randomly chosen and manually categorized into 10 genres. Images in categories such as *Advertisement*, *Resume*, *Report*, *News* exhibit high

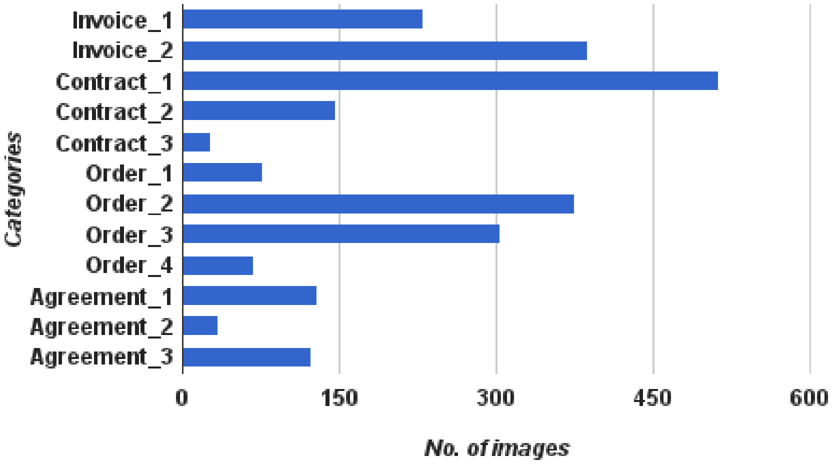


Figure 4.7: Different categories and sub-categories in our randomly sampled FCC dataset.

variation in structure, and images of different genre may have similar structure. Experiments on this dataset will assess whether our approach for classification is robust when there is no fixed layout in a particular category. Table 5.1 summarizes the four datasets constructed for our experiments.

4.3.3 Evaluation Protocol

For retrieval, we compute *precision*, *recall* and F_1 *harmonic mean* as metrics. If a *retrieved* image is relevant then it is counted as a *true positive* (TP), otherwise it is counted as a *false positive* (FP). If a relevant image is not retrieved it is counted as a *false negative* (FN). Using these counts we obtain the *precision*, *recall* and F_1-score using the following equations:

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4.4}$$

For evaluation of *in-class* table discrimination, we compute the accuracy of correct classes for each method.

Feature extraction and classification require various parameters. We used a codebook of size 300 and for HVP-RF, number of features at first, second and third level is 300, 1200 and 2400 respectively including both directions. An additional partitioning is done in the vertical direction which results in a total number of 2400 features. A three level partitioning was used for spatial-pyramid based methods to obtain a total of 6300 features. Since there is no partitioning involved in BOW-RF, the number of features is same as the size of codebook.

For the RF classifier there are two parameters, number of trees (nTree), and the number of attributes selected for each tree (mTry). In our experiments, we set mTry = $\sqrt{N}$, where N is the number of features and nTree = 500 [103]. In all of our experiments, we computed the *median* F1-scores and accuracies of 100 trials of experiments with randomly selected training and test images. Although there is a provision to incorporate class weights in the training of RF and SVM models, in

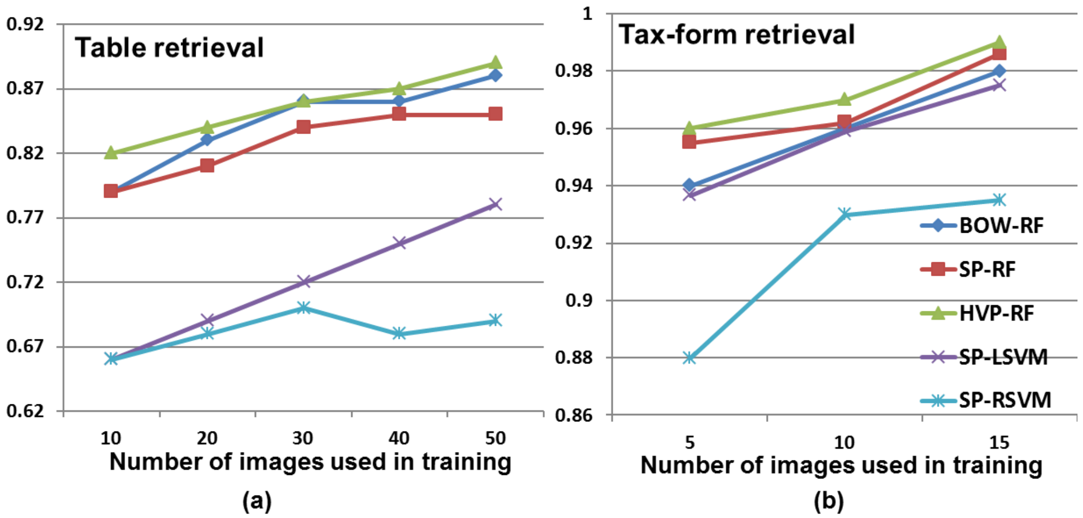


Figure 4.8: Retrieval accuracies (F1-scores) for (a) table images in *Dataset*1 (b) NIST tax-form images in *Dataset*2.

Table 4.2: Table image retrieval accuracies with increasing number of training images used.

#images for training	BOW-RF	SP-RF	HVP-RF	SP-LSVM	SP-RSVM
10	0.79	0.79	**0.82**	0.66	0.66
20	0.83	0.81	**0.84**	0.69	0.68
30	**0.86**	0.84	**0.86**	0.72	0.7
40	0.86	0.85	**0.87**	0.75	0.68
50	0.88	0.85	**0.89**	0.78	0.69

this work, we used uniform class priors to match a more realistic scenario where a user has no idea of the number of relevant images in the document collection. By varying the number of trees trained for each level of partitioning, feature weights can be introduced indirectly in RF. Unlike spatial-pyramid matching with SVM, we found that the improvement in accuracies for different weighting schemes based on partitions was not significant. In our experiments, we report accuracies for SP-LSVM and SP-RSVM using feature weighting scheme similar to [88], and for RF based approaches we used uniform weights.

Table 4.3: Tax-form image retrieval accuracies with increasing number of training images used.

#images for training	BOW-RF	SP-RF	HVP-RF	SP-LSVM	SP-RSVM
5	0.93	0.95	**0.96**	0.93	0.88
10	0.96	0.96	**0.97**	0.96	0.93
15	0.98	**0.99**	**0.99**	0.97	0.93

4.3.4 Results

4.3.4.1 Retrieval

We used *Dataset*1 and *Dataset*2 for retrieval experiments on table and tax-form images respectively. Figure 4.8(a) and Table 4.2 shows the median F_1-scores of 100 trials for table retrieval. Our approach (HVP-RF) achieves the best F_1 of 0.82 with only 10 table images for training. To compare the accuracy of SP-RF and HVP-RF, we performed paired-sample t-test between the two methods and at 0.05 level of significance, the results were significant (Null-hypothesis was rejected with p-value = 2.8020e-31).

SVM with a radial-basis kernel performs poorly compared to SVM with linear kernel due to a high-dimensional input space. We also observe that the global BOW model achieves a comparable accuracy when the number of labeled images for training increases, but the performance is lower with a limited number of training images.

In [88], features at coarser level were given less weights compared to features extracted from fine-regions. This feature weighting scheme was shown to be effective for computing similarity between images. For SP based approaches we observed that weights 0.25, 0.25 and 0.50 for the three levels outperformed uniform weights. However, for HVP-RF the median accuracy with uniform weights was slightly better than with weights. For example, with 50 images for training for table retrieval the median accuracy of HVP-RF was 0.89 while the accuracy was 0.87 with feature weights of 0.2, 0.2, 0.3 and 0.3. We hypothesize that for document images, matching at a coarser level is not necessarily less important than match at finer level. In fact, there may be cases when coarser match of a structure (lines, table borders) is more important than finer match (text content in cells).

Figure 4.8(b) and Table 4.3 shows the median F_1-scores for the retrieval of tax-forms from *Dataset*2. The 3482 Tobacco documents in *Dataset*2 contain about 400 form images that may be confused with the NIST tax forms, making the task more challenging. Our method achieves a median F_1-score of 0.99 with only 15 images used in training. Similar to previous case, SVM with radial basis kernel performs poorly and the SVM with linear kernel performs comparable to BOW-RF method. Overall, HVP-RF and SP-RF performs best among all methods showing the advantage of feature pooling over local regions. We also observed that the performance of RF is more consistent than SVM in the 100 trials of experiments. Additionally, we observed a consistently higher *test-error* compared to the *train-error*, and a higher percentage of *support-vectors* in the training data ($> 75\%$) in different iterations of our experiments supporting our hypothesis that SVM might overfit in high-dimensional feature space.

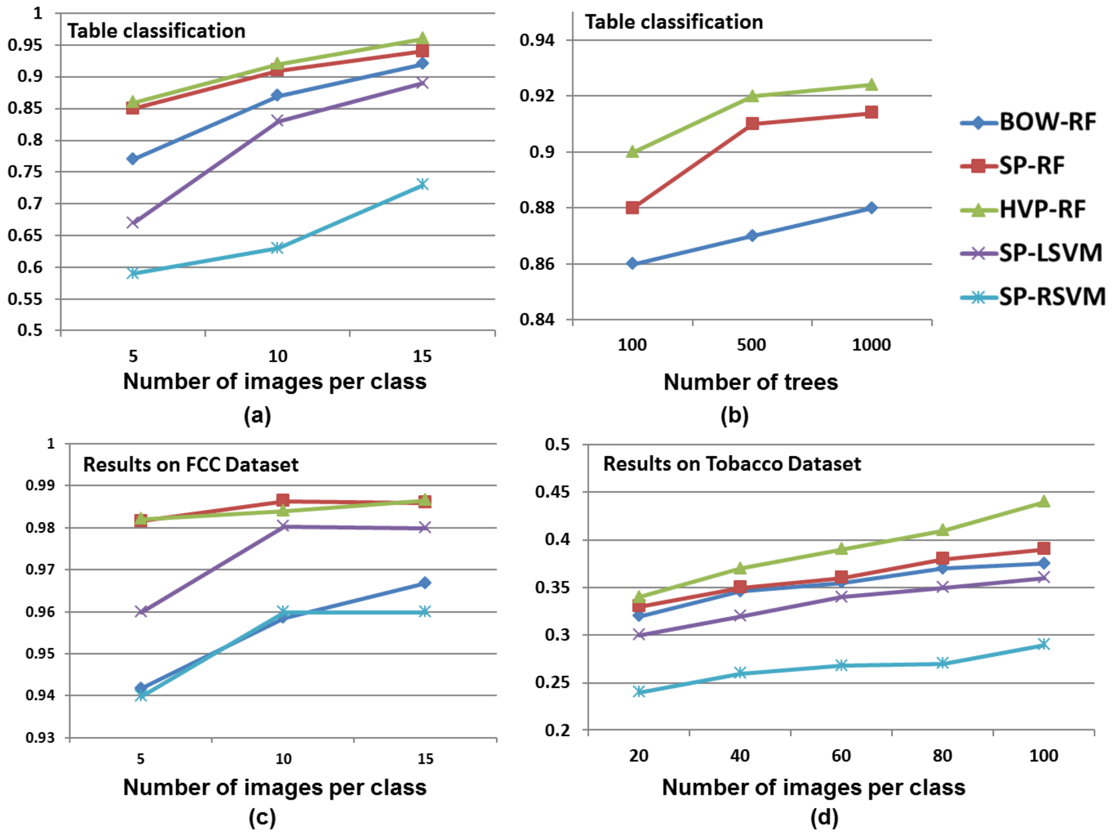
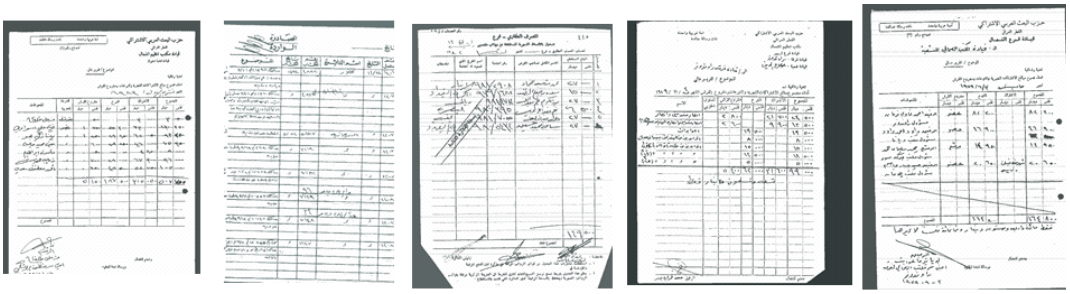


Figure 4.9: (a) Classification accuracies for five methods on table images (824 images, 53 classes). (b) Classification accuracies with different number of trees in random forest. Number of training images used is 10. (c) Classification results on FCC dataset (2413 images, 12 classes) (d) Genre classification results for five methods on Tobacco dataset (3482 images, 10 classes). Number of trees used in RF training for all cases is 500.



Different Categories in Table Dataset

Figure 4.10: Sample images in our Table dataset.

4.3.4.2 Classification

We used the 824 table images in *Dataset*1 to demonstrate the ability of our approach to perform *in-class* table discrimination. Figure 4.9(a) shows the median classification accuracies of 100 iterations. The dataset has a total of 53 classes of tables. Figure 4.10 shows some sample table structures in our Table dataset. The abscissa in the plot shows the number of images used per class for training the classifier. Note that the multi-class SVM results are obtained using K binary SVM classifiers, where K is the number of classes. In contrast, RF is inherently a multi-class classifier so a single training session is required. Both SVM based methods fail to achieve as high an accuracy on the test sets as our approach based on RF. One of the first things to observe is that with only five images per class for training, our approach achieved a median accuracy of 0.86. When the number of training images is increased to 10, HVP-RF and SP-RF achieved an accuracy of 0.92 and 0.91 respectively. We performed a paired-sample t-test to see if the difference in accuracies was significant. At 0.05 level of significance, the results were significant with p-value = 0.0461. It should be noted that the combination of spatial-pyramid features and random-forest (SP-RF) had not been proposed earlier.

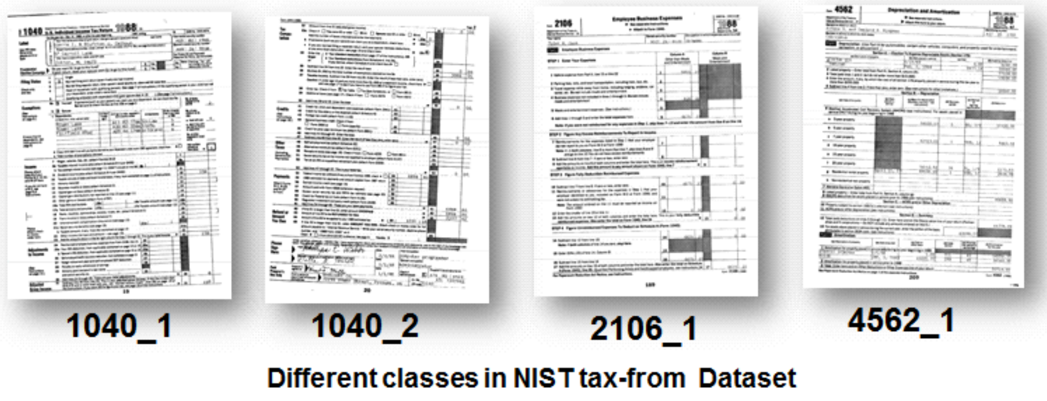


Figure 4.11: Sample images in our NIST tax-form dataset.

For 20 classes of tax-form images in *Dataset*2, all methods except BOW-RF and SP-RSVM achieved an accuracy of 1.0 with just a *single* image for training (20 images in total). Figure 4.11 shows sample images from tax-form dataset. Previous work have reported similar high accuracies on this dataset but they used a much larger training set in their experiments ([32, 36]). Overall, we find that the random-forest based approaches have more robustness and consistency than SVM for *in-class* table and tax-form classification.

Figure 4.9(b) shows the classification accuracies with different numbers of trees used in RF training. The performance of RF based approaches do not degrade significantly when the number of trees is reduced to 100. The plot shows that RF provides a good trade-off when computational resources are limited, and in order to obtain a high classification accuracy a large number of trees is not necessarily needed.

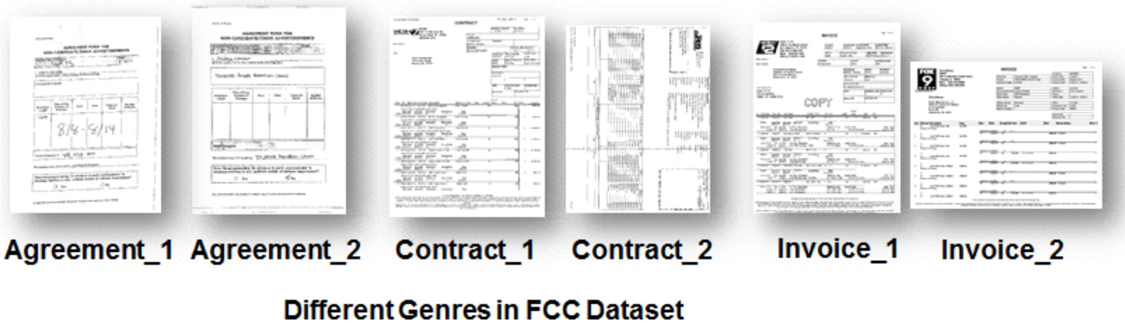


Figure 4.12: Sample images in our FCC dataset.

Figure 4.9(c) shows the median classification accuracies for *Dataset*3 consisting of 12 classes of documents in the FCC dataset. All classifiers were trained for 12 classes taking into account the *sub-types* of contracts, invoices, order and agreement. Figure 4.12 shows sample images from FCC dataset. We further analyzed the classification accuracies of different methods when *sub-types* are combined to form a single coarse category (one of contract, invoice, order and agreement). Table 4.4 shows the median accuracies for each of the five methods. As observed, HVP-RF and SP-RF performs competitively and are best among all methods. BOW-RF method which relies only on global histogram and do not perform any spatial pooling, is not able to achieve as high accuracy as SP-RF and HVP-RF. We also observe slight poor performance of SP-LSVM as compared to our approach.

Figure 4.9(d) shows the median classification accuracies for *Dataset*4 consisting of 10 genres from Tobacco dataset. Compared to tables and tax-forms the structures in different genres exhibit high variation, and hence none of the approaches achieve a high accuracy on this dataset. Figure 4.13 shows sample images of the *Advertisement*, *News*, and *Resume* genre from our dataset. As observed two adver-

Table 4.4: Classification accuracies for five methods on FCC dataset after combining classes into major types (four classes: contracts, invoices, order, agreement).

#images per sub-class	BOW-RF	SP-RF	HVP-RF	SP-LSVM	SP-RSVM
2	0.837	**0.967**	0.930	0.950	0.338
3	0.892	**0.980**	0.960	0.970	0.340
4	0.917	**0.982**	0.970	0.970	0.340
5	0.928	**0.989**	0.986	0.980	0.344

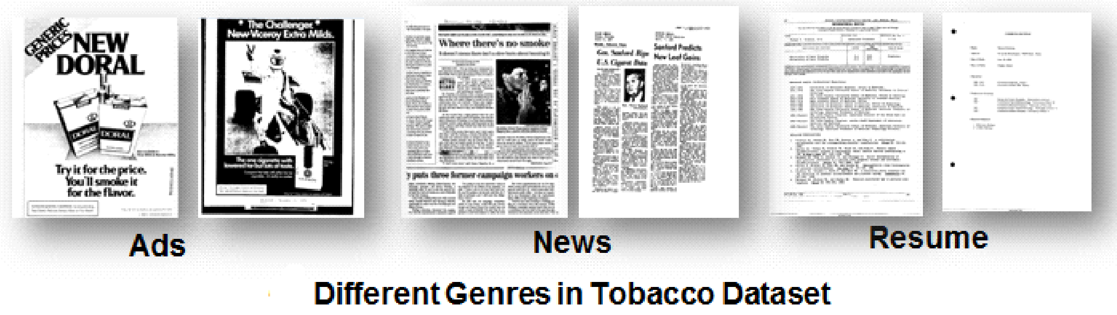


Figure 4.13: Sample images from Tobacco dataset.

tisements look very different structurally, and both ours and SP-based approaches fail to capture this high variation. Our method outperformed other approaches and achieved a median accuracy 43.8% when 100 training images were used per class. Table 4.5 shows the class-confusion matrix for one of the iterations. Due to similarity in structure confusion is high between *Report* and *Note*, *Memo* and *News*, *Resume* and *Form*, *Email* and *Note*, *Email* and *Form*.

We computed *importance plots* using RF for different pairs of classes to visualize and learn important partitions for classification. We computed importance of each feature using two criteria: (a) GINI index (b) Decrease in overall accuracy. For each partition we computed overall importance by summing the importance of features belonging to that region. Figure 4.14 shows the heat-map (higher value im-

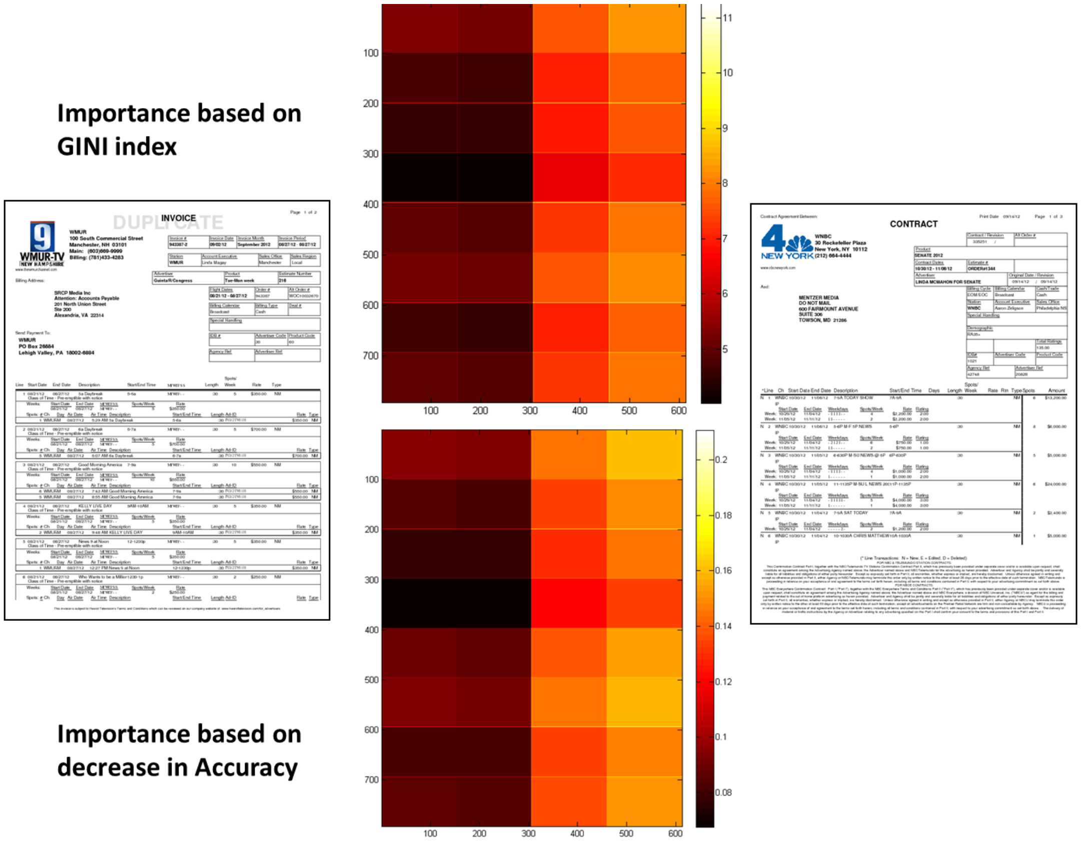


Figure 4.14: Heat-maps using importance plots obtained from random forest training for classes (1) $Invoice_3$ (2) $Contract_1$. Top plot was obtained using GINI index while bottom plot was obtained based on decrease in accuracy.

Table 4.5: HVP-RF class-confusion matrix for genre classification. The number of images used per class for training is 100. A total of 2482 images was used for testing. Overall accuracy was 43.27%.

	Report	Memo	Resume	Scientific	Letter	News	Note	Ad	Form	Email
Report	**102**	22	2	9	30	6	18	0	21	14
Memo	2	**137**	0	3	5	12	3	4	6	4
Resume	2	11	**8**	35	11	1	0	11	16	27
Scientific	7	43	4	**59**	15	12	3	9	22	32
Letter	1	30	0	2	**297**	0	0	2	56	1
News	10	114	0	13	4	**40**	24	5	8	35
Note	20	58	2	8	4	15	**46**	5	31	38
Ad	8	28	2	6	13	2	2	**74**	20	92
Form	2	35	1	14	79	8	2	11	**107**	52
Email	11	42	1	12	9	2	3	9	44	**204**
Accuracy (%)	61.8	26.3	40.0	36.6	63.6	45.5	45.5	56.9	32.3	40.9

plies more importance) for invoice class versus contract class in our experiment. As seen in heat-map, regions in third and fourth vertical column (shown in yellow) are more important than others for classification. Figure 4.15 shows a similar heat-map for the two sub-types of same *Order class*. The map shows that middle regions are more discriminative than others for these two classes.

4.3.4.3 Computation Time

We also compared the training and execution times of four methods on a 2.4 GHz Windows machine with 14 GB RAM (Table 4.6). We used 100 images for training from each class in *Dataset*4, and report the median times (in seconds) of 10 iterations for each method. We observe that RF has a slower training time compared

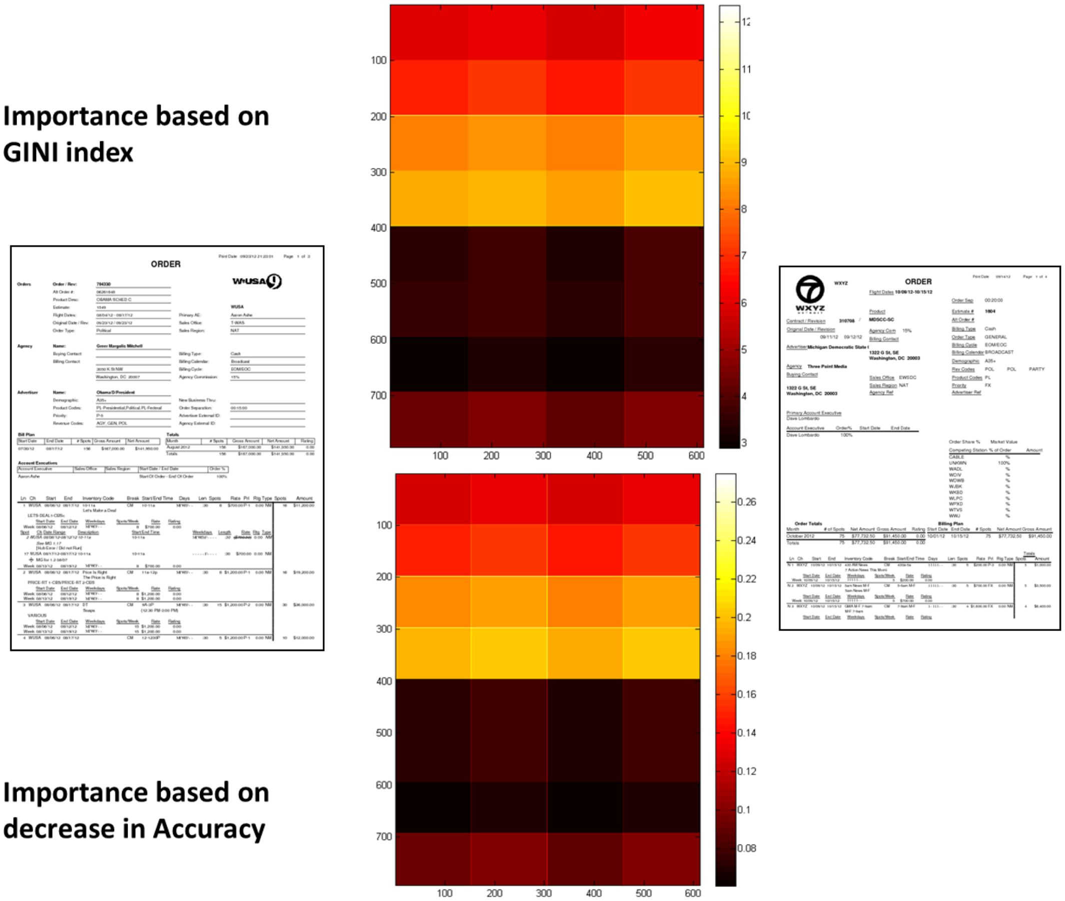


Figure 4.15: Heat-maps using importance plots obtained from random forest training for classes (1) $Order_2$ (2) $Order_3$. Top plot was obtained using GINI index while bottom plot was obtained based on decrease in accuracy.

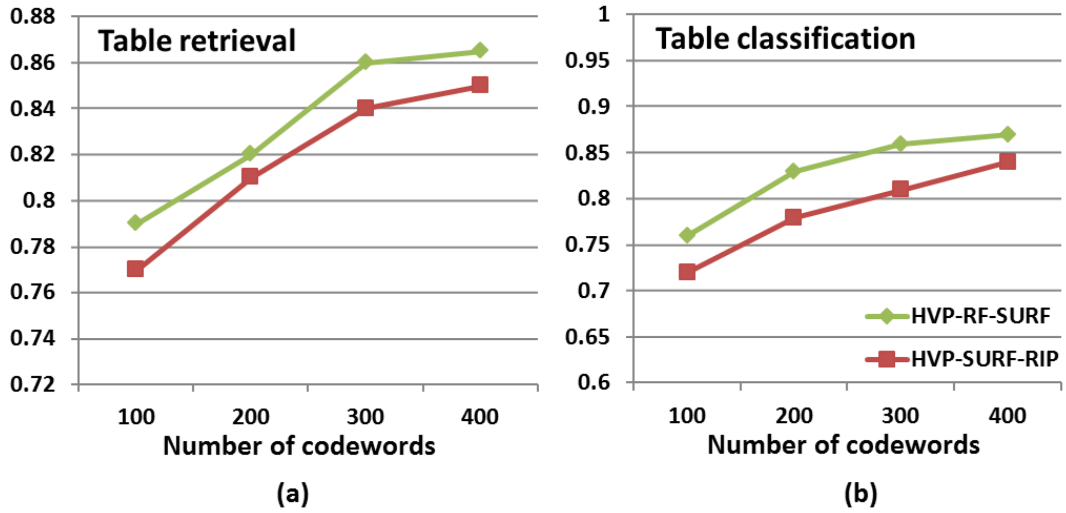


Figure 4.16: (a) Table retrieval accuracies (median F_1 score) with SURF and Raw-image-patch (RIP) based features on *Dataset*1. Number of table documents used in training is 30. (b) Table classification accuracies with SURF and Raw-image-patch (RIP) based features. Number of relevant documents per class used in training is 5.

to SVM but significantly faster execution (test) times. RF training is slower since many trees (order of hundreds) are learned for the ensemble based decision. We used 100 and 500 trees in RF training for reporting the times. A faster execution time makes RF more suitable for large-scale document classification.

Table 4.6: Train time and test time (in seconds) for Tobacco dataset. Number of training images used is 1000 and the number of test images is 2482.

	SP-LSVM	SP-RSVM	HVP-RF (100 trees)	HVP-RF (500 trees)
Train	**8.85**	11.24	107.85	520.04
Test	42.23	49.16	**0.19**	**0.55**

4.3.4.4 SURF vs. Raw-Image-Patch features

Finally, we compared the performance of our approach using SURF and raw-image-patch (RIP) based features and present results in Figure 4.16. RIP based features were presented in [85], and were shown to be very effective for document classification. In our experiments, we find that the performance of RIP based features is competitive only when the number of codewords is large. In contrast, SURF based spatial features achieve better accuracies with fewer codewords. This result is in agreement with a previous work on image quality estimation [104] where a large codebook (order of thousands) was necessary with RIP features. While it is efficient to extract RIP features from images, the increase in the number of codewords results in increased computation time by classifiers. Overall, we find that SURF descriptors are both efficient and provide a more compact codebook than RIP features.

Chapter 5: Structural Similarity for Unsupervised Classification

A majority of the work in defining and applying structural similarity often requires a supervised setting in which labeled examples of each class are required to learn the model. For example, tree based approaches have been popular for business letters and forms [31, 87, 105]. In Chapter 4, we discussed how with the help of few user provided examples we can categorize large document databases for search and browsing. Grouping images with similar structural characteristics in a completely unsupervised way (without user provided examples) is also useful when nothing or very little is known about the image collections at hand. Unsupervised approaches not only saves manual labeling cost and time, but also gives “first hand” insight into the data. In the last part of our thesis, our focus is to develop an approach which automatically estimates the number of classes in a collection, and groups them without any supervised information.

In this Chapter, we present an approach for grouping structurally similar document images for unsupervised exploration of large document collections [106]. We believe there is a significant amount of information in the layout and structure of documents that can be used to group them by *type* as done by humans. While previous approaches have focused primarily on *distance* based similarity, our approach uses a random forest (RF) classifier [100] to first *learn structurally salient patterns* in the document collection. The similarity is then computed based on levels of content and structure matching using the trained decision trees in the RF. Our approach provides an effective framework for grouping images with different levels of similarity - from an approximate match to only a high-level match.

We use the same bag-of-visual words based on SURF descriptors [40] to characterize content type, and spatial pooling strategies that we developed for supervised setting. We avoid supervision in the subsequent stage by constructing a randomly sampled *auxiliary data*, and use it to learn important structural patterns and correlations in our original data [106]. In order to estimate the number of classes, we use *silhouette-coefficient* [107] for different groups obtained after clustering. We experimented with four different datasets of document images varying in size, number and type of classes of document images. Our grouping results using Normalized-cuts clustering method [108] outperforms existing approaches based on spatial-pyramid matching and Euclidian distance [88].

5.1 Related Work

Although there has been limited work on structural similarity based grouping of document images, there has been enormous work in defining similarity and clustering of document and scene images. In this section we briefly review and compare the existing approaches for such similarity computation of images. Most of the previous methods are applied in supervised setting for retrieval and classification. We discuss advantages of our approach and why existing approaches are not directly applicable to define structural similarity for document images.

5.1.1 Similarity Measures

When documents are represented by feature vectors, the most common way to compare them is the Euclidean-distance [109, 110] or the L1 distance [111]. In template matching based approaches the similarity among images is computed by a simple value comparison [112] or using a specific dissimilarity function [113]. A modified version of the cosine similarity is proposed in [110]. In [114], the similarity is computed by means of the Chi-square distance using the distribution of tf-idf vectors, while Bhattacharyya distance between two histograms is used in [115]. Fataicha *et al.* proposed yet another measure based on minimum edit-distance [116]. When the number of features is large, distance based approaches suffer from curse-of-dimensionality and similarities computed are not accurate. In contrast, we propose Random Forest based similarity computation which works well with high dimensional data since only a subset of features are used to train trees.

5.1.2 Unsupervised Classification of Document Images

Joutel *et al.* [86] proposed an approach for page-level retrieval of handwritten historical documents based on the curvelet transform. The approach is effective when local shapes are important for classification but their approach is likely to miss any higher level of structural saliency. In many cases, the similarity is hidden in global structure and relationships among different objects. Chen *et al.* [36] presented an approach for structure based document classification by matching the salient feature points between the query image and the reference images. Their work uses a set of document images for training, whereas our work is completely unsupervised. Saund [117] presented an unsupervised approach based on the sub-graph features extracted from the line joints, and reported a high-accuracy on clustering NIST tax-forms. However, the work is limited to documents with “line-art”.

Our approach differs from previous approaches in several ways: (1) We present an unsupervised document image categorization approach which is applicable to a broad class of documents (2) Similarity in our approach uses different levels of content and structure match. At the lowest level, bag-of-words model based on SURF is used for content description, and for higher levels we partition image recursively and compute statistics of codewords (3) Unlike many previous approaches we do not have a fixed notion of distance-based similarity for document images. We use random forest for learning spatial relationships in the first step, and then use the trained trees for computing similarities of document images.

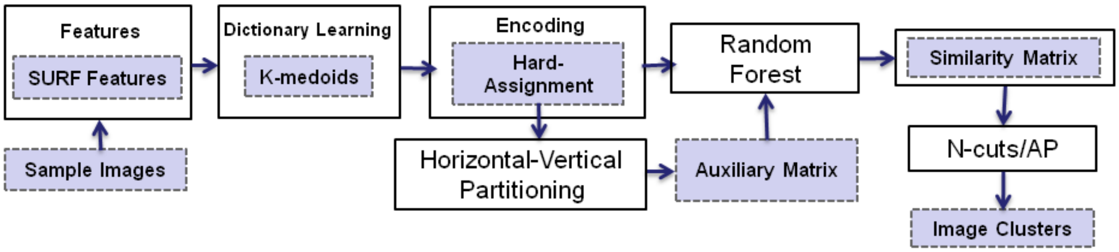


Figure 5.1: Block-diagram of our approach for grouping document images. N-cut and AP refers to Normalized-cut and Affinity propagation based clustering method respectively. If the number of classes is not known, we use Silhouette-coefficient to estimate the number of groups prior to clustering.

5.2 Approach

A block-diagram of our method is shown in Figure 5.1. Key components in this framework are: (1) SURF feature extraction and Codebook construction, (2) Hard-assignment feature encoding, (3) Horizontal and vertical partitioning for spatial feature pooling, (4) Random forest based similarity computation and (5) Unsupervised clustering. First three steps are similar to our supervised approach explained in Chapter 4. We describe in detail our auxiliary data construction and random-forest based similarity computation in the following sub-sections.

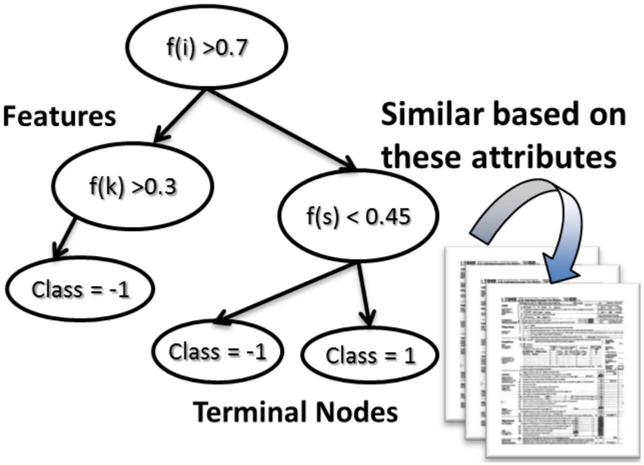


Figure 5.2: Illustration of similarity computation using random forest (RF). After RF training, documents that land in the same terminal node are *similar* based on the features used in that tree. Similarity is incremented for each such case. Finally, the similarities are symmetrized and divided by the number of trees.

5.2.1 Auxiliary Data Construction

We used the random forest (RF) classifier [100] for computing pair-wise similarities. The RF is an ensemble-based learning algorithm which constructs a set of

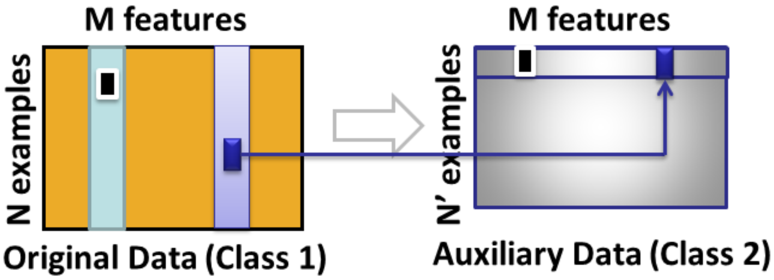


Figure 5.3: Auxiliary data construction for training a two-class random forest classifier. The auxiliary matrix A with the same dimensions as original data is created by randomly sampling values from the feature distributions in the original data. The correlations in original data due to a specific pattern will not be captured in the Auxiliary data.

tree-based classifiers during training. Pair-wise similarity can be computed in RF by counting the number of occurrences of two documents being assigned to same terminal nodes in the trained trees (Figure 5.2).

As such, RF is a multi-class *supervised* classifier and needs labeled data of at least two classes for training. The idea is to train a two-class RF classifier such that the *correlations* and *dependencies* in features are discovered during the tree construction, and specific patterns in the original data are *learned* during training. To do this we first create auxiliary data of same dimension which serves as a second class for training a two-class RF classifier. The tree classifiers of the random forest aim to separate auxiliary from observed data. Hence, each tree uses *splitting* features that are dependent on other features and the resulting RF similarity measure is build on the basis of *dependent* features.

If the number of images is N and the number of features computed for each image using previous step is M, then we have a matrix O of dimension $N \times M$ representing the original data to be clustered. The auxiliary matrix A with the same dimensions as O is created by randomly sampling values from the distribution of features in the original data. As shown in Figure 5.3, each row in A is created by randomly selecting values from the columns of O.

5.2.2 Random Forest based Document Similarity

In the next step, a two-class RF classifier is trained with the original data (O) as one class and the auxiliary data (A) as another. After training, if two documents i and j land in the *same terminal node* of the tree then the similarity between i and j is increased by 1 (Figure 5.2). Finally, the similarities are symmetrized and divided by the number of trees. We find that it is more effective to compute similarities using only the out-of-bag data in training.

5.2.3 Unsupervised Classification

For unsupervised classification, we can use any off-the-shelf clustering method which takes as input the similarity matrix, and outputs different partitions of data. In this work, we used Normalized-cuts [108].

When the number of classes is not known in advance, we use an *internal* cluster validation procedure called *silhouette* to determine the correct number of classes [107]. The silhouette measure for each point jointly evaluates: (1) how well the sample is matched to its current cluster, (2) how badly the sample is matched with neighboring cluster (closest cluster among all other clusters). Let $a(i)$ be the average dissimilarity between data i and other points in the same cluster. Let b(i) be the average dissimilarity of i with the data of neighboring cluster. Then silhouette coefficient of a single point is defined as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{5.1}$$

We compute dissimilarities by taking the negative log of the random forest based similarities computed in the previous step. Using Normalized-cuts, we obtain clustering results for different values of K, and compute the average *silhouette* for all samples in the dataset. The peak in the plot of average silhouette corresponds to more natural grouping (correct number of clusters). We have $-1 \leq s(i) \leq 1$.

5.3 Experiments

5.3.1 Competing Approaches

We compare our horizontal-vertical partitioning and RF based similarity computation (HVP-RF) with a number of baselines explained below:

Global BOW features: We compute a global histogram of codewords for the whole image without partitioning the image. The accuracy against this baseline demonstrates the advantage of our partitioning scheme for computing structural features at different levels. We compute similarities using both Euclidian distance (G-BOW-E) and Random Forest (G-BOW-RF) based approach.

Spatial-pyramid matching: We compare our Horizontal-vertical partitioning strategy with a spatial-pyramid matching proposed by Lazebenik *et al.* [88]. For

Table 5.1: Data sets used in experiments. N is the number of images. The last column specifies the number of categories in dataset.

Dataset	*N*	*#categories*
NIST Tax-forms	5590	20
Table dataset	824	53
FCC dataset	2413	12
Tobacco dataset	3482	10

similarity computation, we compare both our RF based similarity measure (SP-RF) against the Euclidian distance (SP-E) in the feature space. The accuracy against SP-RF and SP-E will signify the importance of Horizontal-vertical partitioning and Random-forest based structural similarity computation respectively.

We used Normalized-cuts [108] as the clustering method to demonstrate the quality of structural similarity computed using various approaches.

5.3.2 Datasets

Our first dataset is a collection of tax-form images obtained from National Institute of Standards and Technology (NIST). There are 20 categories of tax-forms, and a total of 5590 images in the dataset. Our second data set consists of images of different types of hand-drawn tables obtained from a field data. In this dataset, there are a total of 824 images with 53 different types of table structures. Our third data comes from a large collection of documents stored at Federal Communications Commission (FCC) which includes various reports, contracts and filings that broadcast TV stations are required to submit to the FCC. A total of 36,000 images were downloaded and 2413 were manually categorized into 12 classes to create a gold standard. Our fourth dataset is obtained from a large collection of images in the Tobacco dataset [101]. We annotated a total of 3482 images to obtain 10 categories of images. All four dataset is summarized in Table 5.1.

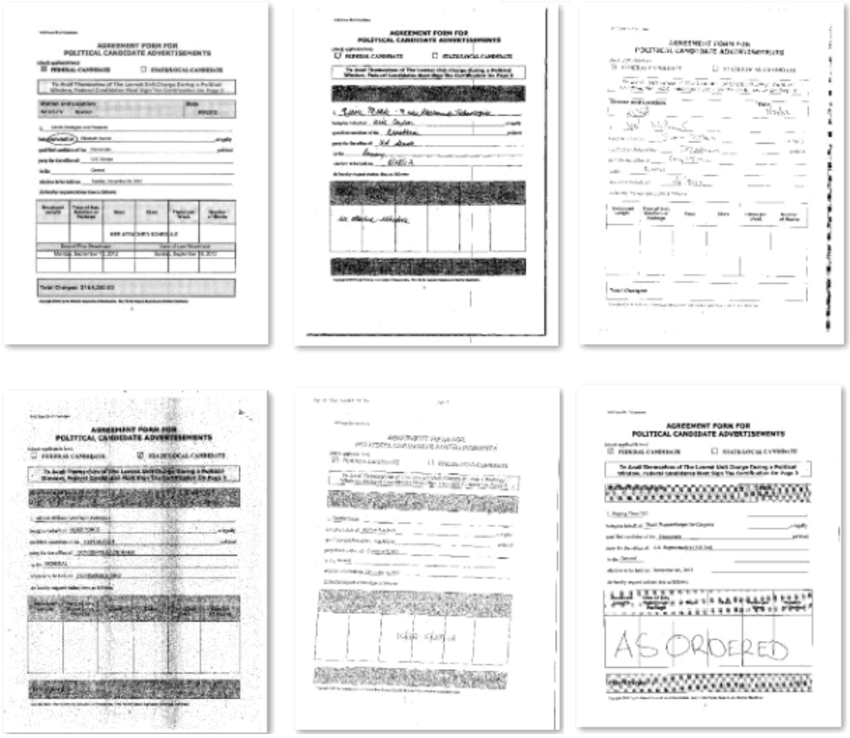


Figure 5.4: Samples from *Agreement* category in the FCC dataset, showing high variation in structure and degradation of documents.

The four datasets selected for experiments differ in the level of structural similarity present between images. For example, in the NIST Tax-form data set, images of a particular class have similar layout and the variation in content is local to the cells filled by user. Additionally, images differ due to the translation and rotation introduced during capture. In the Table dataset variation in structure is more since the tables are drawn and filled by users, and cell dimensions are only approximately matching. Figure 5.4 shows some sample images from FCC dataset.

As observed, there is a high variation in structure and quality of images in the same category. Variation in layout and content is maximum in the Tobacco dataset where the classes only match at conceptual level.

5.3.3 Evaluation Protocol

Most objective functions defined for clustering target of attaining a high-intra cluster similarity and low inter-cluster similarity. But optimal grouping based on this criteria do not necessarily correspond to optimal objective of user of applications. It is usually recommended to perform a direct evaluation based on application's objective but that may be very time consuming and expensive. When the the exact categories of all documents are known, either by a majority voting on human judgements or based on its structure or type, we call it a gold standard. We can then compute a criterion that evaluates how well the clustering matches the gold standard classes. In this work, we use following measures to evaluate our clustering results:

Cluster purity: Purity is a simple evaluation measure which compute how pure resulting clusters are with respect to categories. For computing purity, we first assign each cluster to the class which is most frequent in the cluster. Then, we compute the accuracy of this assignment by counting the number of correctly assigned documents and dividing by N:

$$purity(\Omega, C) = \frac{1}{N} \sum_{k} \max |w_k \cup c_k| \tag{5.2}$$

where The maximum value of purity is 1.

Rand index: When the number of clusters is large, it is easy to achieve a high purity, in particular, purity is 1 if each document is assigned a new label. Thus, purity alone cannot be used to assess the quality of the clustering against the number of clusters. Another interpretation of clustering is to view it as a series of decisions, one for each of the pairs of images in the dataset. A true positive (TP) decision groups two similar documents in the same cluster, a true negative (TN) decision assigns two dissimilar documents to different clusters. There are two types of errors the method can commit: First decision assigns two dissimilar documents to the same cluster (FP). Another decision assigns two similar documents to different clusters (FN). *Rand index* measures the accuracy of correct decisions:

$$RI = \frac{TP + TN}{FP + TP + FN + TN} \tag{5.3}$$

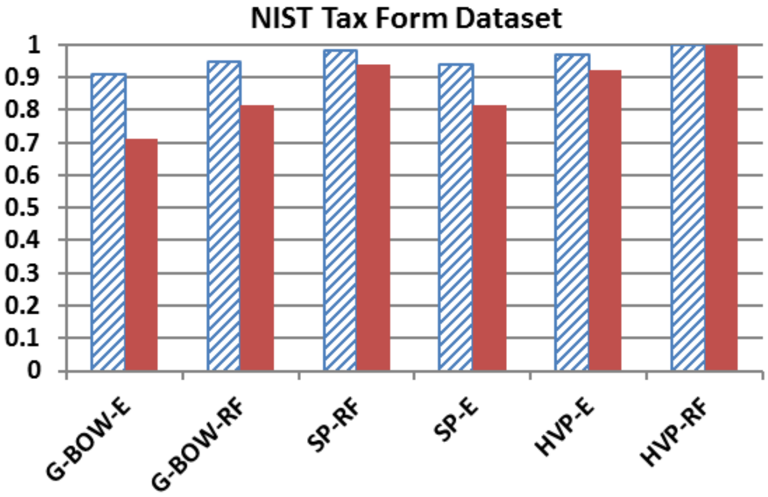


Figure 5.5: Adjusted Rand Index (striped bar in left) and purity (solid bar in right) using Normalized-Cuts for NIST Tax-form dataset.

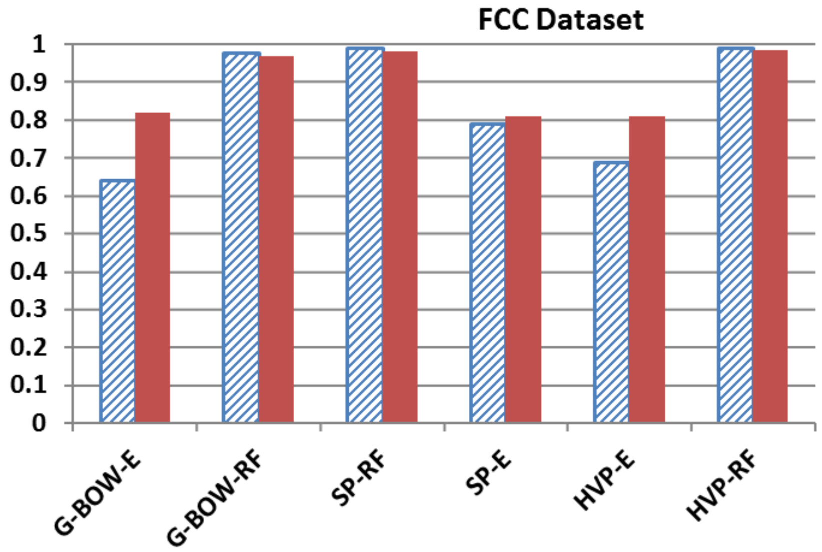


Figure 5.6: Adjusted Rand Index (striped bar in left) and purity (solid bar in right) using Normalized-Cuts for FCC dataset.

One problem with the Rand index is that the expected value of the Rand index of two random partitions does not take a constant value. The general form of an index with a constant expected value is:

$$GI = \frac{index - expectedindex}{maximumindex - expectedindex} \tag{5.4}$$

which is bounded above by 1, and takes the value 0 when the index equals its expected value. Let n_{ij} be the number of samples in both class u_i and cluster v_j. Let n_{iu} and n_{vj} be the total number of objects in class u_i and cluster v_j respectively. Then the adjusted rand index (ARI) is given by:

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{n_{iu}}{2}\sum_j \binom{n_{vj}}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{n_{iu}}{2} + \sum_j \binom{n_{vj}}{2}] - [\sum_i \binom{n_{iu}}{2}\sum_j \binom{n_{vj}}{2}]/\binom{n}{2}} \tag{5.5}$$

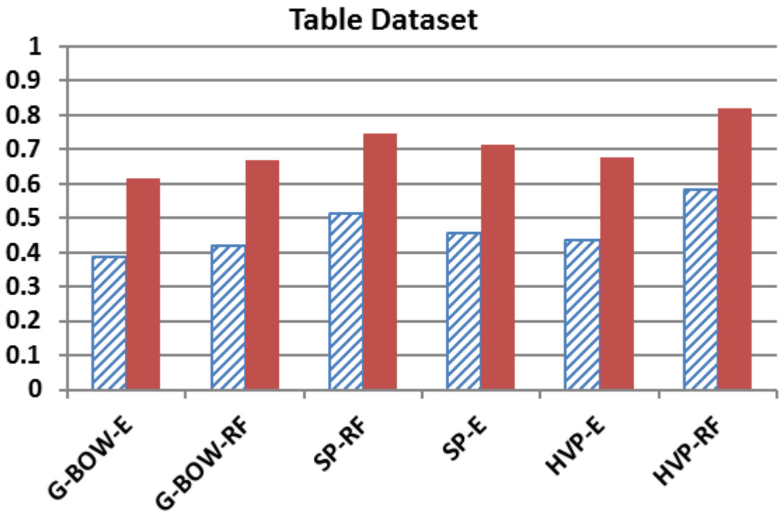


Figure 5.7: Adjusted Rand Index (striped bar in left) and purity (solid bar in right) using Normalized-Cuts for Table dataset.

5.3.4 Results

5.3.4.1 Known number of Classes

Figure 5.5 shows the scores of cluster purity and ARI for the Tax-form dataset. Our method (HVP-RF) achieves an ARI of 1.0 while the SP-RF achieves an ARI 0.98 when the number of clusters is set to 20. It is interesting to note that when the number of clusters are more than the actual number of tax-form categories (i.e. > 20), the proposed approach and SP-RF perform worse than baseline approaches. This may be due to the combined effect of following: (1) The NIST tax-form data set is imbalanced, i.e. the number of images is in each class varies a lot. (2) When the number of clusters is larger, and split is required, the threshold computed by Normalized-cuts favors splitting of larger groups resulting less optimal performance. We obtained a similar bar-chart for the FCC dataset shown in Figure 5.6. SP-

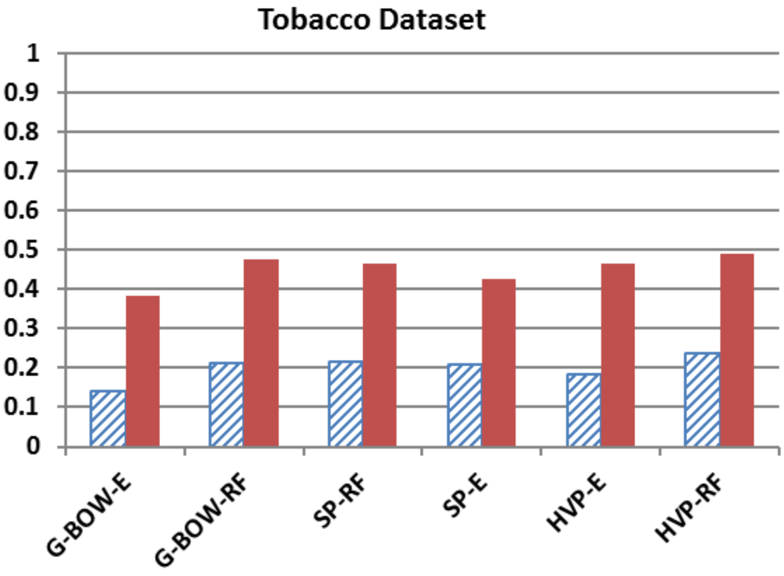


Figure 5.8: Adjusted Rand Index (striped bar in left) and purity (solid bar in right) using Normalized-Cuts for Tobacoo dataset.

RF and HVP-RF achieved a very competitive accuracy of 0.99 for purity and 0.98 for ARI on FCC data. The poor accuracy of Euclidian-distance based approaches clearly demonstrates that similarity computation using Frobenius Norm in high-dimensional feature space is ineffective.

For hand-drawn table images, the variation in structure is higher than tax-form images, and we observe a drop in accuracies achieved by all approaches. A maximum ARI of 0.59 and purity of 0.82 is obtained using our approach (Figure 5.7). The images in Tobacco dataset are only conceptually similar and there is relatively high variation among same class (such as Advertisement, Memo etc.) as compared to previous two datasets. As observed in Figure 5.8, none of the implemented approaches achieves a high ARI on this data set. We find that our approach performed better than SP-RF, SP-E and base-lines approaches in most of

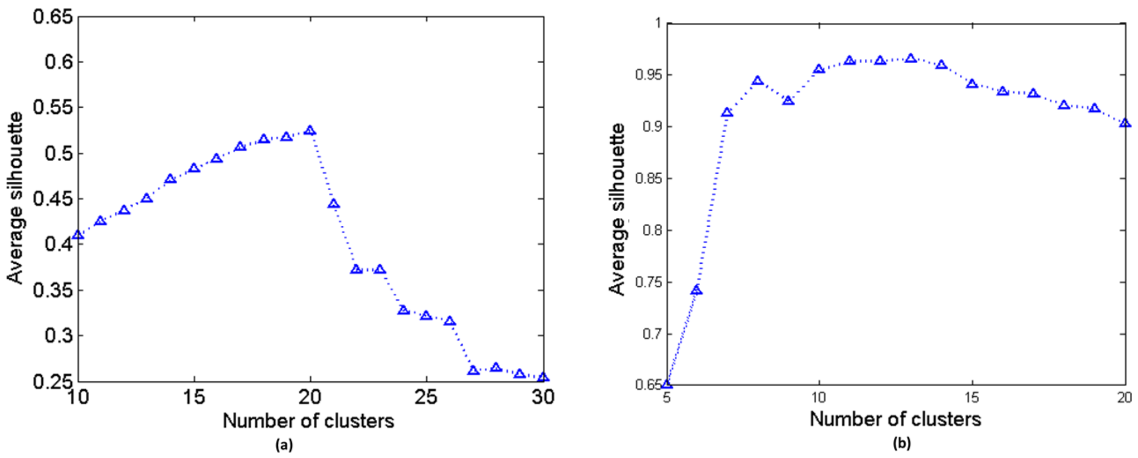


Figure 5.9: Average silhouette for (a) NIST tax form dataset (b) FCC dataset. Higher value corresponds to better grouping. The number of classes in ground-truth is 20 for tax form dataset and 12 for FCC dataset.

the cases.

In summary, we observed that the RF based similarity achieved better grouping compared to its Euclidian counterpart for both global and partitioning based approach. For the partitioning based approach, due to large number of features (high dimension) Euclidian-distance based similarity performs very poorly on all three data sets. Overall, the combination of Horizontal-vertical partitioning with RF based partitioning achieves the best accuracy when number of classes matches number of correct classes.

5.3.4.2 Unknown number of Classes

We used the similarities computed by our method (HVP-RF) to obtain different clustering results. Plots for mean silhouettes for the NIST tax-form, FCC dataset and Table images are shown in Figure 5.9(a),5.9(b) and 5.10 respectively. For tax-form images we observe a sudden drop in the average silhouette after $K = 20$. The

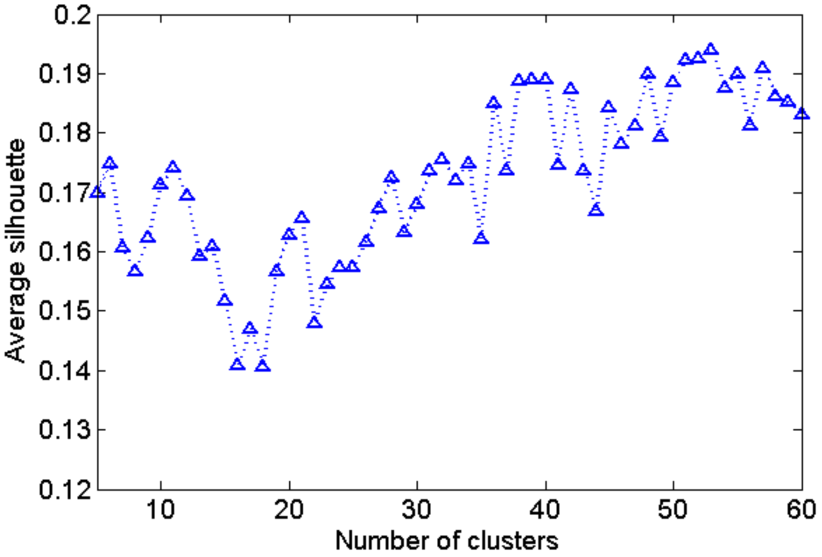


Figure 5.10: Average silhouette for Table dataset. Higher value corresponds to better grouping. Number of different table structures is 53 in the ground-truth.

peak clearly indicates the correct number of classes in the dataset. Similarly, for the FCC dataset the mean silhouette peaks when the cluster number is 11-13 and the values continue to drop till the last point. Figure 5.11 shows the plot of average silhouette based on euclidian distances for three methods (a) HVP-E (b) SP-E (c) G-BOW-E. The peaks in these plots do not indicate the correct number of clusters.

For Table dataset, we observed that this trend is not very consistent in different trials (change in clustering due to different initialization). In that case, we obtained an average of ten trials for Figure 5.10. Although the maximum silhouette value obtained at $K = 53$ indicates the correct number of classes in the Table dataset, we observe that other peaks (in the range 48-55) are quite close. Upon examining the table images we found that some classes constituted of 2-3 different table structure types and their membership was quite subjective. A similar trend was observed for

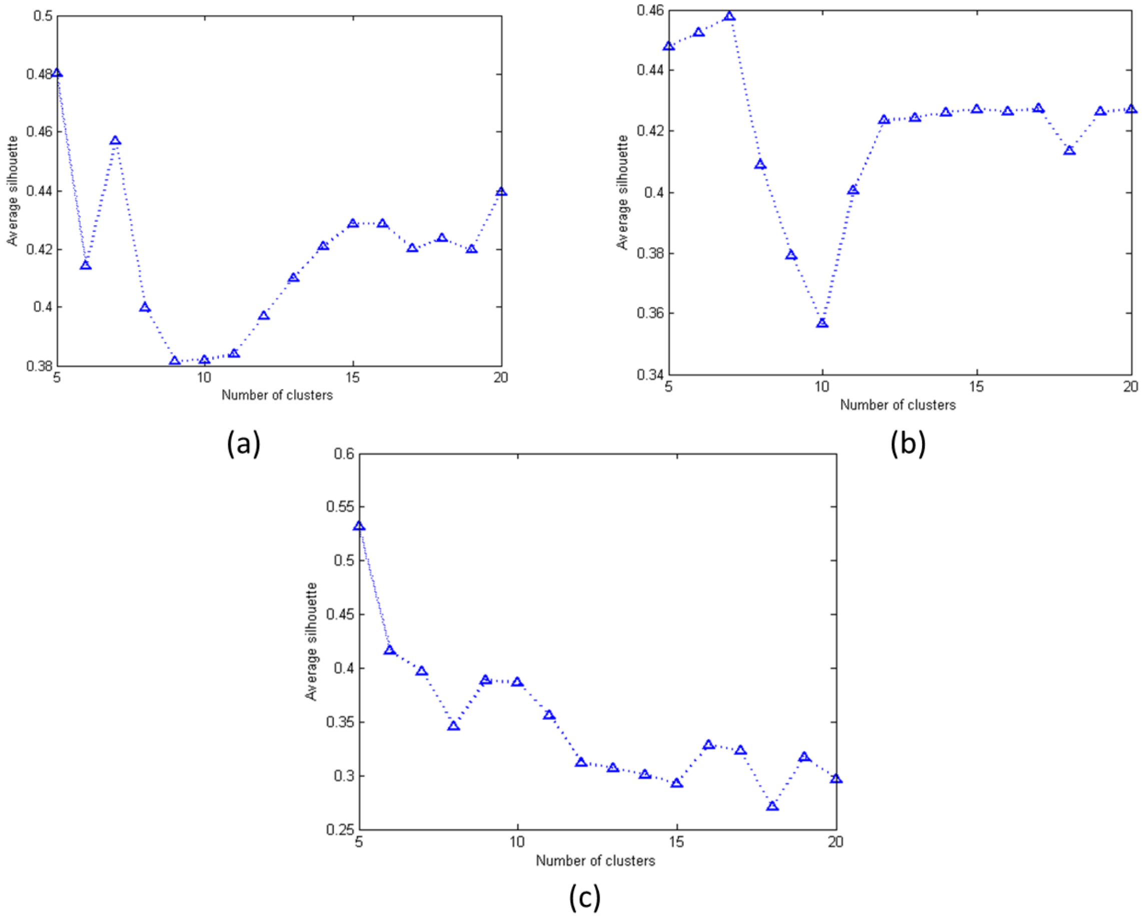


Figure 5.11: Average silhouette for FCC dataset. (a) HVP-E (b) SP-E (c) G-BOW-E. None of the approaches based on Euclidian distance peaks around the correct number of clusters.

Tobacco dataset, and multiple comparable peaks were observed in the range 8-12 (for K).

Chapter 6: Summary of Contributions and Open Problems

6.1 Data selection for SVMs and Applications

Summary: In Chapter 2, we presented a randomized approach for selecting the points from a large set of training points in order to do SVM learning. We first presented a new method for sampling points from the convex hull of the training data for one-class SVM learning. We then extended the idea of random subspaces to sample points from the extended support vector (ESV) set for the two-class problem. We also showed that the points obtained are being selected from the convex-hull for one-class, and from the Extended Support Vector set for the two-class problem. The $O(KN))$ time complexity of our method is better than some previously reported methods [42, 49]. For the two-class case, we showed that the time complexity can be reduced by using *Algorithm 1.* Our method is different from the previous approaches [42] in the sense that we do not target to compute the whole convex hull of data, which is because doing so is intractable for the large training set. We only focus on sampling the convex hull and the ESV by generating random subspaces of high dimensions which are less likely to be repeatable. Unlike some previous methods, we do not require the explicit knowledge of the dimension or the estimate of Support Vectors. On comparison with two other approaches for data selection we find that our approach reduces the training set for SVM more effectively.

As an application, we also presented a method for rule-line removal in document images using efficient integral-image based features and our data-selection method. We showed in our experiments that the integral-image features are effective for text/rule-line classification and the computation time for rule-line removal reduces significantly if we use an integral-image. We experimented with high resolution images (300-600 dpi) to demonstrate that using large scale learning techniques, pixel-based rule-line removal is feasible. The main insight was that by training two SVM classifiers with different running-time cost the pixel-level rule-line removal can be done efficiently.

Future Directions: Our work on data-selection and rule-line removal can be extended in the following ways:

1. **Multi-class data sampling:** Our two-class framework can be easily extended to perform a multi-class data selection by computing the projection error for different classes in a similar setting. We can either take a one-vs.-all or one-vs.-one approach like multi-class SVM. This will be useful in training SVM for multi-class classification for image categorization, super-pixel labeling for segmentation etc.

2. **Extension to Active-learning setting:** In our approach we assumed that the labels of training data are known for data selection. One can extend the idea of subspace based data selection when labels of data points are not

available. That approach is comparable to *Active learning* approaches which incrementally selects most *informative* to construct a model.

6.2 Handwritten Text-line Segmentation

Summary: In Chapter 3, we have presented a graph-based approach for the extraction of handwritten text-lines in monochromatic document images. In our approach, we used two different estimates from Affinity propagation and Breadth-first-search in a local-orientation graph to obtain text-lines. Using the same orientation graph we detect touching and proximity errors in our results. We then apply EM algorithm to correct errors. Our error detection and correction scheme relies on the same graph used for clustering and does not add any computational overhead. Our method is fast due to the removal of small components in the first step. We also presented an effective method for associating a diacritic/accent component to a text-line.

In experiments, we demonstrated the effectiveness of our method on different datasets including the standard ICDAR 2009 Segmentation competition dataset. On a more challenging Anfal dataset our method out-performed previous methods based on projection-profiles and steerable-filter. In general, our method can be used as a post-processing step in any connected-component-based method which gives an initial estimate of text-lines.

Future Directions: In the future, our work on text-line segmentation can be extended in following ways:

1. **Ensemble of multiple segmentation methods:** Due to high variation in handwritten text and script-specific characteristics it has been difficult to find optimal parameters which work across all variations of handwritten text. We observe that different segmentation methods or even parameterized variations of same method produces different *segmentation errors*. In our approach, we used estimates from breadth-first search and Affinity propagation. We can extend our method to use ensemble of multiple segmentation results obtained from applying different methods to improve our current approach. Since our approach is fast, obtaining multiple results with different parameters is computationally feasible. Additionally, we propose to use other fast approaches based on local projections and orientation filters to obtain different estimates. For ensemble learning, one can investigate the co-association approach which resembles the majority voting schemes commonly used in classifier ensembles. A pair of components occurring in the same base cluster signifies a "vote" for the pair being co-located in the final cluster. The results collected from different base clusterings can be mapped to a symmetric $n \times n$ co-association matrix M, where each entry M_{ij} represents the fraction of times that the pair of components (x_i, x_j) has been assigned to the same cluster. In the next step, a standard similarity-based clustering method like agglomerative clustering [118] may be applied to produce an ensemble solution.

2. **Purposive validation of text-line segmentation:** In our current approach, we have used a matching-score based criteria for evaluation of text-line

results. In future, we plan to obtain the end results (OCR or other metrics based on application) and evaluate our approach. The work was originally under the DARPA MADCAT program and we are using the OCR and machine-translation evaluation pipeline for our experiments.

6.3 Structural similarity for Retrieval and Classification

In Chapter 4, we defined structural similarity for the retrieval and classification of document images. In our approach, similarity is captured at different levels. At the local level, we used robust SURF features to capture content similarity. Our recursive partitioning scheme along with histograms of codewords in different partitions provide another level of structure match. Our results on four diverse real-world problems demonstrate that our approach for modeling spatial relationships is effective for both coarse and fine-grained document image classification. We compared our random forest based approach with SVM and showed that RF has a superior performance when the number of features is large and the number of labeled documents is limited for training. In most cases, we showed that an effective and efficient model for retrieval can be learned using only a few labeled example documents.

In Chapter 5, we extended our approach to another similar problem of unsupervised classification of structurally similar document images. Similar to our approach to supervised case, the image is recursively divided into vertical and horizontal partitions and histograms of dictionary atoms are computed for each partition in the next step. Using an auxiliary data constructed by randomly sampling features from the original data, we train a random forest classifier to compute the structural similarities between images. We used a measure called *silhouette*-coefficient to estimate the number of classes. Our results showed that the our partitioning scheme for structural feature computation combined with random forest provides an effective way to compute similarities between images for grouping. We demonstrated the effectiveness of our approach using four real-world data sets. On the two datasets, NIST-tax forms and FCC, our approach obtained almost 100% accuracy on grouping, and outperformed other existing approaches based on spatial-pyramid and Euclidian distance.

In future, our work can be extended in following ways, in support of *Analysts* whose goal is to see the most relevant images first.

1. **Semi-supervised classification of structurally similar document images:** We can further extend the completely unsupervised grouping of document images to incorporate labels of few samples from each group. This is a very realistic scenario when randomly few sample images are selected and categorized before actual clustering is performed. Obtaining a few labels is not a costly and time-consuming processing and it has been shown to aid the ”unsupervised” clustering to obtain better results. In this setting, we have *must-link* and *cannot-link* constraints between labeled examples for which we can update the similarity-matrix as follows: One simple approach can be to update the similarities in the obtained similarity matrix from Random Forest. Similarity values for must-link pairs can be replaced with the maximum

of similarity matrix (or 1), and cannot-link pairs with minimum of similarity matrix (or 0). In this way, one can guarantee that must-link pairs are grouped together, and cannot-link pairs are grouped differently. But this approach ignores the label information during RF training. As an extension, one can incorporate the label information during RF training as follows: In the creation of second auxiliary class for RF classifier, instead of randomly sampling the features, we can sample features based on labels of provided data. This can lead to better decisions in Random forest training.

2. **User guided clustering:** Since humans are very good at providing intermediate constraints for image grouping (such as whether two images belong to the same category or not), we will explore the possibilities of incorporating such constraints during clustering of document images. It is important, however, to minimize the number of questions asked of the users [119]. This approach of performing constrained clustering in active learning framework has been of growing interest in computer vision [120, 119]. It helps in creating datasets for a particular analysis in a very short time. In our approach, we will seek pairwise constraints from users to know if the two selected images belong to the same class or not. Most of the previous work in active learning have focused on selecting a single pair of unlabeled example in each iteration. Instead, we will explore *batch mode active learning* to find more than one constraints in each iteration to expedite the process of clustering.

6.4 List of Publications

Data selection for SVMs and Applications:

1. Jayant Kumar and David Doermann. *Fast Rule-line Removal using Integral Images and Support Vector Machines.* International Conference on Document Analysis and Recognition (**ICDAR**), pp. 584-588, 2011.

2. W. Abd-Almageed, Jayant Kumar and David Doermann. *Page Rule-Line Removal using Linear Subspaces in Monochromatic Handwritten Arabic Documents.* International Conference on Document Analysis and Recognition (**ICDAR**), pp. 768-772, 2009.

Handwritten Text-line Segmentation:

3. Jayant Kumar, W. Abd-Almageed, Le Kang and David Doermann. *Handwritten Arabic Text Line Segmentation using Affinity Propagation.* International Workshop on Document Analysis Systems (**DAS**), pp. 135-142, 2010.

4. Jayant Kumar, Le Kang, David Doermann and Wael Abd-Almageed. *Segmentation of Handwritten Textlines in Presence of Touching Components.* Intl. Conf. on Document Analysis and Recognition (**ICDAR**), pp. 109-113, 2011.

5. Le Kang, Jayant Kumar, Peng Ye and David Doermann. *Learning Text-line Segmentation using Codebooks and Graph Partitioning.* International Conference on Frontiers in Handwriting Recognition (**ICFHR**), pp. 63-68, 2012.

Structural similarity based classification and retrieval:

6. Jayant Kumar, Peng Ye and David Doermann. *Learning Document Structure for Retrieval and Classification.* International Conference on Pattern Recognition (**ICPR**), pp. 1558-1561, 2012.

7. Jayant Kumar and David Doermann. *Unsupervised Classification of Structurally Similar Document Images*, International Conference on Document Analysis and Recognition (**ICDAR**), pp. 1257-1261, 2013.

8. Jayant Kumar, Peng Ye and David Doermann. *Structural Similarity for Retrieval and Classification of Document Images*, Pattern Recognition Letters (**PRL**) (Special Issue for best papers) (To appear).

Zone classification and labeling:

9. Jayant Kumar, Rohit Prasad, Huiagu Cao, W. Abd-Almageed, David Doermann and Prem Natarajan. *Shape Codebook based Handwritten and Machine Printed Text Zone Extraction.* International Conference on Document Recognition and Retrieval (**DRR**), pp. 7874:1-8, 2011.

10. Jayant Kumar, Jaishanker Pillai and David Doermann. *Document Image Classification and Labeling using Multiple Instance Learning.* International Conference on Document Analysis and Recognition (**ICDAR**), pp. 1059-1063, 2011.

6.5 Planned Publications

1. Jayant Kumar, Le Kang, W. Abd-Almageed and David Doermann. *Beyond Coarse Segmentation: Association of Diacritic and Accent Components to Handwritten Lines*, International Journal on Document Analysis and Recognition (**IJDAR**) (to be submitted).

2. Jayant Kumar and David Doermann. *Grouping Structurally Similar Document Images: Unsupervised and Semi-supervised Methods*, IEEE Transactions on Pattern Analysis and Machine Intelligence (**PAMI**)(to be submitted).

Appendix A

A.1 Proof of Claims

A.1.1 Claim 1

Proof. Consider any arbitrary point T in the set CHS returned by the proposed method. Assume $T \notin CH_P$. Geometrically speaking, a convex hull can be visualized as a D-dimensional convex polytope that contains all the points in P. Figure 1(a)(left) shows the simple two-dimensional case. Let L be the line (subspace of dimension D-1) for which the point T was selected. Since L was built incrementally, it must pass through at least two points from the set P, say Q_1 and Q_2. Let M_1 and M_2 be the points in CH_P such T lies between L and the line M_1M_2. This must always be possible as L passes through the interior of the convex polygon C and T lies inside C; hence there must exist the points M_1 and M_2. Construct, M_1P_1 ,TP_2 and M_2P_3 perpendicular to L. Extend P_2T to meet M_1M_2 in $T^{'}$. Now, consider the trapezoid $M_1P_1P_3M_2$: We have,

$$M_1P_1 < T'P_2 < M_2P_3 \implies TP_2 < T^{'}P_2 < M_2P_3 \implies TP_2 < M_2P_3$$

Hence, we arrived at the conclusion that the projection error for M_2 (actually one of M_1 or M_2) for L is more than that for T, which is a contradiction. Hence T must be member of CH_P. More formally, for each point $T \in CHS$ there exists a subspace of dimension $d \leq D - 1$ for which T will assume the maximum projection error. In higher dimensions, the line L of the 2D case is replaced by the hyperplane L (subspace of dimension d) which passes through the interior of the convex polytope and a similar argument can be given.

□

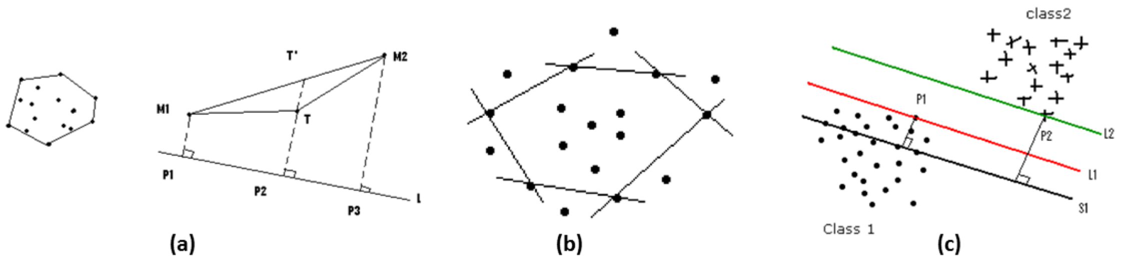


Figure 1: Two-dimensional scenarios for proofs of (a) claim 1 (b) claim 2 (c) claim 3

A.1.2 Claim 2

Proof. Denote the set of the adjacent vertices of T in CH_P by $N(T)$. Consider the hyperplanes formed by the points which are neighbors of T. We know that a subspace of dimension d-1 divides a subspace of dimension d into two halfs. Hence, there exists a subset $S \subseteq N(T)$ and $|S| = d$ such that the hyperplane formed by N(T) will have T as the maximum projection error in one of the halves. Since $N(T) \in P$ there exists a nonzero probability of getting this subspace of dimension d-1, and hence of selecting T. Figure 1(b) shows the case for d=2. □

A.1.3 Claim 3

Proof. Consider the case when $Q_d = P_d$, i.e. both datasets lie in same subspace of dimension d'. Figure 1(c) shows the case for $d = 2$. $S_1 = S1$ is a random subspace and $P1 \in P_Q$, $P2 \in Q$ be the points having projection error e_{maxP} and e_{minQ}. Let

L1 and L2 be the subspace obtained by translating S1 in orthogonal direction to the points P1 and P2 respectively. Due to the convex property there do not exist any point of either class which lies between L1 and L2 and hence L1 and L2 are possible separating hyperplane for P and Q. Immediately, this makes $P1 \in ESV$ and $P2 \in ESV$. Similar construction can be used to argue for the cases when $Q_d > P_d$ and $Q_d < P_d$. □

A.2 Data Selection Algorithm

Algorithm 1 ConvHullSample(P, K, C)

```
Require: $P \in \mathbb{R}^{DXN}$ {Matrix of dimension DxN}
Require: K - Number of selected points, C - Threshold for reconstruction error
  Initialize $k \leftarrow 0$, Set $CHS = \Phi$ { Selected points to be returned}
  repeat
    Randomly select a feature vector $v_k \in P$
    Initialize $d \leftarrow 0$ {Dimension of Subspace}
    $S_d = [v_k / \parallel v_k \parallel]$ {Subspace $S_d$ consisting of just $v_k$}
    Add v = $\{v_i | e_{vi} > e_{vj} \quad \forall v_j \in P\}$ to $CHS$ {$e_{vj}$ is projection error to $S_d$}
    $k \leftarrow k + 1$
    for $i = 1$ to $N - 1$ do
      Randomly select a feature vector $v_i \in$ P
      Compute projection error $e_{vi}$ on $S_d$ :
      $v_p \leftarrow {S_d}^T v_i$
      $r_v \leftarrow S_d * v_p$ {Reconstructed Vector}
      $e_{vi} \leftarrow d(r_v, v_i)$ {Euclidian distance}
      if $e_{vi} > C$ then
        $v_{res} \leftarrow r_v - v_i$ {Residual vector}
        $S = [S \quad v_{res} / \parallel v_{res} \parallel]$ {Update the subspace $S$ with unit norm residual vector}
        Update $d \leftarrow d + 1$
        Find $v_{max} = \{v_i | e_i \geq e_j \quad \forall v_j \in P'_S\}$ // $e_i$ : projection error of points not in S
        Add $v_{max}$ to $CHS$
        $k \leftarrow k + 1$
      end if
    end for
    $Q \leftarrow S_{d-1}$ {$S_d$ is subspace containing all points in P}
    Select one more point from the half-space of $S_d$ formed by $S_{d-1}$ which was not considered previously
    Add new point to $CHS$ {Two points for $S_{d-1}$ is selected}
    $k \leftarrow k + 1$
  until k = K
  return $CHS$
```

Algorithm 2 ExtendedSupportVectorSampleTwoClass(P, Q, K, C)

```
Require: $P \in \mathbb{R}^{DXN1}$, $Q \in \mathbb{R}^{DXN2}$, K - No. of points to be selected, C - Reconstruction error
  Initialize set $ESV_P = \Phi$, $ESV_Q = \Phi$, $k \leftarrow 0$
  repeat
    Set $d \leftarrow 0$ {Dimension of Subspace S of P}
    Randomly select a feature vector $v_k$ from P
    $S_d = [v_k / \| v_k \|]$ {Subspace $S$ consisting of only $v_k$}
    for $i = 1$ to $N$ do
      Randomly select a feature vector $v_i \in$ P
      Compute the projection error $e_i$ of $v_i$ on $S_d$ :
      $v_p \leftarrow {S_d}^T v_i$
      $r_v \leftarrow S_d * v_p$ {Reconstructed vector}
      $e_i \leftarrow d(r_v, v_i)$ {Euclidian distance}
      if $e_i > C$ then
        $v_{res} \leftarrow r_v - v_i$ {Residual vector}
        $S_{d+1} \leftarrow [S_d \quad v_{res} / \| v_{res} \|]$
        Update $d \leftarrow d + 1$
      end if
    end for
    $S \leftarrow S_{d'-1}$ {$S_{d'}$ is subspace of dimension $d'$ obtained incrementally}
    Find $e_{maxP} = \{e_{i1} | e_{i1} \geq e_{j1} \quad \forall v_{j1} \in P_Q\}$ //Maximum projection error to S from $P_Q \subseteq P$
    Find $e_{minQ} = \{e_{i2} | e_{i2} \geq e_{j2} \quad \forall v_{j2} \in Q\}$ //Minimum projection error to S from Q
    if $e_{max1} \leq e_{min2}$ then
      Add $v_{iP}$ to $ESV_P$ and $v_{iQ}$ to $ESV_Q$ {Member of Extended Support Vector set}
      $k \leftarrow k + 1$
    end if
  until k=K
  return $ESV_P$ and $ESV_Q$ { Set of selected points form Extended Support Vector set}
```

Bibliography

[1] B. Gatos, N. Stamatopoulos, and G. Louloudis. ICDAR2009 handwriting segmentation contest. *International Journal on Document Anaysis Recognition*, 14(1):25–33, March 2011.

[2] Zhixin Shi, S. Setlur, and V. Govindaraju. A steerable directional local profile technique for extraction of handwritten arabic text lines. In *International Conference on Document Analysis and Recognition*, pages 176–180, 2009.

[3] Simone Marinai, Marco Gori, and Giovanni Soda. Artificial neural networks for document analysis and recognition. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 27(1):23–35, 2005.

[4] Yann Lecun, Lon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *IEEE proceedings*, pages 2278–2324, 1998.

[5] R. Zanibbi, D. Blostein, and J.R. Cordy. A survey of table recognition: Models, observations, transformations, and inferences. *International Journal of Document Analysis and Recognition*, 7:1–16, 2003.

[6] Cheng-Lin Liu, Kazuki Nakashima, Hiroshi Sako, and Hiromichi Fujisawa. Handwritten digit recognition: benchmarking of state-of-the-art techniques. *Pattern Recognition*, 36(10):2271 – 2285, 2003.

[7] T. Plötz and G.A. Fink. Markov models for offline handwriting recognition: a survey. *International Journal on Document Analysis and Recognition*, 12(4):269–298, 2009.

[8] Simone Marinai and Hiromichi Fujisawa, editors. *Machine Learning in Document Analysis and Recognition*, volume 90 of *Studies in Computational Intelligence*. Springer, 2008.

[9] Kamran Etemad, David Doermann, and Rama Chellappa. Multiscale segmentation of unstructured document pages using soft decision integration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):92–96, 1997.

[10] Henry S Baird and Matthew R Casey. Towards versatile document analysis systems. In *International Workshop on Document Analysis Systems*, pages 280–290, 2006.

[11] Henry S Baird, Michael A Moll, Jean Nonnemaker, Matthew R Casey, and Don L Delorenzo. Versatile document image content extraction. In *Electronic Imaging*, pages 60–67, 2006.

[12] Xujun Peng, Srirangaraj Setlur, Venu Govindaraju, and Ramachandrula Sitaram. Overlapped text segmentation using markov random field and aggregation. In *International Workshop on Document Analysis Systems*, pages 129–134, 2010.

[13] Yefeng Zheng, Huiping Li, and David Doermann. Machine printed text and handwriting identification in noisy document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):337–353, 2004.

[14] C. Cortes and V. N. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[15] Zhidong Lu, Richard Schwartz, Premkumar Natarajan, Issam Bazzi, and John Makhoul. Advances in the bbn byblos ocr system. In *International Conference on Document Analysis and Recognition*, pages 337–342, 1999.

[16] Huaigu Cao, Rohit Prasad, and Prem Natarajan. A stroke regeneration method for cleaning rule-lines in handwritten document images. In *International Workshop on Multilingual OCR*, pages 1–10, 2009.

[17] K. R. Arvind, J. Kumar, and A. G. Ramakrishnan. Line Removal and Restoration of Handwritten Strokes. In *International Conference on Computational Intelligence and Multimedia Applications*, pages 208–214, 2007.

[18] J. Kumar, L. Kang, D. Doermann, and W. Abd Almageed. Segmentation of Handwritten Textlines in Presence of Touching Components. In *International Conference on Document Analysis and Recognition)*, pages 109–113, 2011.

[19] Mudit Agrawal and David Doermann. Voronoi++: A Dynamic Page Segmentation approach based on Voronoi and Docstrum features. In *International Conference on Document Analysis and Recognition*, pages 1011–1015, 2009.

[20] P. Viola and M. Jones. Robust real-time face detection. *International Journal on Computer Vision*, pages 137–154, 2004.

[21] J. Kumar and D. Doermann. Fast Rule-line Removal using Integral Images and Support Vector Machines. In *International Conference on Document Analysis and Recognition*, pages 584–588, 2011.

[22] Robert M. Haralick. Document image understanding: Geometric and logical layout. In *International Conference on Computer Vision and Pattern Recognition*, pages 385–390, 1994.

[23] Kyong-Ho Lee, Yoon-Chul Choy, and Sung-Bae Cho. Geometric structure analysis of document images: A knowledge-based approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1224–1240, November 2000.

[24] J. Kumar, T. Kasar, and AG Ramakrishnan. Edge-based connected component approach for skew correction of complex document images. In *IEEE Region 10 Conference TENCON*, pages 1–4, 2007.

[25] K.R. Arvind, J. Kumar, and A.G. Ramakrishnan. Entropy based skew correction of document images. In *International conference on Pattern recognition and Machine Intelligence*, pages 495–502, 2007.

[26] Yi Li, Yefeng Zheng, David Doermann, Stefan Jaeger, and Yi Li. Script-independent text line segmentation in freestyle handwritten documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:1313–1329, 2008.

[27] Stephane Nicolas, Thierry Paquet, and Laurent Heutte. Text line segmentation in handwritten document using a production system. In *International Workshop on Frontiers in Handwriting Recognition*, pages 245–250, 2004.

[28] U. Pal and S. Datta. Segmentation of bangla unconstrained handwritten text. In *International Conference on Document Analysis and Recognition*, pages 1128–1132, 2003.

[29] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.

[30] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to algorithms. *MIT Press and McGraw-Hill*, 2009.

[31] A. Dengel and F. Dubiel. Clustering and classification of document structure-a machine learning approach. In *International Conference on Document Analysis and Recognition*, volume 2, pages 587–591, 1995.

[32] C. Shin and D. Doermann. Document image retrieval based on layout structural similarity. In *International Conference on Image Processing, Computer Vision and Pattern Recognition*, pages 606 – 612, 2006.

[33] Guangyu Zhu, Yefeng Zheng, D. Doermann, and S. Jaeger. Signature detection and matching for document image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):2015 –2031, 2009.

[34] S. Marinai, B. Miotti, and G. Soda. Digital libraries and document image retrieval techniques: A survey. *Learning Structure and Schemas from Documents*, pages 181–204, 2011.

[35] R. Jain and D. Doermann. Logo retrieval in document images. In *International Workshop on Document Analysis Systems*, pages 135–139, 2012.

[36] Siyuan Chen, Yuan He, Jun Sun, and Satoshi Naoi. Structured document classification by matching local salient features. In *International Conference on Pattern Recognition*, pages 653–656, 2012.

[37] David Doermann. The indexing and retrieval of document images: A survey. *Computer Vision and Image Understanding*, 70(3):287–298, 1998.

[38] G. Zhu and D. Doermann. Logo matching for document image retrieval. In *International Conference on Document Analysis and Recognition*, pages 606–610, 2009.

[39] Zhaohui Zheng, Xiaoyun Wu, and Rohini Srihari. Feature selection for text categorization on imbalanced data. *SIGKDD Explor. Newsl.*, 6(1):80–89, June 2004.

[40] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. *European Conference on Computer Vision*, pages 404–417, 2006.

[41] K.P. Bennett and E.J. Bredensteiner. Duality and geometry in SVM classifiers. In *International workshop and conference machine learning*, pages 57–64, 2000.

[42] E. Osuna and O. De Castro. Convex hull in feature space for support vector machines. *Advances in Artificial Intelligence*, pages 411–419, 2002.

[43] D. Caragea, A. Silvescu, and V. Honavar. Agents that learn from distributed dynamic data sources. In *International Workshop on Learning Agents, Agents*, pages 53–61, 2000.

[44] W. Abd-Almageed, J. Kumar, and D. Doermann. Page rule-line removal using linear subspaces in monochromatic handwritten arabic documents. In *International Conference on Document Analysis and Recognition*, pages 768–772, 2009.

[45] Jiun-Lin Chen and Hsi-Jian Lee. An efficient algorithm for form structure extraction using strip projection. *Pattern Recognition*, 31(9):1353–1368, 1998.

[46] J. N. Said, M. Cheriet, and C. Y. Suen. Dynamical morphological processing: A fast method for base line extraction. In *International Conference on Pattern Recognition*, volume 2, pages 8–13, 1996.

[47] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[48] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. *Technical report MSR-TR-98-14, Microsoft Research*, 1998.

[49] W. Zhang and I. King. Locating support vectors via β-skeleton technique. In *International Conference on Neural Information Processing*, volume 3, pages 1423–1427, 2002.

[50] S. Abe and T. Inoue. Fast training of support vector machines by extracting boundary data. *Artificial Neural Networks*, pages 308–313, 2001.

[51] Y. J. Lee and O.L. Mangasarian. RSVM: Reduced support vector machines. In *First SIAM International Conference on Data Mining*, pages 5–7, 2001.

[52] Y.J. Lee and S.Y. Huang. Reduced support vector machines: A statistical theory. *IEEE Transactions on Neural Networks*, 18(1):1–13, 2007.

[53] J. Wang, P. Neskovic, and L. Cooper. Training data selection for support vector machines. *Advances in Natural Computation*, pages 421–421, 2005.

[54] Daniel Lopresti and Ergina Kavallieratou. Ruling line removal in handwritten page images. In *International Conference on Pattern Recognition*, pages 2704–2707, 2010.

[55] Zhixin Shi, Srirangaraj Setlur, and Venu Govindaraju. Removing rule-lines from binary handwritten arabic document images using directional local profile. In *International Conference on Pattern Recognition*, pages 1916–1919, 2010.

[56] D. Tax and R. Duin. Data domain description using support vectors. In *European Symposium on Artificial Neural Networks*, volume 256, 1999.

[57] Jongwoo Lim, David Ross, Ruei sung Lin, and Ming hsuan Yang. Incremental learning for visual tracking. In *Advances in Neural Information Processing Systems*, pages 793–800. MIT Press, 2005.

[58] Michael Shindler, Alex Wong, and Adam W Meyerson. Fast and accurate k-means for large datasets. In *Advances in Neural Information Processing Systems*, pages 2375–2383, 2011.

[59] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, and A.Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):881–892, 2002.

[60] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[61] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[62] Jayant Kumar, Wael Abd-Almageed, and David Doermann. Dataset: Rule-line removal in handwritten arabic document images,. In *Laboratory for Language and Media Processing*, 2011. http://lampsrv02.umiacs.umd.edu/projdb/.

[63] Manivannan Arivazhagan, Harish Srinivasan, and Sargur Srihari. A statistical approach to line segmentation in handwritten documents. In *International Conference on Document Recognition and Retrieval XIV SPIE*, pages 6500T–6500T, 2007.

[64] Yi Li, Yefeng Zheng, David Doermann, and Stefan Jaeger. A new algorithm for detecting text line in handwritten documents. In *International Workshop on Frontiers in Handwriting Recognition*, pages 35–40, 2006.

[65] R. Plamondon and S.N. Srihari. Online and off-line handwriting recognition: a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 22(1):63–84, 2000.

[66] V. Papavassiliou, T. Stafylakis, V. Katsouros, and G. Carayannis. Handwritten document image segmentation into text lines and words. *Pattern Recognition*, 43(1):369–377, 2010.

[67] Berrin Yanikoglu and Peter A Sandon. Segmentation of off-line cursive handwriting using linear programming. *Pattern Recognition*, 31(12):1825–1833, 1998.

[68] Yi Lu. Machine printed character segmentation; an overview. *Pattern Recognition*, 28(1):67–80, 1995.

[69] Zaidi Razak, Khansa Zulkiflee, Mohd Yamani Idna Idris, Emran Mohd Tamil, Mohd Noorzaily Mohamed Noor, Rosli Salleh, Mohd Yaakob, Zulkifli Mohd Yusof, and Mashkuri Yaacob. Off-line handwriting text line segmentation: A review. *International Journal of computer science and network security*, 8(7):12–20, 2008.

[70] Alireza Alaei, Umapada Pal, and P Nagabhushan. A new scheme for unconstrained handwritten text-line segmentation. *Pattern Recognition*, 44(4):917–928, 2011.

[71] G. Louloudis, B. Gatos, and C. Halatsis. Text line detection in unconstrained handwritten documents using a block-based hough transform approach. In *International Conference on Document Analysis and Recognition*, volume 2, pages 599–603, 2007.

[72] Xiaojun Du, Wumo Pan, and Tien D Bui. Text line segmentation in handwritten documents using mumford–shah model. *Pattern Recognition*, 42(12):3136–3145, 2009.

[73] A. Zahour, B. Taconet, P. Mercy, and S. Ramdane. Arabic handwritten text-line extraction. In *International Conference on Document Analysis and Recognition*, pages 281–285, 2001.

[74] Anguelos Nicolaou and Basilios Gatos. Handwritten text line segmentation by shredding text into its lines. In *International Conference on Document Analysis and Recognition*, pages 626–630, 2009.

[75] Laurence Likforman-Sulem, Abderrazak Zahour, and Bruno Taconet. Text line segmentation of historical documents: a survey. *International Journal of Document Analysis and Recognition*, 9(2-4):123–138, 2007.

[76] Fei Yin and Cheng-Lin Liu. Handwritten chinese text line segmentation by clustering with distance metric learning. *Pattern Recognition*, 42(12):3146–3157, 2009.

[77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. B*, 39:1–38, 1977.

[78] L. Kang and D. Doermann. Template based Segmentation of Touching Components in Handwritten Text Lines. In *International Conference on Document Analysis and Recognition*, pages 569–573, 2011.

[79] L. Kang, D. Doermann, H. Cao, R. Prasad, and P. Natarajan. Local Segmentation of Touching Characters using Contour based Shape Decomposition. In *International workshop on Document Analysis Systems*, pages 460–464, 2012.

[80] L. Kang, J. Kumar, P. Ye, and D. Doermann. Learning Text-line Segmentation using Codebooks and Graph Partitioning. In *International Conference on Frontiers in Handwriting Recognition*, pages 63–68, 2012.

[81] Jayant Kumar, Le Kang, Wael Abd-Almageed, and David Doermann. Dataset: Handwritten arabic textline segmentation and proximity datasets. In *Laboratory for Language and Media Processing*, 2011. http://lampsrv02.umiacs.umd.edu/projdb/project.php?id=65.

[82] Vasant Manohar, Shiv N. Vitaladevuni, Huaigu Cao, Rohit Prasad, and Prem Natarajan. Graph clustering-based ensemble method for handwritten text line segmentation. In *International Conference on Document Analysis and Recognition*, pages 574–578, 2011.

[83] K. Collins-Thompson and R. Nickolov. A clustering-based algorithm for automatic document separation. In *SIGIR Workshop on Information Retrieval and OCR: From Converting Content to Grasping, Meaning*, pages 1–8, 2002.

[84] J. Kumar, R. Prasad, H. Cao, W. Abd-Almageed, D. Doermann, and P. Natarajan. Shape Codebook based Handwritten and Machine Printed Text Zone Extraction. In *International Conference on Document Recognition and Retrieval*, pages 7874:1–8, 2011.

[85] J. Kumar, P. Ye, and D. Doermann. Learning Document Structure for Retrieval and Classification. In *International Conference on Pattern Recognition*, pages 1558–1561, 2012.

[86] G. Joutel, V. Eglin, S. Bres, and H. Emptoz. Curvelets based queries for CBIR application in handwriting collections. In *International Conference on Document Analysis and Recognition*, volume 2, pages 649 –653, 2007.

[87] S. Marinai, E. Marino, and G. Soda. Tree clustering for layout-based document image retrieval. In *Second International Conference on Document Image Analysis for Libraries*, pages 9–18, 2006.

[88] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2169 – 2178, 2006.

[89] J. L. Decurtins and E. C. Chen. Keyword spotting via word shape recognition. In *SPIE's Symposium on Electronic Imaging: Science and Technology*, pages 270–277, 1995.

[90] Nawei Chen and Dorothea Blostein. A survey of document image classification: problem statement, classifier architecture and performance evaluation. *International Journal Document Analysis Recognition*, 10(1):1–16, 2007.

[91] T. Kochi and T. Saitoh. User-defined template for identifying document type and extracting information from documents. In *International Conference on Document Analysis and Recognition*, pages 127–130, 1999.

[92] A.D. Bagdanov and M. Worring. Fine-grained document genre classification using first order random graphs. In *International Conference on Document Analysis and Recognition*, pages 79–83, 2001.

[93] K. V. Umamaheswara Reddy and Venu Govindaraju. Form classification. In *SPIE Document recognition and Retreival*, volume 6815, pages 1–6, 2008.

[94] Christian Wallraven, Barbara Caputo, and Arnulf Graf. Recognition with local features: the kernel recipe. In *IEEE International Conference on Computer Vision*, pages 257–264, 2003.

[95] Pedro Quelhas, Florent Monay, J-M Odobez, Daniel Gatica-Perez, Tinne Tuytelaars, and Luc Van Gool. Modeling scenes with local descriptors and latent aspects. In *International Conference on Computer Vision*, volume 1, pages 883–890, 2005.

[96] E. Barbu, P. Héroux, S. Adam, and É. Trupin. Using bags of symbols for automatic indexing of graphical document image databases. *Ten Years Review and Future Perspectives: Graphics Recognition*, pages 195–205, 2006.

[97] Jianchao Yang, Kai Yu, Yihong Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1794 –1801, 2009.

[98] Yi Yang and S. Newsam. Spatial pyramid co-occurrence for image classification. In *International Conference on Computer Vision*, pages 1465 –1472, 2011.

[99] David G Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, volume 2, pages 1150–1157, 1999.

[100] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[101] D. Lewis and David D. Lewis Consulting. Building a test collection for complex document information processing. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 665–666, 2006.

[102] D. L. Dimmick, M. D. Garris, and C. L. Wilson. NIST structured forms reference set of binary images (sfrs). *NIST Special Database 2*, 1991.

[103] Andy Liaw and Matthew Wiener. Classification and regression by random forest. *R News*, 2(3):18–22, 2002.

[104] Peng Ye, Jayant Kumar, Le Kang, and David Doermann. Unsupervised feature learning framework for no-reference image quality assessment. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1098–1105, 2012.

[105] S. Marinai, E. Marino, and G. Soda. Layout based document image retrieval by means of xy tree reduction. In *International Conference on Document Analysis and Recognition*, pages 432–436, 2005.

[106] J. Kumar and D. Doermann. Unsupervised Classification of Structurally Similar Document Images. In *Intl. Conf. on Document Analysis and Recognition (ICDAR 13)*, pages 1257–1261, 2013.

[107] P.J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[108] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[109] Shuang Liang and Zhengxing Sun. Sketch retrieval and relevance feedback with biased svm classification. *Pattern Recognition Letters*, 29(12):1733–1741, September 2008.

[110] Simone Marinai, Beatrice Miotti, and Giovanni Soda. Mathematical symbol indexing using topologically ordered clusters of shape contexts. In *International Conference on Document Analysis and Recognition*, pages 1041–1045, 2009.

[111] A. Kesidis, E. Galiotou, B. Gatos, A. Lampropoulos, I. Pratikakis, I. Manolessou, and A. Ralli. Accessing the content of greek historical documents. In *Third Workshop on Analytics for Noisy Unstructured Text Data*, pages 55–62. ACM, 2009.

[112] M. Delalandre, J.M. Ogier, and J. Lladós. A fast CBIR system of old ornamental letter. *Recent Advances and New Opportunities in Graphics Recognition*, pages 135–144, 2008.

[113] Y. Leydier, F. Lebourgeois, and H. Emptoz. Text search for medieval manuscript images. *Pattern Recognition*, 40(12):3552–3567, 2007.

[114] G.X. Tan, C. Viard-Gaudin, and A.C. Kot. Information retrieval model for online handwritten script identification. In *International Conference on Document Analysis and Recognition*, pages 336–340, 2009.

[115] S. Uttama, P. Loonis, M. Delalandre, and J.M. Ogier. Segmentation and retrieval of ancient graphic documents. *Ten Years Review and Future Perspectives Graphics Recognition*, pages 88–98, 2006.

[116] Y. Fataicha, M. Cheriet, JY Nie, and CY Suen. Retrieving poorly degraded ocr documents. *International Journal on Document Analysis and Recognition*, 8(1):1–9, 2006.

[117] E. Saund. A graph lattice approach to maintaining dense collections of subgraphs as image features. In *International Conference on Document Analysis and Recognition*, pages 1069–1074, 2011.

[118] A.L.N. Fred and A.K. Jain. Data clustering using evidence accumulation. In *International Conference on Pattern Recognition*, volume 4, pages 276–280, 2002.

[119] A.J. Joshi, F. Porikli, and N. Papanikolopoulos. Multi-class active learning for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2372–2379, 2009.

[120] A. Biswas and D. Jacobs. Active image clustering: Seeking constraints from humans to complement algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2152–2159, 2012.