Computational Forensic

Sukalpa Chanda

Thesis submitted to Gjøvik University College

for the degree of Doctor of Philosophy in Computer Science

2015

A Computational Forensic Approach to the Analysis of Questioned Document Fragments

Faculty of Computer Science and Media Technology
Gjøvik University College

A Computational Forensic Approach to the Analysis of Questioned Document Fragments/Sukalpa Chanda
Doctoral Dissertations at Gjøvik University College 1-2015
ISBN: 978-82-8340-003-8
ISSN: 1893-1227

Declaration of Authorship

I, Sukalpa Chanda, hereby declare that this thesis and the work presented in it is entirely my own. Where I have consulted the work of others, this is always clearly stated.

Signed:

(Sukalpa Chanda)

Date:

Abstract

Fragments of documents are common subjects in forensic analysis of questioned documents. Forensic analysis of torn document is more challenging owing to sparse data content; for example, a document fragment might consist of only part of a word. The degree of difficulty increases when large number of such documents needs to be analyzed. A forensic expert might overlook evidences in this huge pool of data. This dissertation aims to help combat this problem by studying scientific methodologies that can narrow down the search space of a forensic expert. Automatic sorting of document fragments can be accomplished based on criteria set by the forensic expert. This demands execution of the following :(i) text/graphics segmentation;(ii) segmentation of text type (printed/handwritten);(iii) script identification of text;(iv) identification of the writer; (v) identifying the font of the printed text. Adopting various image processing and pattern recognition techniques certain methodologies are proposed for accomplishing such tasks. Rigorous experiments have been carried out to evaluate our scientific methodologies with real life torn document fragments. Feature encoding techniques have been meticulously chosen so that discriminative properties between different objects of interest are well represented, making the classification task easier. For e.g. in case of writer identification we have implemented a feature encoding scheme that reveals variations in character shape structures between different writers. The thesis consists of 10 chapters. A brief overview of every chapter is as follows:

- **Chapter 1** discusses the topic, and the challenges associated with it. This chapter also provides the motivation behind the research addressed in this thesis. In the beginning it briefs the problem and then provides a logical explanation about how the research aims to solve the problem of sorting torn document fragments. Later, it mentions about the contributions of this thesis followed by a brief description of all chapters in "Thesis Outline" Section.

- **Chapter 2** states various background theories that have been used as the basis of solutions proposed in this thesis. Our analysis revealed that sorting of similar torn document fragments can be accomplished by exploiting information on characteristics of its content type like script and font of printed text, writer of handwritten text, etc.. Background knowledge related to these topics are discussed in this chapter. We have extensively used Support Vector Machine (SVM) classifier in all experiments, hence a theoretical discussion on with SVM is also provided.

- **Chapter 3** presents the existing state-of-the-art methodologies on relevant sub- problems that we need to deal in order to accomplish our objective, for example we narrate here existing state-of-the-art methodologies on the following topics : (a) Text/graphics segmentation ; (b) Script identification ; (c) Writer identification ; (d) Font identification.

- **Chapter 4** provides a conclusion and direction towards future research.

- **Chapter 5** is based on an article "Document-Zone Classification in Torn Documents" and is devoted to the problem of text/graphics segmentation and text type discrimination i.e. printed and handwritten text identification in torn document fragments.

Sometimes torn document fragments might consist of text and graphics simultaneously. For reliable forensic analysis in torn document fragments, text/graphics segmentation and text type discrimination is required. This has been addressed in this article.

- **Chapters 6 and 7** are based on articles "Text Independent Writer Identification for Bengali Script" and "Text Independent Writer Identification for Oriya Script". Those chapters proposes solution to writer identification problem for two different Indic scripts (Bengali and Oriya) with limited amount of data per writer. Feature extraction methods were chosen in such a way that the unique structural properties of a writer's handwriting gets well represented. Features were extracted from each segmented character-component and were sent to a classifier for classification between different writers. Later, majority voting was performed amongst all classified character-components to decide the writer of a document page.

- **Chapters 8 and 9** are based on articles "Identification of Indic Scripts on Torn- Documents" and "Script Identification - A Han and Roman Script Perspective". Those two chapters investigates the issue of script identification in torn document fragments and in normal documents, respectively. Comparison between two different kind of feature types i.e. rotation-dependent and rotation-independent features for script identification in torn document fragments is performed. Advantage and disadvantage of both feature types are discussed. Torn document fragments with eleven different scripts are used in our experiments. To get an idea about script identification in normal (non-torn) documents, we also performed some experiments considering the following set of scripts - (Chinese, Japanese, Korean, and Roman).

- **Chapter 10** is based on the article "Font identification - In context of an Indic script" and explores the font identification problem in an Indic script (Bengali) perspective. Curvature based features are used to exploit tiny differences in shapes of characters from different fonts. Support Vector Machine and its Multiple Kernel variant are used for classification of the fonts of characters.

Acknowledgments

First of all, I would like to express my gratitude to my supervisors Prof.Katrin Franke and Prof.Umapada Pal for their valuable suggestions in order to move forward in my research work. I would also like to thank my third supervisor Prof.Slobodan Petroivic for his advice and constant mentoring to help me prepare the thesis draft.
I am thankful to my external collaborators Prof.Fumitaka Kimura and Associate Prof.Tetsushi Wakabayashi from Mie University in Japan for their support. I am also grateful to all other faculty members at GUC, from whom I took different courses as a part of my study program.
Finally, I would like to thank all my colleagues and friends at GUC for providing me a friendly ambience and a pleasant work place.

Contents

Chapter 1

Introduction

1.1 Topic

Documents found in a crime scene could provide us significant information about the crime and also about the people those are associated with the crime. Questioned document analysis is the process where various scientific techniques are applied by a forensic expert for examining such documents. Questioned document analysis helps us to reveal many hidden evidences about the committed crime. Nowadays forensics and criminal analysis are becoming extremely data intensive. In some situations, forensic experts need to deal with truckloads of paper documents from heterogeneous sources. In those situations, very often a forensic expert encounters many torn documents. The primary task of forensic expert is to find potential evidence from this huge pile of document fragments. From figure (1.1)(the figure adopted by author from [15]), we can easily visualize the degree of difficulty that a forensic expert needs to deal with while searching for evidences from huge number of document fragments. A forensic expert can find potential evidences by means of following strategies:

(a) Establish and estimate the timeline of creation of different concerned documents.

(b) Try to relate or find a link between two questioned documents, when the documents have apparently different origin or have been discovered from geographically apart places. This can be done by considering: whether the content types are alike in those two questioned documents?

(c) Determine the origin for concerned documents by resolving the issue of common authorship (in case of handwritten document images).

Reconstructing an original document from its torn fragments is a difficult and challenging task. A torn document fragment could be of arbitrary shape, size and may consist of random content. Difficulty increases with the number of fragments and handling such situation is a painful task for humans. To the best of our knowledge, until now most of the work on torn document reconstruction has focused on the shape of the torn fragments, in order to fuse torn pieces into a single document. They essentially try to solve the problem by taking the approach of solving a jigsaw puzzle. But it takes a very long time to accomplish this task as they need to search for the correct document fragment pieces from the pile of torn document fragments. In order to reduce the time and search space complexity for piecewise reconstruction, pre-selection or sorting is required. By means of this sorting we can select only those fragments which are similar to each other in some aspects, and restrict searching amongst those selected torn document fragments. The sorting criteria could be the content type (such as similarity between handwritten text, printed text etc.) of those torn document fragments. In this thesis, we intend to propose automatic forensic analysis tools to analyze torn document fragments based on content types. Some applications for such tools could be in the following area:

(a) To help forensic experts to identify similar documents in terms of source of origin.

(b) To keep an account of all office notes with respect to its writer, content type (like invoice, etc.).

(c) To make a digital repository of loose manuscript papers.

Figure 1.1: A forensic expert dealing with torn document fragments.

1.2 Research Problem

Our objective is to sort out similar document pieces of arbitrary shape, size and orientation from a pile of torn document fragments based on their content type. We have no prior information about the contents in those documents. They may contain handwritten text, diagrams, tables, printed text, or a mixture of all or some of them. We can sort out documents into certain groups based on the attributes given in table (1.1). From the table (1.1)

Table 1.1: Attributes that can be used for sortng similar document fragments.

Printed Text Attributes	Script Font Type Style (Italics/Bold)
Handwritten Text Attributes	Script Writer
Graphical content	Present or not?

it is clear that we can measure similarities between documents in terms of content type like:

(a) Are there any printed text? If so, then are the scripts similar? If so, then are the size and font used similar? etc.

(b) Are there any handwritten text? If so, are the scripts similar? If so, then are the text written by same people? etc.

(c) Are there any graphics in the document?

But there are certain problems in getting such attribute information. Firstly, all of our features should be capable of dealing with arbitrary orientation of data, since we cannot manipulate the image acquisition process in real life systems. Also, there is a high possibility of frequently encountering documents with a totally new attribute value. Hence, our forensic analysis tool should be a generic one, with no restrictions on the following:

- Script of the image text, (printed or handwritten);
- Writer of the text in case of handwritten text;
- Font of the text present in printed document fragments etc;

1.3 Motivation

Documents found in a crime scene might contain a pool of unseen evidence, that can be detected using a range of techniques and specialised equipments. Document Forensics areas analyse authenticity of questioned documents [70]. Using a variety of scientific processes, a document examiner answers questions in the context of the reliability and authenticity of a document. They might also answer queries regarding document timeline i.e. when a document was prepared, whether the document was altered by any means, and also help to recover information from erased or oblitereted parts in the document[70]. White collar crime is becoming more and more common in the corporate sector. Due to easy availability of sophisticated electronic equipments (e.g. photo copier, fax, printer etc.) a white collar crime scene sometimes involves disputed documents from different sources and on a large scale (think of a scenario where a forensic expert needs to analyse or compare handwritten notes from over several hundred employees, printed document pages generated from hundreds of printers, photo copied documents from a whole office building etc.,). Searching for evidences in this huge pile of documents is difficult. The aforementioned tasks are even more time consuming when the documents are torn. In such cases, computers can be utilised to sort similar document fragments into different groups. The notion of similarity could apply to any characteristics (like physical appearence or contents) of those document fragments. This could speed up the document examination process to aid the human forensic expert. The forensic expert can focus on one particluar group of document fragments, only those documents on which the forensic expert assumes to contain some reliable evidence of the committed white collar crime. Commercial softwares are available for the task of reconstructing shredded documents, the link[http://www.prlog.org/10077526-unshredder-new-shredded-document-reconstruction-software.html] shows an example of such software that is capable of dealing with shredded documents. But no such ready-made software is available for reconstructing torn document fragments. A real time system for reconstructing torn document fragments could be of immense use also in an economic crime scenario. For example, it could be used by a forensic expert to trace the origin of a threat note from a terrorist organisation. With the intention of removing the above mentioned bottlenecks in developing generic torn document analysis system, in this thesis we propose a computational intelligence-based forensic analysis system for questioned document fragments.

1.4 Related Research Problem and Our Approach to Solve the Problem

To accomplish our objective we need to address following research questions:

1. How can we develop a hierarchical/scalable approach capable of handling any document pieces for content-wise zone classification?
2. What kind of features will help to discriminate printed and handwritten text?
3. What kind of features will help to discriminate different scripts like Roman, Arabic, Devanagari, Bengali, Oriya, Urdu, etc.?
4. How can we handle text-independent writer identification problems with limited data from each writer?

Hence to solve the problem following solutions are required:

- Develop hierarchical/scalable approach capable of handling any document pieces for content-wise zone classification.
- Implement features which help us in segmenting type of document text (printed or handwritten text).
- Devise generic features to handle text independent writer identification problems even with limited data from a writer.
- Develop/Investigate a feature extraction method for font identification.

A broad schematic view of a generic torn document forensic analysis system is shown in figure (1.2). We need to execute various feature extraction and classification task for different objectives, for relaible forensic analysis of torn documents. For example initially we could try to identify the layout of the document or try to identify various regions/zone in the document (like where are text blocks, graphics etc). In the case of a text zone, we try to identify whether the text is machine printed or handwritten. For printed text we may carry out feature extraction and classification to identify the script of the text zone. Once we have identified script of a machine printed text zone, then we might try to recognize the particular font of that script. Consequently in case of a handwritten text zone, we may process the text to obtain segmented character-components. Considering those segmented character-components, we could perform necessary feature extraction and classification for possible writer identification or character recognition in documents. Please note that mentioned tasks cannot be easily performed in the context of torn document fragments. The reasons behind it are as follows:

(a) Document fragments contain a limited amount of information/data.

(b) Text orientation can be arbitrary.

(c) Document fragments are generally rather small with an average size of 4x5 cm.

Analysing the flowchart in figure (1.2), it is clear that segmenting the document fragment into different regions like graphics, text etc., is a crucial task for reliable forensic analysis. Feature extraction methodologies were meticulously chosen after logically analysing the objects to be differentiated. For example, to analyse/identify the content zone of torn documents, we used feature like Gabor filter known for their capability in performing texture analysis. Script identification on printed text has been accomplished involving all major scripts from around the world (Roman, 11 Indic scripts like Bengali,Oriya, Devnagari, Urdu, Malayalam,etc., and Han-based scripts) using features like Gradient, Directional chain-code histogram, Zernike moments, etc. Since a considerable amount of work on writer identification has been done for Roman script, we opted for writer identification on two Indic scripts:- Bengali (the fifth most popular script in the world) and Oriya (another popular Indic script). Keeping in mind the structural shape of characters, we have chosen a feature encoding scheme that best represents writer variability. For classification tasks involved in the experiment we mainly used the Support Vector Machine (SVM) classifier.

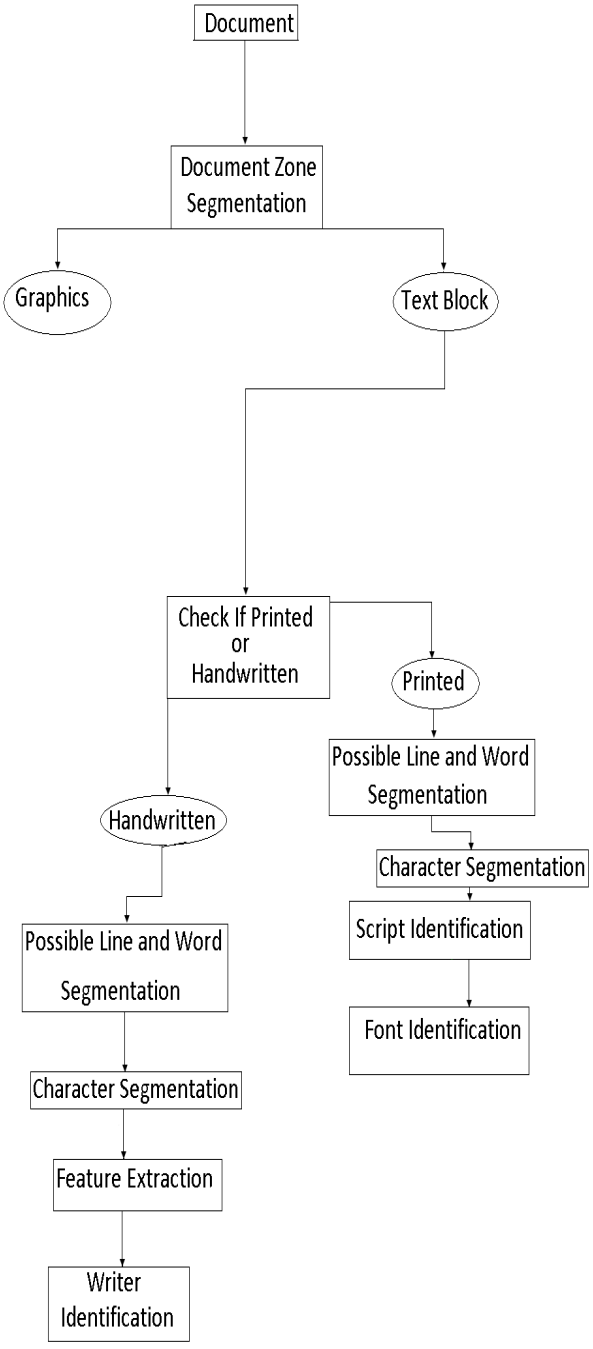

Figure 1.2: Basic flowchart of a generic torn document forensic analysis system.

1.5 Contribution of the Thesis

Primary contributions of this thesis are as follows:

- Text/non-text segmentation in the context of torn document fragment is accomplished.
- Writer identification problem for two different Indic scripts in a constrained environment (sparse data per writer) is addressed here.
- Script identification problem involving 11 different indic scripts in torn documents of arbitrary shape, orientation and size is being addressed here.
- Font identification in context of an Indic script (Bengali) is investigated and we obtained encouraging results with our method.

1.6 Thesis Outline

The thesis at hand is divided into two parts, the first part consist of 4 chapters and the second part consist of published research articles in form of chapters 5 to 10. The first four chapters narrates the motivation behind this research, it also discusses associated research challenges, proposed solution to the problem and relevant state-of-the art methodologies, followed by a conclusion and direction towards future research work. Chapter 5 to 10 constitutes only published articles in context to the challenges associated with our torn document forensic analysis problem. A brief discussion of the following chapters is as follows:

- **Chapter 2** states various background theories that have been used as the basis of solutions proposed in this thesis. Our analysis revealed that automatic forensic analysis of torn document fragments can be accomplished by exploiting information on characteristics of its content type like script and font of printed text, writer of handwritten text, etc,. Background knowledge related to these topics are discussed in this chapter. We have extensively used Support Vector Machine (SVM) classifier in all experiments, hence a theoretical discussion on SVM is also provided.
- **Chapter 3** presents the existing state-of-the-art methodologies on relevant sub- problems that we need to deal in order to accomplish our objective, for example we narrate here existing state-of-the-art methodologies on the following topics :- (a) Text-Graphics segmentation; (b) Script identification;(c) Writer identification; (d) Font identification.
- **Chapter 4** provides a conclusion and direction towards future research.
- **Chapter 5** is based on the article "Document-Zone Classification in Torn Documents" and is devoted to the problem of text/graphics segmentation and text type discrimination i.e. printed and handwritten text identification in torn document fragments. Sometimes torn document fragments might consists of text and graphics simultaneously. For reliable forensic analysis in torn document fragments, text/graphics segmentation and text type discrimination (printed or handwritten) is required. This has been addressed in this article.
- **Chapters 6 and 7** are based on articles "Text Independent Writer Identification for Bengali Script" and "Text Independent Writer Identification for Oriya Script". Those chapters proposes solution to writer identification problems for two different Indic scripts (Bengali and Oriya) with limited amount of data per writer. Feature extraction methods were chosen in such a way that the unique structural properties of a writer's handwriting gets well represented. Features were extracted from each segmented

character-components and were sent to a classiifer for classification among different writers. Later, majority voting was performed amongst all classified character-components to decide the writer of a document page.

- **Chapters 8 and 9** are based on articles "Identification of Indic Scripts on Torn- Documents" and "Script Identification - A Han and Roman Script Perspective", which investigates the issue of script identification in torn document fragments and in normal documents, respectively. Comparison between two different kind of feature types i.e. rotation-dependent and rotation-independent features for script identification in torn document fragments is performed. Merits and demerits of both feature types are discussed. Torn document fragments with eleven different scripts are used in our experiments. To get an idea about script identification in normal (non-torn) documents, we also performed some experiments considering the following set of scripts - (Chinese, Japanese, Korean, and Roman).
- **Chapter 10** is based on article "Font identification - In context of an Indic script" explores the font identification problem in an Indic script (Bengali) perspective. Curvature based features are used to exploit tiny differences in shapes of characters from different fonts. Support Vector Machine and its Multiple Kernel variant are used for classification of the character fonts.

Chapter 2

Background Information and Related Theories

A theoretical background knowledge is required to validate and support any scientific experiment. Since our experiments are predominantly based on a torn document's visual characteristics (content of the document), a brief discussion on these topics is essential. Moreover, our experimental task involves a lot of inference/decision-making, so a basis on the statistical learning theory is useful. Texture behaviour, script and font of a document as well as individuality of handwriting are characteristics for forensic analysis of torn documents and hence we briefly discuss about this at the beginning of the chapter. Next background of Statistical Learning Theory and Support Vector Machine (SVM) are discussed as we used SVM in our experiments.

2.1 Texture

Texture of any object can be visually recognized very easily but to define it formally is very difficult [136]. There are approximately 3000 different types of paper used alone in Europe as mentioned in [56]. Though the basic manufacturing process is the same, the difference in colour, surface texture, weight etc., is caused by the variations in physical and chemical composites used for manufacturing the paper. The main ingredients in commonly used papers are rag and/or processed wood fibers. Rag is used to increase the paper's robustness and for making paper documents like bank cheques. The length of the wood fibers greatly affects to the quality of the paper. The luminosity of the paper surface is due to variability in pigments such as caolin, calcium carbonate and talcum. Mentioned facts are discussed in [56] with the help of other published works. Also texture of a paper largely depends on those factors. This characteristic information can also be utilized for sorting torn document fragments.
Coggins [40] noted that various definitions on texture co-exist in the computer vision literature [136]. Examples of texture patterns are shown in figure (2.1). Though computer vision literature lacks a precise definition of texture, there are few intuitive properties of texture stated in [136] and are as follows:

- "Texture is a property of areas; the texture of a point is undefined. So, texture is a contextual property and its definition must involve grey values in a spatial neighborhood. The size of this neighborhood depends upon the texture type, or the size of the primitives defining the texture".
- "Texture involves the spatial distribution of grey levels. Thus, two-dimensional histograms or co-occurrence matrices are reasonable texture analysis tools".
- "Texture in an image can be perceived at different scales or levels of resolution. For example, consider the texture represented in a brick wall. At a coarse resolution, the texture is perceived as formed by the individual bricks in the wall; the interior details in the brick are lost. At a higher resolution, when only a few bricks are in the field of view, the perceived texture shows the details in the brick".
- "A region is perceived to have texture when the number of primitive objects in the region is large. If only a few primitive objects are present, then a group of countable

objects is perceived instead of a textured image. In other words, a texture is perceived when significant individual ”forms” are not present”.

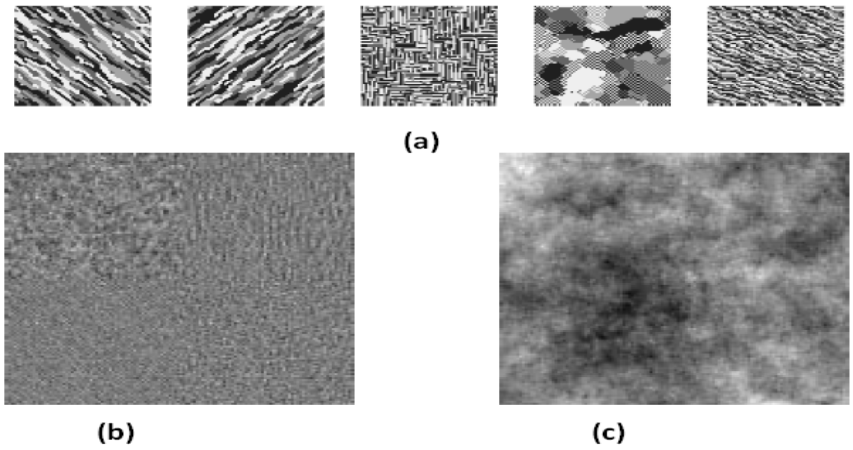

Figure 2.1: (a) Textures generated by discrete Markov random field models. (b) Four textures (in each of the four quadrants) generated by Gaussian Markov random field models. (c) Texture generated by fractal model.(Figure adopted by author from [136]. Copyright with World Scientific Publishing Co.

Texture-based features have huge application in the area of document image analysis. Document processing tasks like postal address recognition and interpretation of geographical maps depends heavily on texture analysis-based methods [136]. In OCR based postal automation applications the first step is to separate the region of interest in the image which contain useful information [136]. An example explains the fact in figure (2.2), (figure (2.2) is adopted by author from [136]). Here in the diagram it can be observed that the bar-code region have a distinct texture which got discriminated by Gabor-based texture analysis feature .

Figure 2.2: Locating bar code in a newspaper image. (a) A scanned image of a newspaper that contains a bar code. (b) The two-class segmentation using Gabor filter features. The bar code region in the image has a distinct texture.Figure adopted by author from [136]. Copyright with World Scientific Publishing Co.

2.2 Scripts and Fonts

Script is a graphical representation of a spoken language [33]. Spoken sentences of a language are expressed in written form using a script [6]. Script represents a well defined and systematic arrangement of different yet precise graphical shapes (characters). The origin of a writing system of a language as well as the forms and functions of the individual characters included in the script used for writing a language is studied under a discipline called Paleography [43]. Script types can be broadly divided according to their (i) geographical origin; (ii) Structural appearance ; (iii) Evolutionary journey from spoken words to written form. Scripts from the west (like Greek, Roman, Cyrillic etc.,) are known as occidental, whereas all Asian scripts are termed as oriental [50]. The latter can be further categorized into (i) Indic scripts; (ii) Han based scripts (Chinese, Japanese, Korean); (iii) Arabic and Persian (though they are closely related to one of the Indic script "Urdu"). A document content type can be very easily characterized by its script and font used for the text. Script can tell us a lot about the probable place of origin of a document.

2.2.1 Major Writing Systems in the World & its Relationship to Scripts

A taxonomy of scripts can be done based on its corresponding writing system [60].

Logographic System :- When a symbol graphically represents a complete word it is refered as a logogram. Scripts that use this form of writing system normally have thousands of characters. Han scripts (Chinese, Japanese, Korean) are perfect examples of scripts that are formed using the logographic system of writing.

Alphabetic System :- When a set of characters represent phonemes of a spoken language, they are termed as Alphabet. Scripts like Greek, Latin, Cyrillic, and Armenian belongs to this group. The Latin script, also called Roman script, is used by many languages throughout the world with little/huge modifications from one language to another.

Abjad System :- The Abjad system has symbols for consonantal sounds only, otherwise they are similar to alphabetic systems. One more difference in comparison to alphabetic systems also exists, Abjads are written from right to left within a textline. See figure 2.3 for an illustration.

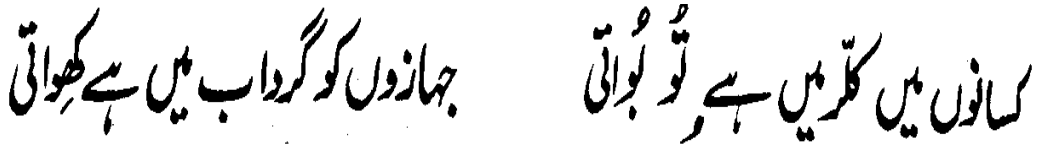

Figure 2.3: Example of an Abjad script text.

Abugida System :- Scripts of the Brahmic family are written using Abugida, which is another alphabetic-like writing system used in many scripts of India and south-east Asia. See figure 2.4 for an illustration.

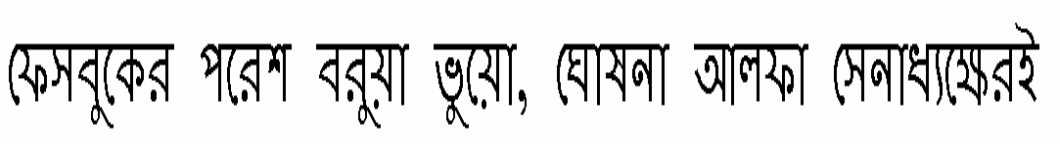

Figure 2.4: Examples of an Abugida script text.

2.2.2 Evolution of Font

In typography, a font is traditionally defined as a quantity of sorts composing a complete character set of a single size and style of a particular typeface [44]. The type face determines the overall design of the character shapes [5]. The style refers to the average stroke width of characters, boldface vs. lightface (normal), and the posture of the body, italic vs. Roman, The size of a font is typically given in points (1 inch = 72 points), or in Picas (1 inch =6 pica). For example, the set of all characters for 9-point Times New Roman italics is a font. Similarly, the 10-point Times New Roman italics would be a separate font as the size is different here. Further details on script and font can also be found in a chapter "Language, Script and Font Recognition" from a chapter in the book [45].
Typography can be considered as a special form of art. As art in a society has changed with time, different typography styles have also evolved over time [46]. Though the design or appearance of letters had already been evolving for many centuries, fonts, and typefaces appeared in the 15th century [46]. Western typography got largely influenced by the design of inscriptional capitals those were sculpted on Roman buildings. Western typography artists developed traditional typefaces those exhibit structurally appropriate design, angled stresses, contrasting strokes, and serifs [46]. The first typefaces used for printing the first book in Europe in 1455 had distinct gothic blackletter traits. The Bensch Gothic font is visually similar to those first ancestors of Western typography [46].
Typefaces like Bastarda, fraktur, rotunda, and Schwabacher were presented in later years in Italy, and they exhibit traits like distinctive professional design, structurized and highly organized glyphs [46]. Some computer fonts that closely mimic those medieval typefaces are Stonehenge Regular, Breitkopf Fraktur, Typographer Rotunda, and Schwabacher. During the 'Renaissance' period a new style of writing, known as "cursiva humanistica" appeared. A wide range of Italic or cursive typefaces evolved due to those slanted glyphs design [46]. Typefaces known as "Canon de Garamond" and "Petit Canon de Garamond" was developed by Claude Garamond and those typefaces served as the prototype for modern day Garamond style fonts, such as Apple Garamond, SGaramond Regular etc.,[46].
Due to the invention of lithography in 1796, typography was used on wide range of applications from newspapers to posters and advertisements [46]. With the advent of personal computers at present a new font or typeface can be designed in a relatively easy way with the help of specialized computer applications or font editors.

2.3 Writer Individuality

Since pre-historic age human beings were engaged in writing on cave walls. Though we cannot write unless we have been taught, it is not a natural instinct of humans [14]. The generation of handwritting movements is a complex interaction between brain and spinal chord [56]. For each writing system, the formation of symbols/characters have a definite and ideal movement. The spatial relationships between symbols and the directional conventions on the page needs to be communicated between generations [14]. Handwriting is a very complex skill in which linguistic, cognitive, perceptual and motor components need to be coordinated into an integrated fashion [14]. Handwritings have a physiological/psychological link with the brain. Research activities of five people namely Robert Saudek, (England, psychologist/graphologist), Dr. Rudolph Pophal (Germany, neurologist/graphologist), Klara Roman (Hungary, psychologist/graphologist), Dr. Werner Wolff (Germany/America, psychologist), and Dr.Alexander Luria (Russia/physiologist) established the fact about persons handwriting's physiological /psychological link to the brain [17].
During childhood, every kid more or less posses similar handwriting characteristics. But with the passing of time, those writing characteristics along with their own style characteristics forms their own unique handwriting [13]. Two or more people might sometimes share a couple of individual characteristics, but the chance of those people sharing 20 or

30 individual characteristics is very unlikely [13]. There exist two fundamental factors that can be attributed to this individuality of handwriting [28]. They are genetic and memtic (cultural) factors. The factors are as follows:

a The relative sizes of the carpal bones of wrist and fingers and their impact on pen grip[28].

b The left or right handness[32].

c Muscular strength, fatigability, peripheral motor disorder.

d Central nervous system properties.

An excellent paper which dealt with biomechanical handwriting characteristics is due to Franke [57], where it is shown that the signing behavior of genuine writers and impostors is only likely to differ in terms of local characteristics.
The history of handwriting analysis could be traced back to Confucius, who stated: "Beware of the man whose writing sways like a reed in the wind." An Italian physician named Camillo Baldi published a method in 1622 to recognize the nature and quality of a writer from his handwritten letters. This is considered as the first extensive work on handwriting analysis. Baldi proposed the fundamental premise that continues to guide handwriting analysis even today: "It is obvious that all persons write in their own peculiar way . . . Characteristic forms . . . cannot be truly imitated by anybody else" [19]. This particular phenomenon has been utilized by forensic handwriting experts for quite a long time. But with the demands of time, handwriting experts today are aided by computer programs which actually can identify an individual on the basis of his handwriting. These computer programs are based on the strong theoretical and scientific platform of Statistical Learning Theory. Such computer programs evolved in early 90's and today they officially take part in questioned document analysis process in various forensic departments across the globe. During forensic analysis of handwriting, handwriting analysts try to maintain a strict protocol with criminal suspects [19]. In the analysis process the suspect could never see the questioned document. During the procedure, the suspect is devoid of any information on how to write certain words. The suspect is asked to write texts that are present in the questioned document, certainly without showing him the questioned document. Then the spelling and handwriting of certain words and phrases can be compared between the text writen by the suspect and the questioned document. The text needs to be written by the suspect at least three times in presence of an official witness [19]. Handwriting analysis is one of the most extensively used forensic analysis technique. Though in one particular prosecution "United States v. Saelee (2001)", the court noted that forensic handwriting analysis techniques in use lacks in reliability [18][16]. The court raised the issue that most basic principles of handwriting analysis "everyone's handwriting is unique"' had never been demonstrated. The technique used in forensic analysis of handwriting appears to be entirely subjective and without any controlling standard.
Though a study by Prof. Sargur Srihari (from State University of New York at Buffalo) for scientific validation of handwriting analysis could provide us some points in favour of forensic handwriting analysis using computers. Prof. Srihari subjected 1,500 writing samples to computer analysis[18] and noted that in 96 percent of cases, the writer of a sample could be positively identified based on quantitative features of his handwriting such as letter dimensions and pen pressure [18].

2.4 Statistical Learning Theory

Statistical learning theory provides an automatic methodology for gaining knowledge and making consequent predictions/decisions from a set of data. This is studied in a statistical

framework, where there are some assumptions of a statistical nature about the way the data is generated. The process of inductive inference is explored/analysed in statistical learning theory using the following steps [22]:

- Observe a phenomenon.
- Construct a model/hypothesis of that phenomenon (gainning Knowledge).
- Make predictions using this model.

'Machine Learning' aims towards automating this process whereas 'Learning Theory' formalizes it [22]. In this process, the learning algorithm is provided with many data samples (every data sample consist of a particular label generally termed as "class label" and an associated description of the data sample in the form of a feature vector). After the model is developed during the training phase, the task of the learning algorithm is to construct a function to map unseen data samples to correct labels. This function should be such that while predicting the label of previously unseen samples (test samples) it makes few mistakes. Certainly it is always possible to build a function that represents exactly the data (training data) in a desired form. But unfortunately such a technique will develop a hypothesis / learning model which will exhibit poor performance on unseen examples(this situation is termed as over-fitting). Learning algorithms search regularities in the training data, that can be generalized from the observed past to the future [139]. Amongst many hypothesis/learning model we normally try to select a learning model which best represents our problem/data and is simultaneously simple. But unfortunately there are no ready methods to measure the efficacy and quantify simplicity of a model. The superiority of one model over another cannot be asserted universally; rather it depends on problem/data. This phenomenon is termed as "No Free Lunch" theorem.
Issues like "generalization" and consistency in machine learning perspective are worth mentioning here. In naive words "generalization" means the ability of the hypothesis / learning algorithm to perform equally well with both training and testing data. To understand "generalization" in terms of machine learning one needs to understand *expected risk* and *empirical risk*. Let us assume that we are dealing with a classification problem using a classifier consisting of adjustable parameters λ. Our training set is $(X_1, Y_1), ..., (X_n, Y_n)$, here X is the feature vector for a sample in the training dataset and Y is the associated class label. The classifier will tune its parameters λ to learn the mapping $x \rightarrow y$. The expectation of test error can be used to measure the performance of this classifier as shown in equation below [82].

$$R(\lambda) = \int E(y, (x, \lambda))P(x, y) \tag{2.1}$$

This is termed as expected risk. But in practice one must settle for the empirical risk measure which is defined as follows:

$$R_{emp}(\lambda) = \frac{1}{n}\sum_{i=1}^{n} E(y, (x, \lambda)) \tag{2.2}$$

The *empirical risk* as shown in equation (2.2) is just a measure of the mean error over the available training data [82].

In general it is expected that if a classifier f_n's performance is evaluated on its training set, the empirical risk $R_{emp}(f_n)$ should be relatively small otherwise we can state that the learning algorithm is not capable of explaining even the training data. In real life if the difference $|R(f_n) - R_{emp}(f_n)|$ is small then we say that a classifier f_n generalizes well[139]. From this definition it can be asserted that, good generalization performance does not indicate that a classifier has a small overall error $R(f_n)$. It only tells us that the empirical error

$R_{emp}(f_n)$ is a good estimator of the true error $R(f_n)$. The most undesirable real life scenario is the situation where $R_{emp}(f_n)$ is much smaller than $R(f_n)$. Another aspect in the design of learning algorithm is to maintain trade-off between overfitted and underfitted learning model. In the terminology of applied machine learning, a complex learning model tends to exhibit over-fitting, while an overly simple learning model design would lead to under-fitting [139]. Best performance can be achieved by balancing a trade-off between these two factors which is depicted in the figure (2.5).

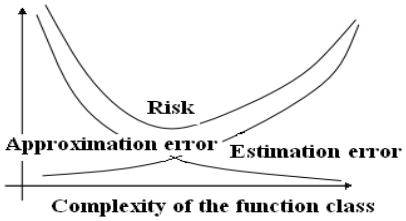

Figure 2.5: The trade-off between estimation and approximation error. If the function space used by the algorithm has a small complexity, then the estimation error is small, but the approximation error is large (underfitting). If the complexity of is large, then the estimation error is large, while the approximation error is small (overfitting). The best overall risk is achieved for "moderate" complexity. Figure (2.5) is customized by author from [139] with consent from the first author. Copyright with original publisher.

"Consistency" is a concept which is closely related to generalization [139]. But, unlike generalization consistency is a property not of an individual function, but a set of functions. The notion of consistency aims to make a statement about what happens in the limit of infinitely many sample points [139]. It states that a learning algorithm, when encountered with more and more training examples, should eventually "converge" to an optimal solution [139].

We have used Support Vector Machine (SVM) for classification purpose in all our experiments. SVM is originally a binary linear classifier which looks for a maximum margin between two different class examples. A kernel SVM takes care of the non-linear classification problem. Since we found SVM in general to outperform other classifiers for the classification task involved in our problem, it is worth taking a brief insight at SVM.
Support Vector Machine-Today Support Vector Machine(SVM) acts as a strong pillar of Statistical Learning Theory. The key idea behind SVM is to find a separating hyperplane that formes the maximum margin between examples of two different classes. But in such a binary linear classification problem, it might be possible to have a separable hyperplane in multiple ways, this situation is depicted in figure (2.6).

It can be easily noted that many possible hyper-planes can be drawn which can separate the data. But only one hyperplane amongst all gives us maximum distance between two class examples. If we use any hyperplane to classify, it might end up closer to one set of datasets compared to others, and this is not desirable [74]. Hence we can assert that the concept of maximum margin classifier or hyperplane is an optimal solution. The next figure (2.7)gives the idea of maximum margin classifier.

Figure (2.7) gives an illustration of a linear classifier with the maximum boundary. The objective of linear SVM is to find a hyper-plane that will separate a given set of data points, this can be extended to non-linear boundaries using kernel trick [117]. According to Statistical Learning Theory maximum generalization can be achieved by maximum margin. Now we can express a linear SVM mathematically. In order to mathematically calculate the maximum margin we need to implicitly fix a scale. We introduce canonical hyper plane for both classes with following notation:(a) x is our data (Feature Vector); (b) b is a bias;

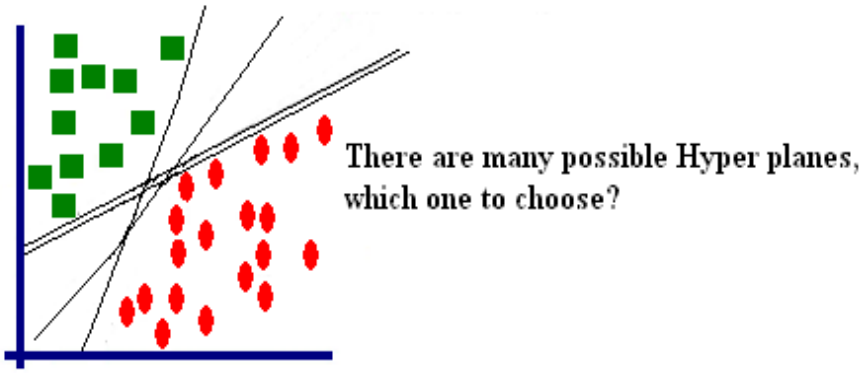

Figure 2.6: Please note there are many hyper-planes which can be fit in to classify the data but we should seek for the best hyperplane. This acts as the primary motivation behind SVM. Picture customized by author from slides by Andrew W. Moore [95].

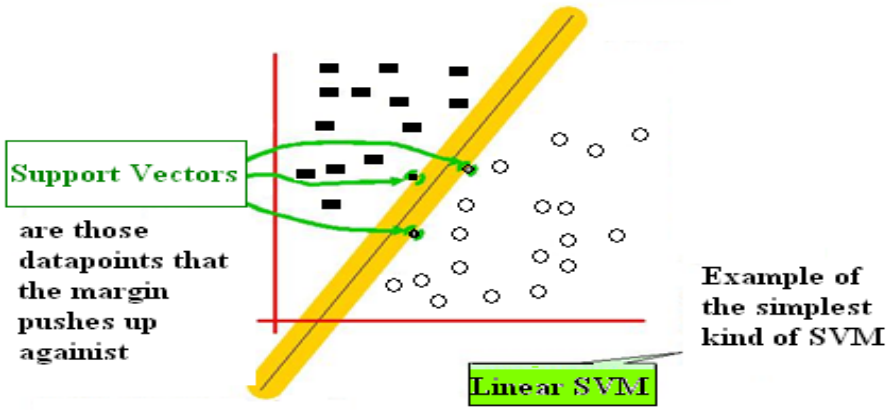

Figure 2.7: Illustration of max. margin. Picture customized by author from slides by Andrew W. Moore [95].

(c) w is the weight vector, where w is normal to the hyperplane. Here we set the canonical hyperplane in such a way that all positive examples lying on it should get evaluated to a score of exactly +1, i.e $x_{+1i} \cdot w + b = +1$, similarly for all negative samples lying on the canonical hyperplane on the opposite side, should get evaluated to a score of exactly -1,i.e $x_{-1i} \cdot w + b = -1$.

Let us consider two arbitrary points on both class example. The distance between them is given by $X_1 - X_2$, (red line) in figure (2.9). The margin/distance between X_1 and X_2 can be obtained by projecting it on the vector normal to the hyperplane, (green line) in figure (2.9)

For each class example, distance =1 between margin and the sample (see dotted black line in figure (2.10). We do this to implicitly fix a scale, i.e. we define a metric. On subtracting $x_{-1i} \cdot w + b = -1$ from $x_{+1i} \cdot w + b = +1$ we can see that the projection of the vector w on the vector $x_{+1i} - x_{-1i}$ can be written as $w \cdot (x_{+1i} - x_{-1i}) = 2$. Here canonical hyperplanes

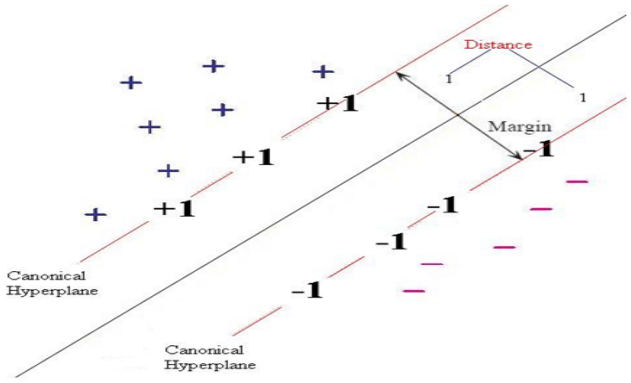

Figure 2.8: Setting the canonical hyperplane. Picture customized by author from the source [31] with formal consent.

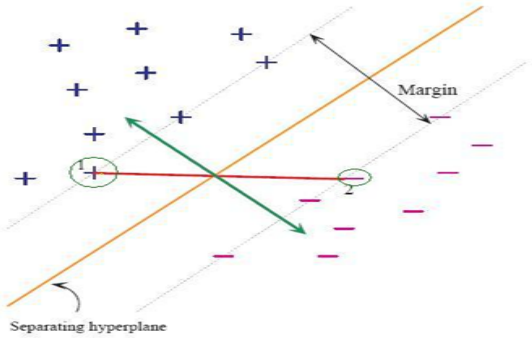

Figure 2.9: Deriving the margin between samples of two different class example. Picture customized by author from the source [31] with formal consent.

are in yellow line. Data points that lie on yellow lines are known as Support Vectors.

It turns out that this maximum margin problem can be expressed as a constrained optimization problem, its primal formulation is as follows:

$$\begin{aligned} &\max \frac{1}{\|w\|} \\ &\text{subject to} \qquad y_i(x_i \cdot w + b) - 1 \geq 0 \forall i, i = 1, \ldots N \end{aligned} \tag{2.3}$$

Since the equation above holds strong duality we can formulate it to a minimization problem as follows:

$$\begin{aligned} &\min \frac{1}{2}\|w\|^2 \\ &\text{subject to} \qquad y_i(x_i \cdot w + b) - 1 \geq 0 \forall i, i = 1, \ldots N \end{aligned} \tag{2.4}$$

Putting the constraint back in the objective function the corresponding Lagrangian can be formed which is as follows:

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_i \alpha_i [y_i(x_i \cdot w + b) - 1 \geq 0] \tag{2.5}$$

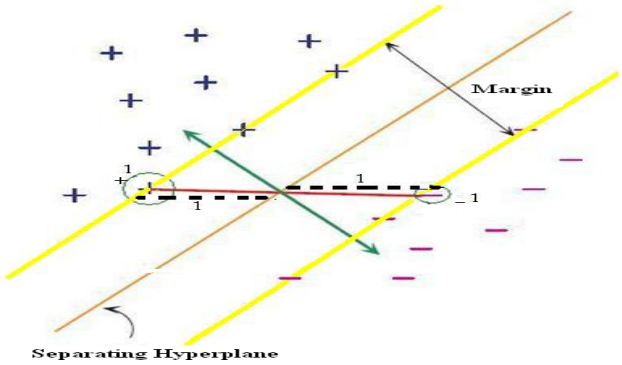

Figure 2.10: Deriving the margin between samples of two different class example. Picture customized by author from the source [31] with formal consent.

We need to solve the equation above for the primal variables of the Lagrangian w and b. We need to calculate the partial derivative of the Langrangian with respect to the primal variables w and b and setting each of them to 0. Now $\|w\|^2$ is nothing but $\sum_i {w_i}^2$.
$\therefore$

$$\begin{aligned}\frac{\partial L}{\partial w} \Rightarrow 1/2\frac{\partial \sum_i {w_i}^2}{\partial w} - \frac{\partial \sum \alpha_i y_i x_i \cdot w}{\partial w} = 0 \\ \Rightarrow w - \sum \alpha_i y_i x_i = 0, \\ \Rightarrow w = \sum \alpha_i y_i x_i \\ \text{Since} \\ \frac{\partial \sum_i {w_i}^2}{\partial w} \Rightarrow [\frac{\partial w_1^2 + \ldots w_n^2}{\partial w_1}, \ldots, \frac{\partial w_1^2 + \ldots w_n^2}{\partial w_n}] \\ \Rightarrow 2[w_1, w_2, \ldots w_n] \\ \Rightarrow 2w\end{aligned} \tag{2.6}$$

Similarly,

$$\frac{\partial L}{\partial b} \Rightarrow \frac{\partial \sum \alpha_i y_i b}{\partial b} = 0 \Rightarrow \sum \alpha_i y_i = 0 \tag{2.7}$$

Putting the value of the primal variables in the lagrangian function we get corresponding dual of our primal problem which is as follows:

$$\begin{aligned}&\max_{\alpha} && \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \\ &\text{Such that,} && \sum_i \alpha_i y_i = 0 \\ &\alpha_i \geq 0, && \forall i = 1, \ldots n\end{aligned} \tag{2.8}$$

Linear SVM works well when the data is linearly separable. Often in real life we need to deal with datasets those are non-linear in nature. In such situations kernel tricks are used to map/project the input data to a high-dimensional space where the data becomes linearly separable. The kernel trick allows SVM's to perform this task. The kernel trick can be applied to any algorithm that solely depends on the dot product between two vectors. Wherever a dot product is used, it is replaced by a kernel function [20]. The Kernel function maps the attributes of the input space to the feature space. A schematic diagram is shown

in figure (2.11). Kernel methods offer a modular framework, which operates in two steps [41] and are as follows:

- "In the first step, a dataset is processed into a kernel matrix. Data can be of heterogeneous types".
- "In the second step, a variety of kernel algorithms can be used to analyze the data, using only the information contained in the kernel matrix".

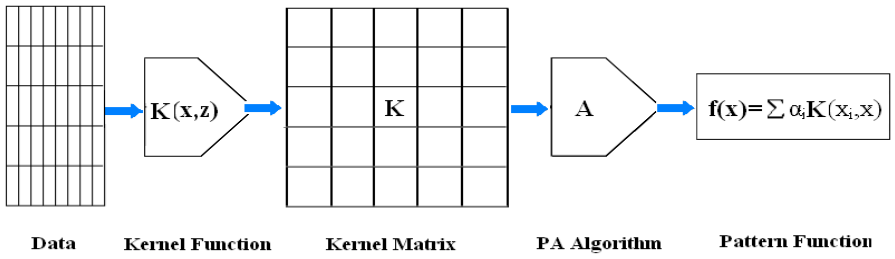

Figure 2.11: Schematic Diagram of Kernel Methods. Picture customized by author from [42].

It serves two purposes [42].

- Embedding data in a vector space.
- Looking for (linear) relations in such space. If the map is chosen suitably, complex relations can be simplified, and easily detected. Please refer to the figure (2.12).

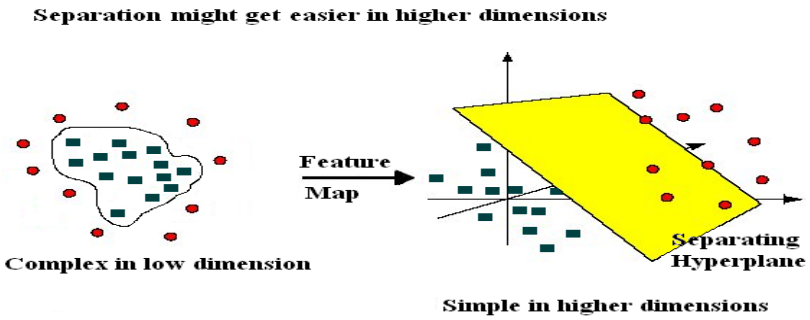

Figure 2.12: Mapping Data to High Dimensional Feature Space. Picture customized by author from [51].

Performance of non-linear SVM largely depends on the kernel function and its associated parameter values. Kernel functions are continuous, and symmetric in nature. The Gram Matrix of the Kernel function must be positive definite in nature. By positive definiteness we mean that the kernel matrices will always have positive Eigen values. Due to this positive definite property of the kernel matrix, the optimization problem involved with the SVM formulation will be convex in nature and the solution will be globally unique. Formally any kernel matrix must satisfy Mercer's theorem,which is as follows: Every (semi)

positive definite, symmetric function is a kernel: i.e. there exists a mapping Φ such that it is possible to write: $K(x_i, x_j) =< \Phi(x_i), \Phi(x_j) >$ i.e mathematically we can write,

$$\iint K(x,y)f(x)f(y)\,\mathrm{d}x\,\mathrm{d}y \geq 0 \forall f \tag{2.9}$$

A kernel, in this context, is a symmetric continuous function.

Introducing kernel functions to linear SVM changes the dual formulation of SVM, which just replaces the dot product of vectors x_i and x_j with their corresponding kernel matrix entry. The new dual for non-linear kernel SVM is as follows:

$$\begin{aligned} &\max_{\alpha} && \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j k(x_i, x_j) \\ &\text{Such that} && \sum_i \alpha_i y_i = 0 \\ &\alpha_i \geq 0, && \forall i = 1, \ldots n \end{aligned} \tag{2.10}$$

Some kernel functions which are not strictly positive definite sometimes performed well [20]. An example is the Sigmoid kernel, which looses its positive semi-definiteness characteristics for certain values of its parameters. Boughorbel et. al.[21] by means of experiments proved that conditionally positive definite kernels may outperform most classical kernels in some applications[20].

It is worthy mentioning that the efficacy of a learning model also largely depends on the features (characteristics/attributes) of objects that are used for learning/training of the model. This gives rise to the notion of the "ugly duckling" theory. The message from this theorem states that - In the absence of assumption there is no "best" feature representation. The theorem was proposed by a Japanese theoretical physicist "Professor Satosi Watanabe", who also studied pattern recognition and cognitive science out of his interest. The Ugly Duckling theorem is an argument which states that classification is impossible without some sort of bias [140]. It is named after Hans Christian Andersen's famous story of "The Ugly Duckling." It gets its name because it shows that, all things being equal, an ugly duckling is just as similar to a swan as two swans are to each other. So what we need to emphasize is that we should use correct attribute to represent objects to a learning model, which will make classification of unseen objects easier.

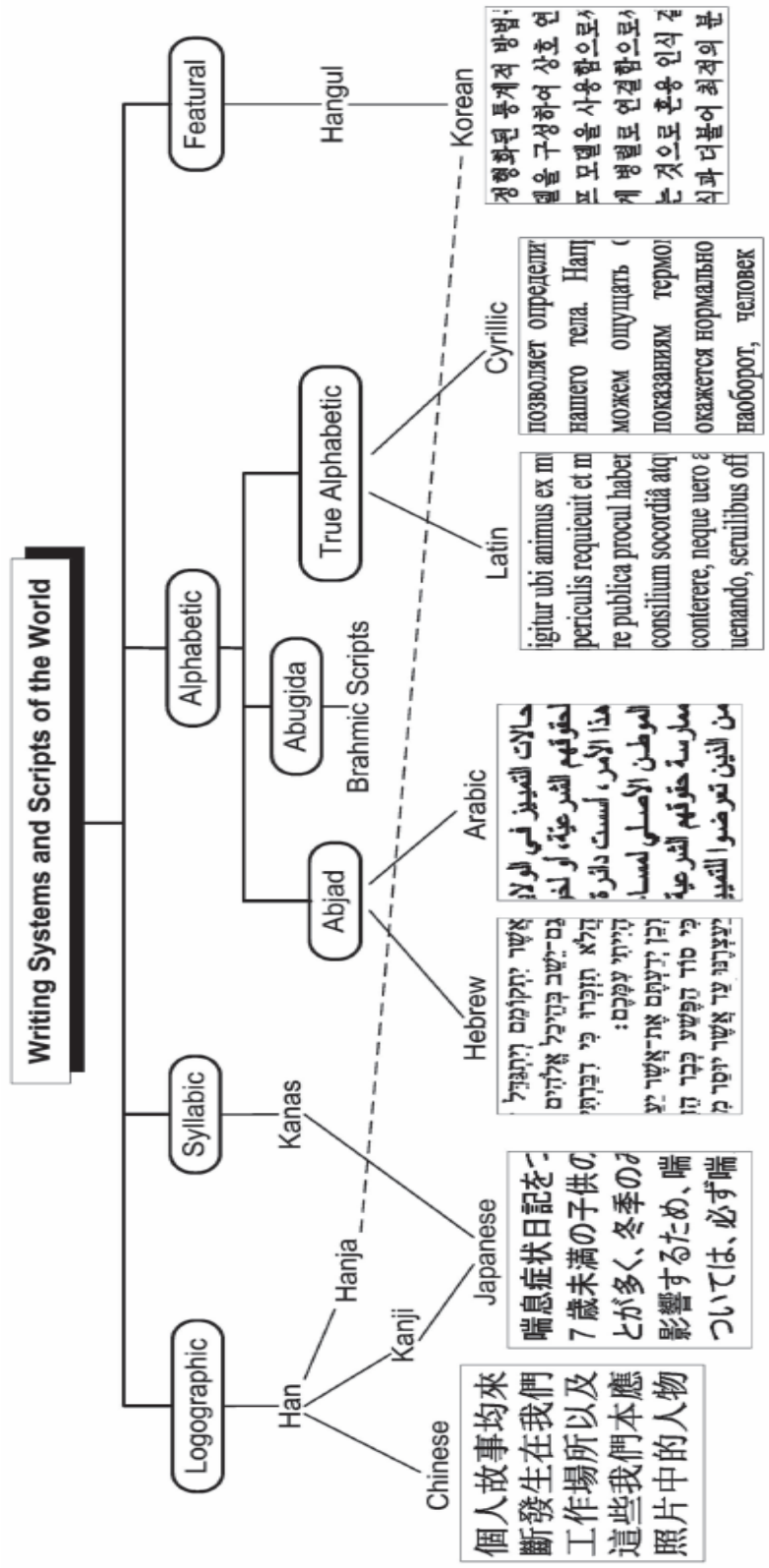

Figure 2.13: Writing system and script relationship. Figure (2.13) adopted by author from [60]. Copyright with IEEE.

Chapter 3

Literature Review

3.1 Introduction

We have asserted earlier that the problem of automatic sorting of document fragments needs to deal with several sub-problems like "TEXT and GRAPHICS SEGMENTATION", "SCRIPT IDENTIFICATION", "WRITER IDENTIFICATION", "FONT IDENTIFICATION", hence a detail on the present state-of-the-art methods on this topics are worth mentioning.

3.2 Text and Graphics Segmentation

Earlier research shows that texture analysis is of great importance for the problem of text/graphics segmentation. Even simple Co-occurrence matrices-based features are quite effective for discriminating different textures. Haralick et al. [64] proposed some textural features based on grey tone spatial dependencies, and used those features for identification task of three different kinds of image data. Data sets were divided into training sets and test sets. In their experiments, identification accuracy is 89% for the photomicrographs, 82% for the aerial photographic imagery, and 83% for the satellite imagery. An algorithm for calculating parameters of co-occurrence matrices is proposed by Argenti et al. [53]. It has been applied for classification and segmentation of artificial and natural scenes, the proposed method uses supervised learning and maximum likelihood estimates for the classification task. Lettner et al. [86] proposed a method which utilises the whole spectral feature space for foreground-background separation in multi-spectral images of ancient documents. This method was based on a Markov Random Field (MRF) model. Higher order MRF helped in separating the character from the background more efficiently. Jain et al. [73] presented a simple method for document image segmentation in which text regions in a given document image are identified. It uses Gabor filters to perform the segmentation task. This method uses grey level image and no thresholding operation is required on the input image. The method works well at lower resolution also.
Using a set of spatial characteristics of text/graphics component segmentation is proposed by Fletcher et al. [55]. Using transformation or vectorisation, some researchers have tried to extract graphical objects from text [63]. An et al. [4] reported an algorithm for segmenting photographs, handwritten text, printed text, blank spaces etc., using a multi-stage post classifier-based approach. Raveaux et al. [109] proposed a method that takes advantage of colour properties, by computing a relevant hybrid colour model. They construct a binary image composed of contour information using an edge detection step. Later, connected components in the contour image are classified according to a graph representation. A prototype selection scheme consideres text and graphics diversity for structural data.

A model-based trainable approach for high volume page segmentation applications is introduced by Shafait et al. [123]. The method is able to train given models on a small training set without labelled page segmentation. Instead of trying to model generic page layouts, the approach of style-directed layout analysis is used because this closely resembles the document generation process, hence it can obtain better performance on a specific class of documents. Then, a probabilistic matching algorithm is used to find the most likely layout of a page, given its layout model. Finally, using an Expectation-Maximization

learning algorithm it learns geometric variability of model components from training data without the need for page segmentation ground-truth. Earlier Keysers et al. [79] proposed a scheme for block classification, three features that gave encouraging results in context of content-based image retrieval (CBIR) were compared. An error rate of less than 1.5 % was achieved. Shafait et al. [121] introduced a new representation and evaluation procedure of page segmentation. The method permits analysis of the behaviour of page segmentation algorithms, under segmentation at different layout labels.

Shafait et al. [122] proposed a methodology where using a canonical representation of ground truth data it guarantees pixel-accurate evaluation results for arbitrary region shapes. Analyzing the results obtained after evaluating widely used segmentation algorithms on the UW-III database, it is evident that the new evaluation scheme is capable in indentifying several specific flaws in individual segmentation methods. Li et al. [87] proposed an approach of Text-line segmentation in freestyle hand written documents. From an input image document, it first estimates the probability map, each element in the map represents the probability of the underlying pixel belonging to a text line. Then a level set method is used to determine the boundary of neighbouring text lines. The method does not use any script-specific knowledge. This approach combines the advantages of both the bottom-up and the top-down approaches. Mao and Kanungo [91] described the software architecture of the PSET evaluation package to help the researchers to analyse their page segmentation algorithm.

Pal et al. [99] proposed a method that deals with machine-printed and hand-written text in Devangari and Bangla script. This scheme, based on structural and statistical features of machine-printed and hand-written text, gave an accuracy of about 98.6 %. Kuhnke et al. [83] designed a classification system which reads a raster image of a character and outputs two confidence values, one for a machine-written and one for a hand-written character class, respectively. The proposed system features a pre-processing step, which transforms a general uncentred character image into a normalised form, then the feature extraction phase extracts relevant information from the image, later a feed-forward neural network performs the final classification. It gave a recognition rate of 96.8 % on the training set and 78.5 % on the test set.

A rule-based system was developed by Fisher et al. [54] with no prior knowledge about the document structure. The proposed article there have modified and integrated portions of selected published image segmentation algorithms such that their resulting system is adaptive, so that necessary parameters are dynamically determined for each document image under consideration. Saitoh et al. [113] proposed a system for image segmentation and identifying text area in a document. There a tree graph text area layout is made for ordering the text areas, which were later used for conversion to a structured document format.

Patricio et al. [105] proposed a robust technique for segmenting text and graphics; it was independent of fonts and type of document. They [105] used Fourier descriptors as features. The magnitude spectrum of the grey level histogram of an image window is being used as the initial set of discriminant features. For classification of text and graphics/images a three-layer perceptron neural network, trained with the back propagation rule was applied, giving excellent results with 99% success.

A method using clustering methods for the purpose of text/graphics separation in coloured map was proposed by Roy et al. [110]. They followed pyramid segmentation for grouping isolated characters into words. A robust technique is proposed by Chowdhury et al. [38] for segmenting all sorts of graphics and texts from document pages. The main thrust was given on the segmentation of the graphics as well as text from a document page, which is already half-tone segmented, and may not be even fully skew corrected. The claimed novelty of this work is in the detection of the graphics drawn entirely with dots or

dashed lines.

Another text extraction method from graphical document images was proposed in [68] using a sparse representation framework. In their proposed method, wavelet transform and curvelet transform are used to represent text and graphics components. Morphological Component Analysis is used for the promotion of sparse representation of text and graphics components. The method has a high recall rate in favor of text components.

Guo et al. [61] presented an Hidden Markov Model based algorithm to distinguish between machine printed and handwritten materials. In their proposed system classification was performed on the word level. Experimental results showed that the method achieved a success rate of 72.19% on fully extracted handwritten words and 90.37% on partially extracted handwritten words. Zheng et al. [144] reported on printed and handwritten text segmentation using K-NN, SVM and Fisher classifiers with features like pixel density, aspect ratio and Gabor features. The method proposed by Jang et al. [75] first performs valid connected component grouping, followed by feature extraction and classification. A set of features related to width and position of groups of valid connected components was used by a neural network. The experiment involves address images extracted from Korean mail. The correct classification rate for 3,147 testing images was about 98.9%. Kandan et al. [77] described two level classification algorithms to distinguish handwritten elements from printed text in a printed document. The whole process is divided into two stages. In the first stage, two classifiers were used and a comparison between the nearest neighbour classifier and Support Vector Machines (SVM) classifier to localise the handwritten text was done. The features that are extracted from the document are seven invariant central moments. Using these features, they classify the text as hand-written component. At the second stage, they use Delaunay triangulation-based technique to re-classify the misclassified elements. Bukhari et al. [24] introduced a new approach that can handle a high degree of variable curls. It achieved an accuracy of 97.96% implying high segmentation accuracy with curled textlines. Another recent endeavour is due to [106], where they have used a boosted tree classifier and obtained a recall of 98.74% for printed text and 93.67% recall for handwritten text.

3.3 Script Identification

Script is defined as the graphical form of the writing system, that are used to write statements expressible in language. We can assert that a script class refers to a particular style of writing and the set of characters used in it [60]. Different languages are written in many different scripts. A script may be associated with only one language or may be associated with many languages. For example, Some Indic languages like Sanskrit, Hindi, Konkani, Marathi, use Devnagari script; Using the Latin alphabet set a large number of languages like French, German, English etc., are written. It can also be noted that sometimes the same script might not be used for writing a language forever, it might change over time or geographical location. For example, once upon a time Malay language used to use Jawi script but now it uses the Latin alphabet [60]. Another such example is Sanskrit, which in India is written in Devnagari, whereas in Sri Lanka is written in Sinhala script [60]. The present state-of-the-art reflects that until now, script recognition techniques have been broadly divided into the following two types [60]:

(i) Structure based script identification method.

(ii) Visual appearance-based script identification method.

Using these two fundamental approaches, script identification has been performed at different levels e.g. (a)Page level; (b)Paragarph /Text line level; (c)Word/character level.

3.3.1 Structure-based Script Identification Method

Script identification between Han and Latin script is proposed by Spitz [129]. He used optical density distribution of characters and frequently occurring word shape characteristics as features. An automatic script identification technique at the document/page level has been described by Hochberg et al. [69]. In their proposed scheme, template symbols for a particular script class were generated by size-normalizing and clustering textual symbols obtained from particular script. While testing, the Hamming distance is used to compare all textual symbols extracted from the input document with the template symbols, and then scored against every script class. The script class with the highest average score is decided as the script of the document.

Among Indian scripts, there are some related works. More than a decade back a work on text-line wise script identification was proposed by Pal and Chaudhuri in [97]. The method uses a decision-tree-based classifier along with features like projection profile, statistical and topological features, and stroke features for discrimination of printed Latin, Urdu, Devnagari, and Bengali script lines. Subsequently, they proposed an automatic system for the identification of Latin, Chinese, Arabic, Devnagari and Bengali text lines in printed documents [98]. Later, a generalised scheme for script line identification in printed Indic multi-script documents was proposed by Pal et.al. [102], their method considered 12 Indian scripts like Devnagari, Bengali, Latin, Gujrati, Kannada, Kashmiri, Malayalam, Oriya, Gurumukhi, Tamil, Telugu and Urdu. They used features like headlines, horizontal projection profile, water reservoir-based features, left and right profiles, and feature based on jump discontinuity. They achieved script identification accuracy of 97.52% at line level. Sinha et al. [126] proposed a word-wise script identification scheme for Indic scripts. Patil and Subbareddy [104] proposed a neural-network-based system for word-wise identification of English, Hindi and Kannada language scripts. Zhou et al. [145] proposed a Bengali/English script-identification scheme using connected component analysis.

Elgammal et al. [52] proposed a method for discriminating Arabic and English text. They used number of peaks and the moments in the horizontal projection profile and the distribution of run-lengths over the location-length space as features. They noted that an Arabic text line generally has a single peak in the horizontal projection profile , while that of an English text line has two major peaks. This difference in the number of peaks can be used to distinguish these two scripts. They also used the third and fourth central moments of the horizontal projection profiles as features. They achieved an accuracy of 96.8% at the word level.

Tan et al. [39] and Lu et al. [88] proposed word-wise script identification using character shape codes method. In the method proposed in [39] basic document features like elongation of bounding boxes of character cells, and the position of upward concavities were used to generate shape codes that performs script identification involving Latin, Han, and Tamil script in printed text. In [88], the proposed method follows a template matching approach for script identification. A Hamming distance between the query image and the template word image is considered for script identification. They obtained 99% accuracy in discriminating eight Latin-based scripts. A bilingual OCR for printed documents was developed due to [76] that identifies Devnagari and Telugu scripts by considering the presence and absence of 'shirorekha' or the 'head-line' feature .

Lee and Kim [85] proposed a self-organising network-based method to identify printed character script type and classify them amongst Latin, Chinese, Korean and mixed scripts.

3.3.2 Visual Appearance-based Script Identification Method

Wood et al. [141] proposed a method using vertical and horizontal projection profiles of document images for determining scripts. Busch et al. [30] used features like wavelet energy features, wavelet log mean deviation features, wavelet co-occurrence signatures,

wavelet log co-occurrence features, and wavelet scale co-occurrence signatures. Their experiments involved eight different script types: Latin, Han, Japanese, Greek, Cyrillic, Hebrew, Devnagari and Farsi. In their experiments, printed document images were first binarised, skew corrected, and text block normalised. Later, Fisher linear discriminant analysis technique was applied to reduce the dimensionality of the feature vectors. Finally, Gaussian Mixture Model (GMM) was used for script classification. Using texture analysis based features, Tan [134] described an automatic method for identification of Chinese, English, Greek, Russian, Malayalam and Persian text and obtained an accuracy of 96.7%. Pan et al. [103] proposed a scheme to discriminate Chinese, Japanese and Korean script using Gabor filter-based features. They extracted features on a 256×256 pixel window and used an ANN classifier for the classification task. But in a real life scenario, this approach could face some problems where we might have characters from two different scripts within this window area. Some script identification schemes at the word level using texture based features are reported in [89], [72], [48]. Ma et al. [89] extracted Gabor filter based features from each word in a bilingual document. In their paper, Ma et al.[89] considered bilingual documents consisting of one Latin-based script and one non-Latin script. Dhanya et al.[48][47] mentioned two approaches for word-wise script identification from bi-lingual documents containing English and Tamil scripts. In one article they proposed a method which structures words into three distinct spatial zones and utilises the information about the spatial spread of the words in these zones. In the other article the directional energy distribution of words using Gabor filters with suitable frequencies and orientations were analysed.

3.4 Writer Identification

Writer identification is a vibrant field of research due to its scope of application as a computational forensic method. There are many pieces of seminal work on writer identification [27], [58], [93], [112], [114], [118], [119], [131], [132], [147]. The present state-of-the-art on writer identification can be clustered into a few groups based on (a) feature type used (gloabal texture-analysis features, local shape features etc) and (b) the approach (text-independent or text-dependent writer identification). Dependent on the text content, text-dependent methods only match the same characters and require the writer to write the same text again. The text-independent methods are able to identify writers regardless of the text content, and does not require comparison of exactly the same characters in a definite sequence [130]. For better readability, we describe the present state-of the art in writer identification problems with respect to different approaches (text-independent and text-dependent).

3.4.1 Text-Independent Writer Identification

Said et al. [112] developed a writer identification system using Gabor filter and greyscale co-occurence matrix-based features. Gabor filter features achieved 96% writer identification accuracy, outperforming greyscale co-occurrence matrix features. In the proposed method of Helli and Moghaddam [66], Gabor filter-based features were represented for each person by using a graph. This graph is constructed using relations between extracted features by employing a fuzzy method. This determines the similarity between features extracted from different handwritten samples of each person. In the identification phase, a graph similarity approach is deployed to determine the writer.

Schlapbach et al. [114] proposed an HMM-based writer identification and writer verification method. For handwriting of each writer an individual HMM was trained. To identify the writer of an unknown text, the unknown text needs to be presented to all HMM. Each HMM returns a log likeli-hood score. The HMM associated with a particular writer and which obtained the highest score was assumed to be identified writer. The experiment involved 650 writers. The identification accuracy was 96.56%. Also, their scheme

was tested in a writer verification framework. This experiment involved writings from 120 persons. Experimental results showed an Equal Error Rate (EER) of about 2.5%.

Two very similar approaches are due to Marti et al. [93], and Hertel and Bunke [67]. According to those articles, utilizing several visual characteristics of a handwriting, text-independent writer identification can be performed. Those visual characteristics of handwriting are (a) height of the three main writing zones, (b) slant and width of the character, (c) the distances between connected components, (d) the blobs enclosed inside ink loops, and the upper/lower contours. Later [115] shows that, using a k-nearest-neighbour classifier coupled with some feature selection methods, identification rates exceeded 90% in test cases on a subset of the IAM database considering 50 writers [92].

Bensefia et al. [11][9] proposed a grapheme based method, where graphemes extracted from the segmentation of cursive handwritten text encode the individual characteristics of handwriting. Their proposed scheme achieves 95% correct identification on the PSI database and 86% on the IAM database. Writer identification was performed in an information retrieval framework, while for writer verification graphemes from two different samples were used for comparison.

A system-based on Gaussian mixture models (GMMs) was proposed for writer identifcation in handwritten text on whiteboard data by Schlapbach et al. [116]. The mixture models provided a method of representing the distribution of the features extracted from the handwritten text. Bulacu et al. [25] used edge-based directional probability distributions extracted from handwriting images as features in forensic writer identification. Later, Bulacu et al. in [26] consider each writer as a stochastic generator of ink-trace fragments, or graphemes. Using the probability distribution of these graphemes a common codebook of graphemes are generated by a clustering algorithm that characterizes a writer. They used a complementary shape representation using normalised bitmaps to encode graphemes. The article also compared three different clustering methods for generating the grapheme codebook: k-means, Kohonen SOM 1D and 2D. In another work Schomaker and Bulacu [118] perform offline writer identification using connected-component contours in upper-case handwritten samples. Later, Bulacu and Schomaker [27] proposed a texture level and allograph level feature-based writer identification scheme.

Recently, Brink et al. [23] proposed a writer identification system where the width of each ink trace, combined with its direction, is used as a source of information for off-line writer identification. Such measurements were computed based on pixel contours which they term as the Quill feature. It is a probability distribution of the relation between the ink direction and the ink width. They obtained encouraging results using a Nearest Neighbour Classifier.

Approaches based on graphemes are evident in [8][10], where in [8] a morphological grapheme-based analysis followed by a template matching technique performs the task of writer identification.

Thumwarin and Matsuura [133] proposed an online writer identification method for Thai script based on the velocity of the barycentre of pen-point movement. The barycentre was determined from the centre point of script and two adjacent pen-point positions with respect to time in the handwriting process. In their article, the Fourier coefficients of the velocity and trajectory of the barycentre were considered as the input and output of a finite impulse response system. The impulse response of the system is interpreted as a feature of the handwriting. A Gabor filter feature-based approach was conceived by He et al. [65] for writer identification in the context of Chinese script. Recently, Bhardwaj et al. [12] proposed to use a generative model in the form of Latent Dirichlet Allocation(LDA) that automatically infers writing styles from a handwritten document collection without any pre-defined set of rules. Later, this information was used to represent each writer as a distribution over multiple writing styles for classification of any test sample. They experimented using two different feature sets consisting of contour angle features as well as structural and concavity features.

3.4.2 Text-Dependent Writer Identification

Srihari et al. [131] proposed two different types of features for writer identification in a text-dependent scenario. According to [131], features that can inform us about grey-level entropy and threshold, number of ink pixels, number of interior/exterior contours, number of four-direction slope components, average height/slant, paragraph aspect ratio and indentation, word length, and upper/lower zone ratio are termed as "Macrofeatures". Those features can operate at document/paragraph/word level; whereas features that can provide information about gradient, structural, and concavity attributes and can operate at either word or character level were termed as "Microfeatures". Their experiment involved thousand writers, and each writer copied a fixed text of 156 words (the CEDAR letter) three times. In their experiment, microfeatures outperformed macrofeatures in identification tests with an accuracy exceeding 80%. In later years, Zhang et al. [143] proposed a text-dependent method where a handwritten word image was characterized by gradient, structural, and concavity features. Their corpus comprised of four different English words written by 1000 individuals. Final classification was carried out by K-nearest neighbour classifier. Another text-dependent approach is proposed by Zois and Anastassopoulos [147]. In their proposed method, a feature vector was derived by means of morphologically processing the horizontal profiles (projection functions) of a word in English and Greek. They also provided an extensive study of the statistical properties of the feature space. They performed a comparison of accuracy between bayes classifier and neural network in the context of their experimental setup. A text-dependent writer identification system was proposed by Gupta and Namboodiri [62] where a cascade of classifiers was used. There they made some experiments with text from Devnagari script. A very recent endeavour by Somaya Al-Maadeed [3] details an Arabic script text-dependent writer identification scheme. The author used features like edge-direction distribution, moment invariants, and word measurements to perform the writer identification task.

3.4.3 Off-line Writer Identification in Context of Indic Scipts

Though a large number of people in the world use Indic scripts, to the best of our knowledge, there are only few works on writer identification in the context of Indic script [59][107]. Purkait et al. [107] proposed a text-dependent writer identification scheme for Telegu script. They used morphological features along with K-nearest Neighbour classifier, and obtained encouraging results. Garain et al. [59] proposed an AR co-efficient feature-based writer identification system for 40 Bengali writers. They used at least 200 words per writer for training and testing their system. But, very often a questioned document does not have such a huge number of handwritten text words. Hence, to analyse a questioned document of Indic script with a smaller amount of information, a reliable writer identification system is needed. In order to deal with problem, like scarcity of data content in questioned documents, we here propose a robust writer identification system for two Indic scripts: Bengali and Oriya.

3.5 Font Identification

Some of the earliest work on font identification is due to [148][146][81][124] etc. But mostly they dealt with Roman script fonts. Using typographical features font detection has been described by Zramdini et al. [148]. Zhu et al. [146] described an automatic method for identification of six Chinese and eight Roman fonts. He used a Gabor filter-based texture analysis approach for discriminating fonts. Khoubyari et al. [81] reports a system which could identify 33 different fonts of Roman script. A Nearest-neighbor classifier has been used by Manna et al. [90] for discriminating between four different fonts in Roman script. Shi et al. [124] proposed a system to discriminate between nine Roman fonts. There they have used properties of the input page and also used graph matching results of recog-

nized short words. A font identification system for Arabic script is being proposed in [2]. Slimane et al.[127] proposed an Arabic font recognition system which does not require a character segmentation in Arabic script text. With the use of Gaussian mixture model they efficiently deal with 10 different fonts with 10 different font sizes. In recent past this work is extended by Slimane et al.[128] for the purpose of dealing with low resolution images. Another font identification methodology for arabic text is due to Ben Moussa et al.[7], they proposed using of fractal geometry for global texture analysis for font recognition in an arabic document image. The main features were obtained by combining BCD (box counting dimension) and DCD (dilation counting dimension) techniques. They obtained an average recognition rate of about 96.2% using KNN (K nearest neighbor) and 98% using RBF (radial basic function). Font identification on chinese script is proposed by [49]. They perform Box-Cox transformation and LDA (Linear Discriminant Analysis) process on the chinese character thereafter extracting wavelet features from those characters. The final classification is performed using a MQDF (Modified Quadratic Distance Function) classifier and obtained a recognition rate of 90.28% on a single unknown character and 99.01% when five characters are used for font recognition. It can be noted that the present state-of the art addresses font identification mainly from a Roman script perspective. There is no systematic study on font identification in context of any Indic script. Although the Indian subcontinent is home to a huge population with eleven official scripts.

Chapter 4

Contribution, Summary and Practical Considerations

The studies described in this thesis are motivated by the increasing demand for a reliable automatic questioned document analysis system for torn document fragments. A forensic expert might overlook a tiny piece of evidence while dealing with a huge pool of document fragments. An automated system can be developed to assist him, such a system will sort similar document fragments based on the similarity of their content types. Such an automated system should be able to handle challenges like arbitary orientation, different document layouts etc. This thesis intends to remove those mentioned bottlenecks of handling huge pool of torn document fragments for forensic analysis, by developing a true generic forensic analysis system for torn document fragments. This could narrow down the search space for a human forensic expert and help him to focus on relevant evidence.

The thesis at hand tries to solve the above mentioned problems in order to aid a human forensic expert. Associated research questions addressed in this thesis are as follows :

- How can we develop a hierarchical/scalable approach capable of handling any document pieces for content-wise zone classification?
- What kind of features will help to discriminate printed and handwritten text?
- What kind of features will help to discriminate different scripts like Roman, Arabic, Devnagari, Bengali, Oriya, Urdu, etc.?
- How can we handle text-independent writer identification problems with limited data from each writer?

4.1 Brief Discussion on First Three Introductory Chapters in Part I

For better readability a brief discussion on chapter 1-3 is provided below.

- **Chapter 1** introduces the topic of the thesis, together with the motivation and challenges associated with the thesis. Proper investigation reveals that the problem of automatic forensic analysis on document fragments can be countered by solving various sub-problems like (a) content-wise zone segmentation in document fragments; (b) identifying script of the text in document fragments and then grouping documents having similar script text; (c) grouping document fragments based on writer of the handwritten text; (d) grouping document fragments based on the font of printed text. With the help of a flowchart, a logical explanation is given, showing how the research of this thesis aims to solve the problem of automatic forensic analysis of torn document fragments. The chapter also mentions the primary contributions of the thesis and finally provides an overall outline of the thesis.
- **Chapter 2** discusses the necessary background theories that have been used as the basis of the solutions for different sub-problems associated with the problem of automatic forensic analysis on document fragments. As writer individuality characteristics could play an important role in grouping similar document fragments, a discus-

sion is provided on the reasons behind the uniqueness of handwriting amongst different persons. Different aspects of script and font are also discussed in this chapter, since script and font information could play an important role in clustering similar document fragments. The chapter ends with discussions on statistical learning theory which acts as the pillar for SVM classifier, a theoretical discussion on SVM and how SVM ensures a better generalisation property of the learning model.

- **Chapter 3** presents the existing state-of-the-art methodologies on relevant sub- problems that we need to deal with in order to accomplish our objective, for example we narrate here existing state-of-the-art methodoligies on (a) Text/graphics segmentation ; (b) Script identification ;(c) Writer identification ; (d) Font identification.

4.2 Brief Discussion on Paper Contributions

- **Chapter 5 "Document-Zone Classification in Torn Documents"** addresses the problem of text-graphics segmentation and text type discrimination (i.e. whether a text is printed or handwritten) in the context of torn document fragment images. All torn images considered in our experiments consist of sparse data with arbitary orientation and random background color and texture. A two-stage approach is followed here where initially text and graphics components are segmented, then discrimination between printed and handwritten text is performed. Two different feature extraction methods were deployed at two stages. In the first stage, Gabor filter-based features were used for segmentation of text and graphics regions present in the document fragment. In the second stage, text components were processed with chain-code histogram feature in order to differentiate the text type between printed and handwritten text.

- **Chapters 6 and 7 "Text Independent Writer Identification for Bengali Script" and "Text Independent Writer Identification for Oriya Script" respectively** - discuss writer identification problem for two different Indic scripts (Bengali and Oriya) in a constraint environment of sparse text. Features like Gradient, Chain-code histogram and curvature were chosen keeping in mind the structural properties of the text script. For example, Oriya characters are mostly circular in shape, Curvature-based features were chosen to represent the difference in degree of roundness in characters written by different persons. For Bengali text, experiments were conducted using two feature extraction methods namely the chain-code histogram feature and gradient feature. Both feature extraction methods give us an idea about the structural shape of the character-components under consideration. Chain-code histogram defines the structural shape by means of keeping count of strokes in different directions whereas the gradient feature extraction method defines the structural shape by using the gradient information in pixel distribution. We used SVM as the classifier for both cases. Analysis on different numbers of top choices was performed. We obtained encouraging results (95.19% for Bengali script considering 104 writers, and 94% for Oriya script considering 100 writers).

- **Chapter 8 "Identification of Indic Scripts on Torn-Documents"** - investigates the issue of script identification in torn document fragments. We executed experiments and compared two different kind of feature types namely Gradient (rotation- dependent) and Zernike moments (rotation-independent). Merits and demerits of both feature types were discussed. For the gradient feature extraction, a pre-processing step becomes inevitable, which detects the orientation of the text and then the concerned text was rotated to appear in normal orientation. During the pre-processing step, the texts were processed with morphology operator to make them appear thick yet maintain their original shape and appearance. This was done to counter the sparse pixel

distribution in those texts.Then a PCA-based approach was deployed to detect the correct orientation of the text. Torn document fragments with eleven different scripts in them were used in our experiments. We obtained a script identification accuracy of 94.65% at word level in such torn document fragments while using Gradient features.

- **Chapter 9 "Script Identification - A Han and Roman Script Perspective"** - discusses the case of script identification in normal document images considering Roman and three Han scripts. Script identification between Chinese, Japanese and Korean script is a challenging task. After line and consequent character segmentation is performed Chain-code histogram-based features were extracted from each segmented characters. Chain-code feature represents the shape of the text/character by keeping a count of strokes in different directions. This feature extraction method gives a detail shape information about the character-component under consideration. The feature extraction method proved its efficacy in discriminating scripts like Chinese and Japanese, which are sometimes visually very much alike.

- **Chapter 10 "Font identification - In context of an Indic script"** - deals with font identification in the context of an Indic script (Bengali). Curvature based features were used to exploit tiny differences in shapes of characters present in different fonts of Bengali text. Initially as a pre-processing step: line, word and character segmentation were performed on the binary representation of the whole document image. Later, curvature features were extracted from each segmented character, and those features were fed to the classifier to identify the font of that character. Finally, using majority voting technique, the font of the document image is being asserted. The Support Vector Machine and its Multiple Kernel variant were used for classification of fonts of characters. Our experiments considered 400 test documents with 10 different fonts.

The rest of this chapter provides a discussion on contribution of the thesis, practical considerations and future works.

4.3 Summary of Thesis Contribution

- Segmentation of a document image on the basis of its content has been a topic of interest for a long time amongst the researchers. However the problems of segmentation on torn document images were rarely investigated. Torn documents are characterized by sparse and arbitrary oriented data, which makes this apparently simple problem a real challenging task. The proposed method is capable of segmenting text and graphics. In addition it discriminates the text between printed and handwritten text even with sparse and arbitrary oriented texts present in torn document images.

- Writer Identification problem has been rigorously explored by many researchers. However all those methods assume or need huge amount of data for reliable performance. Morover most of those methods deal with Roman script text. In a real life crime scenario it might not be possible to aquire huge amount of text/data from the crime spot. Our proposed method can perform writer identification with high accuracy even when there is scarcity of text in both training and testing document images. Features used in our experiments were chosen keeping in mind the structural shape of the character of the text in context. We got comparable accuracy with other methods even while using very small amount of text for training and testing our system.

- Even though the problem of script identification has been nurtured by many researchers. Our analysis revealed that all such methods basically addressed the problem of script identification in terms of normal document images. However, in a real life forensic analysis scenario, encountering full sized document images may not be

obvious all the time. Rather, it might be useful to perform script identification in torn document images. Contents in such torn document images have random orientation. Our proposed method is capable of detecting orientation of the text present in those torn documents and then performing consequent orientation correction of the text and feature extraction from the text. Our experiments involved document fragments consisting of text of all major Indic scripts, Han scripts and Roman script. Encouraging results were obtained and we demonstrated that it is better to perform feature extraction after fixing the orientation of the text rather deploying an orientation independent feature extraction method.

- Even though a huge population in the world uses Indic scripts, font identification has been mainly explored in context of Roman and Arabic scripts. To the best of our knowledge, no research work has been devoted to font identification of any Indic script. We proposed a method for font identification of an Indic script "Bengali" using curvature feature and classifier like MKL-SVM and SVM, and experiments were carried out involving 10 different Bengali fonts. Features were extracted from characters after a document image is processed for line, word and character segmentation respectively. Fonts of every single character were classified and later, based on majority voting on font classification of every character, the font of the test document image was asserted.

4.4 Practical Considerations

Any questioned document analysis process, irrespective of the approach (manual or automated), demands high reliability, since the outcome of the examination plays a vital role in convicting an accused person. Special care should be taken to ensure that questioned document fragments are not manipulated either physically or digitally. Deletion, falsification of digital images, intrusion of physical/digital documents, and manipulation of physical documents are common threats to such a system. The methodologies discussed in this thesis can work as desired only when such threats are prevented. For a real life questioned document analysis system, a pre-processing step should be devoted to check for such malfunctions. A brief discussion on such threats and how we can encounter them are as follows:

- The image acquisition process of such a system must apply a digital signature and time stamp while scanning all document images found at the crime scene. This could prevent the intrusion of digitally manipulated document images into the system at later stages.

- In the case of handwritten document fragments, common ways of alteration are:- (a) insertion of a word to change the meaning of a sentence; (b) obliteration or darkening of some portions of the text using some opaque ink (smeared over-writing);(c) augmentation of various characters by adding strokes to the original character, thereby forming a new valid character. This could be done with high reliability, since shapes in Roman characters are sometimes close to each other especially in handwritten form. For example a forger can easily transform a '3' to '8', or 'P', to 'R', etc,. Micro-spectrophotometry-based ink-analysis in the pre-processing step can help us detect such forgeries.

4.5 Future Work

The thesis takes a multi-disciplinary approach and considers various image processing and pattern recognition techniques to solve the problem relevant to the context. Our methods

gave satisfactory results, yet we can explore further towards a more robust and "intelligent" automatic questioned document analysis system. A brief discussion on the future issues is as follows:

- The writer identification problem for Indic script in a constraint environment (e.g. sparse data per writer) has been investigated in the thesis. Though the results obtained are quite good, we can think of an approach of combining local structural features with a global texture analysis-based features, which might give us higher accuracy. Another possibility is implementing an L_1 norm MKL-SVM where every base kernel represents a different feature type, as we know that different features at different base kernels might enhance the accuracy. Generally, L_1 norm regularizer on the kernel weights will eventually select those features that are best for writer discrimination. A theoretical framework may also be developed to explain why certain character-allographs exhibit maximum discrimination amongst different writers. Moreover, we could also investigate whether the amount of text/data per writer required for training is correlated with the number of classes involved in the writer identification problem. That means, whether we need to consider more training data per writer to retain high accuracy when the number of writers/classes increases.
- For writer identification, our proposed scheme performs line, word and primitive (character-component or its parts) in a document image. Then features are extracted from individual primitives. Using those extracted features, SVM classifies these primitives and associates them with one of the writers. Later, a majority voting technique is used to assert authorship of the entire document. We can investigate to see if we can develop an approach where writer identification can be done without taking the burden of line, word, and character segmentation.

Finally, let us hope that this work will encourage research activities on torn document analysis and in particular torn document analysis for Indic scripts.

Chapter 5

Document-Zone Classification in Torn Documents

Abstract- Arbitrary orientation and sparse data content are common characteristics of torn document. To ensure accuracy and reliability in computer-based analysis, content-zone segmentation is required. In our previous work, we studied segmentation of handwritten and printed text. A questioned document-piece in the form of an office note, however, might also contain non-text data like logos, graphics, and pictures. Hence a more precise content-zone classification is required. In this chapter we propose a two-Tier approach for non-text, handwriting and printed text segmentation. The first Tier aims to discriminate text and non-text regions. The second Tier classifies handwritten and printed text within all text zones identified during the first Tier. Gabor features and chain-code features are used in Tier-1 and Tier-2, respectively. By using SVM classifier we successfully identified 97.65% of 31,227 text regions in our current test data. The proposed approach identified 98.69% of printed and 96.39% of handwritten text amongst all identified text regions.*

Keywords:- Torn Document Recognition, Text Classification, Printed and Handwritten Text Segmentation, Text Graphics Segmentation.

5.1 Introduction

The forensic analysis of questioned documents is challenging when dealing with scarcity of data content. Such situation often occurs in the investigation of torn documents (compare Figure 5.1.) Moreover, not only sparse data content but also arbitrary orientations of the document fragments are encountered when these have been digitized in large volumes. Document fragments might contain parts of graphics, logos, and pictures along with handwritten and printed texts. All these entities demand different processing and feature encoding for accurate analysis. Hence the preferred approach is to identify different zones first, followed by processing each segmented zone with respective modules, e.g. optical character recognition or image understanding. Approaches for segmenting graphics, machine-printed and/or handwritten text are known. These approaches have been mainly developed for application in digital libraries, i.e. the digitalization of books, magazines and office documents. Our initial studies revealed that known segmentation methods perform less when dealing with torn documents fragments. The main reasons are: (i) Text orientation in torn document fragments can be arbitrary. (ii) Fragments are rather small with an average size of 7x10 cm, and (iii) contain a limited amount of text data. Several researchers have studied the standard problem of text/graphics segmentation. Fletcher et al. [55] used a set of spatial characteristics of text/graphics component to filter out different components. Chowdhury et al. [38] worked on similar approach as Fletcher et al.[55]. Shafait et al. [122] proposed a novel pixel accurate representation for arbitrary shaped page segments. They also propose performance measures to identify and analyze different classes of segmentation errors. An et al. [4] reported an algorithm of

*Content of this chapter is mainly based on the article - "Document-Zone Classification in Torn Documents", Sukalpa Chanda, Katrin Franke and Umapada Pal, In Proc. 10th International Conference on Frontiers in Handwriting Recognition , pp.25-30, ICFHR 2010.

segmenting photographs, handwritten text, printed text, blank spaces etc., using a multi-stage post classifier-based approach. Patricio et al. [105] used Fourier descriptors for the purpose of text and graphics segmentation. Some research publications on segmenting machine print and handwritten English text exist. Kandan et al. [78] used central moments-based feature with Nearest Neighbor (NN) and Support Vector Machine (SVM) classifier. Guo et al. [61] used Hidden Markov Model (HMM) for extracting handwritten text words from printed text documents. Recently, Zheng et al.[144] reported on printed and handwritten text segmentation using K-NN, SVM and Fisher classifier with features like pixel density, aspect ratio and Gabor features. Kuhnke et al. [83] developed a method to identify machine-printed and handwritten English characters. Similar works but on oriental scripts have been reported also. Pal et al. [99] developed a scheme for segmenting printed and handwritten text of two Indic scripts. Jang et al. [75] proposed a system for segmenting handwritten Korean text from machine-printed Korean text using geometric features and a Multi-Layered Perceptron (MLP) classifier. This paper deals with discriminating text and

Figure 5.1: Example of document images with graphics and printed text present simultaneously.

non-text zones in torn document fragments. The fragments comprise few data content and are arbitrarily oriented. We propose a two- Tier approach to deal with these challenges. For the segmentation of text and non-text zones we use Gabor features in the first Tier. For segmenting printed and handwritten text in the second Tier, we are using a wellknown, robust and fast 64 dimension chain-code-based feature. A Support Vector Machine (SVM) classifier is used for classification in both Tiers. To the best of our knowledge this is for the first time the problem of detecting pictures, sparse handwritten and printed Roman text in the context of arbitrary orientation is addressed.
The organization of the paper is as follows. In Section 5.2 we present the overview of our method. In Section 5.3 we discuss the experimental setup used for evaluating our scheme. Here we also provide details on the dataset used. In Section 5.4 we present our results followed by conclusion in Section 5.5.

5.2 Method Overview

In this Section we are providing an overview of our two- Tier approach for segmenting non-text data (graphics, logos, pictures), handwritten and printed text in torn documents (Subsection 5.2.1). We also describe our method of orientation estimation (Subsection 5.2.2) and we present details on the features and classifier implemented (Subsection 5.2.3 and 5.2.4).

5.2.1 General Overview

A torn document fragment comprises sparsely written text along with small parts of graphics/pictures (Figure 5.1.) Hence one needs to analyze data content locally. In order to analyze data locally we deployed a sliding window strategy. For the sake of computational

efficiency non-overlapping NxN pixel windows are placed over the entire image and the region beneath each window is classified into a text (printed and/or handwritten) or non-text region. In order to decide whether a text region or a non-text region is present, spectral features i.e. Gabor features are used for texture analysis. During our initial experiments we noticed that Gabor features were capable to discriminate text and non-text, yet it could not further discriminate text regions into printed and handwritten. In order to utilize the presence of curvatures in handwriting for discrimination from printed text, we decided to use a variant of local shape descriptor feature. Subsequently, two-Tier architecture was established that is shown in Figure 5.2. In the first Tier, we discriminate text and non-text region. In the second Tier, we process the identified text regions to discriminate between handwritten and printed text. We noticed that without special treatment the arbitrary orientation of text found in the document fragments might yield wrong classifications of printed and handwritten text. Thus, another preprocessing step was needed. We estimate and correct the orientation of text prior to classification in Tier-2 as detailed in the following Subsection 5.2.2.

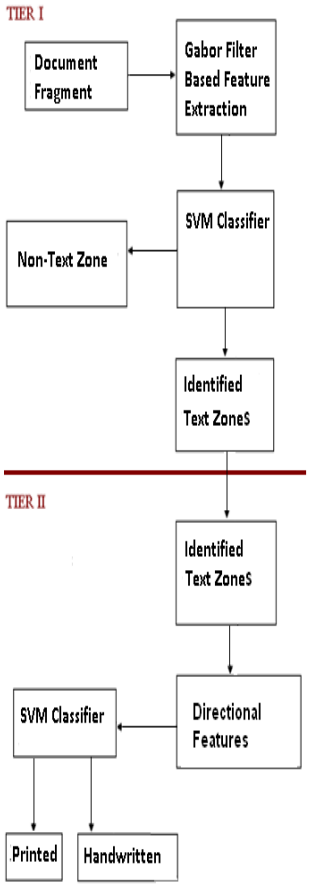

Figure 5.2: Schematic overview our two-Tier architechture.

Our method proposed for content-zone classification in torn documents could be summarized as follows: (i) Perform texture analysis using Gabor features described in Subsection 5.2.3 and identify all text regions; handwritten and machine printed text. (ii) For each labeled text region and word component determine the direction of highest variation (extension) using an implementation of principal-component analysis (PCA) [120]. (iii) Rotate

each word component according to the direction of its first eigenvector. (iv) For all character components compute local shape descriptor feature as described in Subsection 5.5.3. (v) Perform classification of character components and apply majority voting scheme for each word component. (vi) In case of a tie, declare those set of character components as rejected character components/words.

5.2.2 Orientation Estimation

For discriminating handwritten and printed text in Tier-2, the expected arbitrary orientation of document pieces demands a dedicated processing step - the estimation of text orientation. It should be noted that a document piece might contain handwritten and printed text in different orientations. We considered a PCA-based approach proposed in [120]. However, during our initial experiments we noticed that this method was not always able to derive the correct direction for a group of connected components (word-component). Analyzing the cause we realized that due to sparse pixel distribution in those components the orientation estimation is not robust. We overcome this limitation with the help of dilation operator (with a rectangular shape structural element of dimension 1 x 3)[†], applying five-times-dilation operation of Mathematical Morphology. See figure 5.3 for illustration.

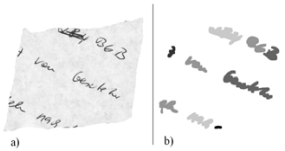

Figure 5.3: Document piece with small amount of handwriting: (a) Original image and (b) Same image after binarization and dilation operation.

5.2.3 Feature Extraction

We have chosen two different sets of features for our two- Tier approach. The objective of Tier-1was to discriminate text and non-text part (picture/diagram/logo) in a document image. The spectral patterns for a text and non-text region are quite different and therefore well suited for texture analysis. Gabor filter-based features are therefore used in Tier-1. For Tier-2, we use a chain-code histogram-based feature. The main difference between handwritten and printed text is their curvature and shape structure. Most of the Roman printed text-characters have a uniform shape. Whereas handwritten Roman texts are of arbitrary curly allograph styles. In addition, there are many straight strokes and lines for printed text that are absent in their handwritten counterpart. Those characteristics are well represented by our proposed directional histogram-based feature. The details of feature extraction for both Tiers are given below.

Gabor Filter: A two-dimensional Gabor filter in spatial and frequency domain can be defined by the following formula:

$$g(x, y, \lambda, \theta, \psi, \sigma, \gamma) = exp(-\frac{x^{'2} + \gamma^2 y^{'2}}{2\sigma^2}) \cos\left(2\pi \frac{x^{'}}{\lambda} + \psi\right)$$

where

$$\begin{cases} x^{'} = x\cos\theta + y\sin\theta, \\ y^{'} = -x\sin\theta + y\cos\theta, \end{cases}$$

[†]Text changed/added.

In this equation, λ represents the wavelength of the cosine factor, θ represents the orientation of the normal to the parallel stripes of a Gabor function, ψ is the phase offset, σ is the sigma of the Gaussian envelope and γ is the spatial aspect ratio, and specifies the ellipticity of the support of the Gabor function. We tried a combination of different values of these parameters during our experiment. The method used in extracting features is as follows: (i) Slide a non-overlapping window of NxN pixels over the fragment image. (ii) Compute the corresponding Gabor value within each NxN window using the formula above. (iii) Encode the Gabor value of each of the pixels in the sliding window as a vector component of our feature vector. We experimented three different window sizes: 10x10, 20x20 and 30x30 pixel.

Chain-Code Histogram Feature: At first we compute the bounding box of a character component. This bounding box is then divided into 7 x 7 blocks. In each of these blocks the direction chain code for each contour point is noted and frequency of direction codes is computed. Here we use chain code of four directions only: directions 1 (horizontal); 2 (45 degree slanted); 3 (vertical) and 4 (135 degree slanted). See Figure 5.4 for illustration of four chain-code directions. We assume chain code of direction 1 and 5 are same. Also, we assume direction 2 and 6, 3 and 7, 4 and 8 are equivalent, because if we traverse from point 4 to point 8 its going to have the same count as point 8 to point 4. Subsequently, in each block, we get an array of four integer values representing the frequencies of chain code in these four directions. These frequencies are used as feature. Thus, for 7 x 7 blocks we get 7 x 7 x 4 = 196 features. In order to reduce the feature dimension, after the histogram calculation in 7x7 blocks, the blocks are down sampled with Gaussian filter into 4 x 4 blocks. As a result we obtain 4 x 4 x 4 =64 features for further classification. To normalize the features we determine the maximum value of the histograms from all the blocks and divide each of the above features by this maximum value to get the feature values between 0 and 1. A more detailed description about the feature can be found in [101].

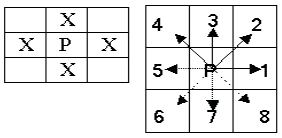

Figure 5.4: For a point "P" the direction code of its eight neighboring points is shown.

5.2.4 Classifier Details

In our experiments, we have used a Support Vector Machine(SVM) as classifier. The SVM is defined for two-class problem and it looks for the optimal hyper-plane, which maximizes the distance,the margin, between the nearest examples of both classes, named support vectors (SVs). Given a training database of M data: $\{x_m \mid m = 1, \cdots, M\}$, the linear SVM classifier is then defined as:

$$f(x) = \sum_j \alpha_j x_j \cdot x + b$$

where $\{x_j\}$ are the set of support vectors and the parameters α_j and b has been determined by solving a quadratic problem. The linear SVM can be extended to various non-linear variants, details can be found in [29][137]. In our experiments Gaussian kernel SVM outperformed other non-linear SVM kernels, hence we are reporting our recognition

results based on Gaussian kernel only. The Gaussian kernel is of the form:

$$k(x, y) = exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

For Tier-1, Gaussian Kernel gave highest accuracy when gamma parameter '$\frac{1}{2\sigma^2}$' is set to 32.00 and penalty multiplier parameter is set to 10 †. For Tier-2, we noticed that Gaussian kernel gave highest accuracy when the value of its gamma parameter '$\frac{1}{2\sigma^2}$' is set to 4.00 and the penalty multiplier parameter is set to 10 †.

5.3 Dataset And Experimental Design

Instead of applying two-Tier approach initially we tried to segment picture/diagram/noise, printed text and handwritten text in a single Tier. For this purpose we implemented Gabor filter and other spectral-based features, such as Fourier transform, grey level co-occurrence matrix features etc. Amongst all of them the Gabor filter features came out to be the best. But Gabor filter features were not robust enough to discriminate between hand-written and printed text, though it was highly successful in discriminating text and non-text component in a document image. We have experience of classifying printed and handwritten text in torn document images with high accuracy [34]. Considering our prior experience we applied here a two-Tier approach. In first Tier, we use Gabor filter features to isolate the text components (irrespective of types - handwritten or printed.) In the second Tier, we use a directional chain code feature for discriminating the text into handwritten or printed text. A schematic diagram of our two-Tier architecture is shown in Figure 5.2. We performed two experiments. One based on a single Tier approach where Gabor was deployed to segment between text and non-text region, and also further discriminate the text region into handwritten or printed text. In the second experiment we implemented our mentioned two Tier architecture. The results from both experiments are given in Section 5.4. It is clearly evident from the results that two-Tier architecture outperformed the single-Tier approach.

Our test dataset constitutes of 225 document images in tiff format of arbitrary orientation. From our test files we obtained 31,227 blocks of text data and 7913 blocks of non-text data (picture/graphics/logo/noise). The training dataset comprise of a different set of 225 document images in tiff format. The scanning resolution was 72 dpi for both training and testing document images †. In each of the training files, the data content was homogenous in nature but with arbitrary orientation. That means, the training files either contain texts or pictures/logos. From our training files we obtained 25136 blocks of text, and 6139 blocks of picture/diagram. Here, by block we mean to say region of 20×20 pixels. Also the background texture of all document images in both test and training set were heterogeneous in nature. Some were white, some were colored, and some had horizontal and vertical stripes in them.

5.4 Results and Discussions

We got best optimized results when orientation of Gabor filter was set to $\pi/6$, with spatial frequency set to 2, and sigma set to $2 * \pi$. Those values were set after a series of experimentation with combinations of different value for each parameter. This was the only filter used to generate the Gabor response †. Here in TABLE 5.1, we report the accuracy of our system in single Tier architecture. In a single Tier architecture, we used only Gabor filter as our feature to do the following task (i) Identifying the text regions present in a document image (ii) and then try to discriminate the text region into handwritten or printed text. In TABLE 5.2 we report accuracy of our system when we implement our mentioned two Tier architecture under all best optimized parameter value. In TABLE 5.3 we report

the accuracy of our system for Tier-1 with different size of the sliding window but with best-optimized Gabor-filter parameter.

5.4.1 Accuracy on Gabor filter-based features

In TABLE 5.1 we report the accuracy of our proposed scheme when we only use Gabor filters for discriminating text and non-text regions as well as discriminating printed and handwritten text. We can easily notice that Gabor fails to competently discriminate text regions between printed or handwritten text.

Table 5.1: System accuracy using Gabor feature only.

Correctly Identified text blocks	97.65%
Correctly identified text type(handwritten /printed)	Handwritten 39.13%
	Printed 46.50%

5.4.2 Overall accuracy of the proposed scheme

Here we report the overall accuracy of our system. It is evident from the results that Gabor were very successful in isolating text zones from non-text zone in Tier-1. In Tier-2, we processed those identified text zones of preceding Tier, with our chain code histogram-based directional feature. From those identified text zones of Tier-1 about 97.54% (98.69% + 96.39%)/2 of text was correctly identified as handwritten and printed text. We can easily notice that using our directional features in Tier-2, we could easily discriminate a text region as printed or handwritten text with much better competence.

5.4.3 Accuracy with respect to different window size

Below we demonstrate the effectiveness of selecting the right sliding window size for Gabor filter feature extraction. It is noted that even with the most optimized parameter set, the Gabor filter failed to perform well when the sliding window size was relatively small (10x10), but with same parameter values of the Gabor filter the accuracy was considerably enhanced with increased window size. In Figure 5.5, we illustrate the fact with two different output images of a test image. In one output image the size of the window was 10x10 and in other it is 20x20. From TABLE 5.3 it can be noted that the Gabor filter operating on a bigger window size performed better than the other. However, to make a trade-off between computational speed and accuracy we stick to 20x20 window size [†].

5.4.4 Error Analysis

We noticed that most of the errors in Tier-1 came out in case of blurred data content. This happened when the edges of the foreground data content tend to get fused with background texture. That is, when the contrast of the foreground part is very light compared to the background contrast. Examples of such images are shown in Figure 5.6.

Table 5.2: Overall system accuracy.

Tier I	Correctly Identified text	97.65%
Tier II	Handwritten	96.39%
	Printed	98.69%

Table 5.3: System accuracy for text class (rounded) at Tier-1 with 3 different window sizes.

Accuracy for Text regions in Tier-1.		
10 X 10	20 X 20	30 X 30
76.00%	97.65%	98.30%

Figure 5.5: Top, an input image. Bottom left, corresponding grayscale output image with window size 10. Bottom right, another output image with window size 20. In output images, identified text zones are marked with black blocks.

Figure 5.6: Document samples with low-contrast foreground.

In such cases, most of the time our Gabor features wrongly classified those texts as part of graphics/logos. In Tier-2, there were more errors in identifying handwritten text compared to printed text. This happened mostly due to wrong orientation detection of the handwritten character components.

5.4.5 Comparison with Similar Other Works

Since we deployed a two-Tier approach, we will compare our results separately for each Tier with similar work. In Table 5.4 we compare our results with published research on text/graphics separation whereas in Table 5.5 we compare our results obtained on segmenting handwritten and printed text. It can be noted from Table 5.4 that An et al. [4] obtained an accuracy of 80.02%, 29.08% and 69.24% for photographs, handwritten and printed text respectively. Chowdhury et al. [38] obtained an accuracy of 97.02% for detecting text and 97.97% in detecting graphics. Our method could successfully identify 97.65% of text blocks present in our test-data set. The research results reported in Table 5.5 considered segmentation of documents based on the content type for particular regions, but did not handle adverse conditions of arbitrary orientation and scarecity of data content as we have considered here. Examples of document fragments with multiple orientation and

sparse data are shown in Fig. 5.1. Yet the comparison is represented in a tabular format, where we compare our work with respect to those works in terms of accuracy, features, dataset size, methods used etc. We can see that Zheng et al. [144] got an highest accuracy of 96.00% using SVM classifier and using features like Gabor filter, Run length histogram features etc. They further improved their results to 98.10% by implementing a Markov Random Field-based post-processing step. Patricio et al. [105] obtained an accuracy of 99% in segmenting text and graphics[†]. Guo et al.[61] evaluated their scheme on document images consisting of 187 handwritten words. They obtained a precision of 92.86% from their scheme using HMM. Kandan et al. [77] used rotation invariant moment feature and obtained a highest accuracy of 93.22%. Our proposed method obtains an overall accuracy of 97.54% in identifying printed and handwritten text out of 97.65% correctly identified text blocks present in our test images.

Table 5.4: Comparison of results on graphics/text segmentation.

Method proposed by	Data set size	Feature used	Classifier used	Accuracy obtained
An et al. [4]	83 Document images	Pixel based features	5 NN using hashed K-d trees	80.02% (graphics) 29.08% (handwriting) 69.24% (machine print)
Chowdhury et al. [38]	200 document images	Structural features.	Rule-based system	97.02%(Text),and 97.97%(Graphics)%
Patricio et al. [105]	299 text regions 288 graphics regions	Fourier transform feature	Neural network (MLP)	99% (graphics), and 99% (text)
Our new method	31277 text blocks and 7913 non-text block	Gabor filter	SVM (Gaussian)	97.65% of all Text blocks

5.5 Conclusion

We propose a complete scheme for segmenting picture/graphics, sparse printed and handwritten text with arbitrary orientation in torn documents. We achieved satisfactory results when dealing with torn document fragments, also when there are not much text components in them. Possible source of betterment could be a method for more sophisticated binarization. In future, we could consider using Gabor filter with integral images and running pseudo- Gabor.

Table 5.5: Comparison of results on printed/handwritten text segmentation.

Method proposed by	Data set size	Feature used	Classifier used	Accuracy obtained
Zheng et al. [144]	4549 text blocks	Structural features,Run-length histogram features,Crossing count histogram feature, Gabor filter	SVM, Fisher, K-NN	96.00%(98.10%), 95.50%, 94.20%
Guo et al. [61]	187 handwritten words	Projection profile etc.	HMM	92.86% (precision)
Kandan et al. [77]	1678 handwritten elements in 150 document images	Moment feature	SVM, NN	93.22%, 87.85%
Our new method	31,277 text blocks of 20x20 pixels	Directional features	SVM:Gaussian kernel	98.69% (printed) and 96.39% (handwritten) Out of 97.65% of correctly identified text blocks from Tier -1

Chapter 6

Text Independent Writer Identification for Bengali Script

Abstract-Automatic identification of an individual based on his/her handwriting characteristics is an important forensic tool. In a computational forensic scenario, presence of huge amount of text/information in a questioned document cannot be always ensured. Also, compromising in terms of systems reliability under such situation is not desirable. We here propose a system to encounter such adverse situation in the context of Bengali script. Experiments with discrete directional feature and gradient feature are reported here, along with Support Vector Machine (SVM) as classifier. We got promising results of 95.19% writer identification accuracy at first top choice and 99.03% when considering first three top choices.*

Keywords- Text independent writer identification; Bengali script; Document Analysis; Computaional forensics.

6.1 Introduction

Writer identification is a vibrant field of research due to its scope of application as a computational forensic method. There are many pieces of work on writer identification [[27], [58], [93], [112], [114], [118], [119], [131], [132], [147]]. Said et al. [112] developed a writer identification system which is text independent, they took a texture analysis based approach. Schomaker and Bulacu [118] proposed an offline writer identification system, using connected-component contours in uppercase handwritten samples. Later Bulacu and Schomaker [27] proposed texture level and allograph level feature-based writer identification scheme. Srihari et al. [132] have used a combination of global and local features. Though a large number of people in the world use Indic scripts, to the best of our knowledge, there is only one work on Indic script [59] in the context of writer identification and they proposed an AR co-efficient feature-based writer identification system for 40 Bengali writers. They have used at least 200 words per writer for training and testing their system. But, very often a questioned document is deprived of such huge number of handwritten text words. Hence, to analyze a questioned document of Indic script with lesser amount of information, a reliable writer identification system is in demand. In order to encounter adversary, like scarcity of data content in questioned documents of Bengali script, we here propose a robust writer identification system.

6.2 Line And Character Segmentation

For line segmentation, at first, we divide the text into vertical stripes. Stripe width of a document is computed by statistical analysis of text height in the document [100]. Each of those stripes is processed to form Piece Wise Separating Lines (PSL) [100], and joining those PSLs we segmented the text lines. A histogram based approach was used to segment

*Content of this chapter is mainly based on the article - "Text Independent Writer Identification for Bengali Script", Sukalpa Chanda, Katrin Franke, Umapada Pal and Tetsushi Wakabayashi, In Proc. 20th International Conference on Pattern Recognition,pp.2005-2008,ICPR 2010.

words in each text line. Since most of the characters in a word in Bengali script are connected through headline, for character segmentation we first find individual component and compute their background portion using water reservoir principle [100]. Based on the water reservoir area, the touching characters in a word are segmented into individual character/character allograph. For details about line, word and character segmentation see [100].

6.3 Feature Extraction - Directional Features And Gradient Features

Character allographs of one particular writer are quite different from character allographs of other writers, even when they write same text. Our directional features are good local shape descriptors and hence they are capable of expressing this character allograph level dissimilarity present in the handwritings of different writers. Our gradient feature gives more information in terms of dissimilarity between character allograph of different writers. Dimension of our directional feature and gradient feature are 64 and 400, respectively.

6.3.1 Feature computation for directional features

At first we compute the bounding box of a character component. This bounding box is then divided into 7x7 blocks [101]. In each of these blocks the direction chain code for each contour point is noted and frequency of direction codes is computed. Here we use chain code of four directions only: directions 1 (horizontal); 2 (45 degree slanted); 3 (vertical) and 4 (135 degree slanted). See Fig. 6.1 for illustration of four chain-code directions. We assume chain code of direction 1 and 5 are same. Also, we assume direction 2 and 6, 3 and 7, 4 and 8 are equivalent, because if we traverse from point 4 to point 8 we will have the same count as point 8 to point 4. Subsequently in each block, we get an array of four integer values representing the frequencies of chain code in these four directions. These frequencies are used as feature. Thus, for 7x7 blocks we get 7x7x4 =196 features. In order to reduce the feature dimension, after the histogram calculation in 7x7 blocks, the blocks are down sampled with Gaussian filter into 4x4 blocks. As a result we obtain 4x4x4 =64 features for further classification. To normalize the features we determine the maximum value of the histograms from all the blocks and divide each of the above features by this maximum value to get the feature values between 0 and 1. See [101] to get details about this feature.

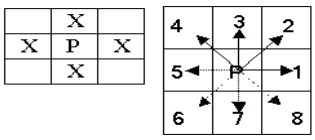

Figure 6.1: For a point "P" the direction code of its eight neighboring points is shown.

6.3.2 Feature computation for gradient feature

To obtain 400-dimensional gradient features we apply the following steps. (i) The input binary image of each character allograph is converted into a gray-scale image applying a 2x2 mean filtering 5 times. (ii) The gray-scale image is normalized so that the mean gray scale becomes zero with maximum value 1. (iii) Normalized image is then segmented into 9x9 blocks. (iv) A Roberts filter is applied on the image to obtain gradient image. The arc tangent of the gradient (direction of gradient) is quantized into 16 directions and the strength of the gradient is accumulated with each of the quantized direction. By strength of

Gradient(f(x,y)) we mean $f(x,y) = \sqrt{(\Delta u)^2 + (\Delta v)^2}$ and by direction of gradient $(\theta(x,y))$ we mean $(\theta(x,y)) = \tan^{-1}(\Delta u/\Delta v)$ here $\Delta u = g(x+1, y+1) - g(x,y)$, $\Delta v = g(x+1, y) - g(x, y+1)$ and $g(x,y)$ is a gray scale value at an (x,y) point. (v) Histograms of the values of 16 quantized directions are computed in each of 9x9 blocks. (vi) 9x9 blocks is down sampled into 5x5 by a Gaussian filter. Thus, we get 5x5x16=400 dimensional feature.

6.4 Dataset Details And Experimental Design

Our dataset consists of two sets of handwriting from each of 104 writers. One set (training set) consist of exactly same piece of text from all writers. The other set (testing set) contains different text with varied number of words, from each writer. (Our training set comprises of 53 Bengali words, and in average our testing dataset consists of 50-60 words per writer). Both of the training and testing data set were scanned to 300 dpi in tiff file format. The printed text used by all writers for writing their respective training files are shown in Fig. (6.2). Character allograph obtained from respective training file of a writer are used in training our classifier for that particular writer.

We mainly performed our experiment to prove the following: (i) Reliability of our proposed scheme when dealing questioned document with relatively low data content. (ii) Robustness of our features to express discriminating characteristics of each individual writer. (iii) How system accuracy is affected when first two top choices and first three top choices are considered instead of only the top choice.

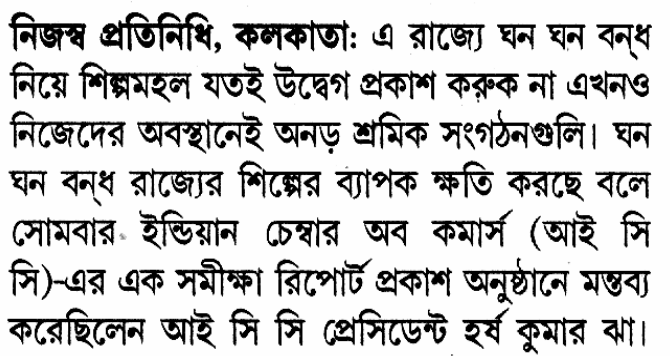
নিজস্ব প্রতিনিধি, কলকাতা: এ রাজ্যে ঘন ঘন বন্‌ধ
নিয়ে শিল্পমহল যতই উদ্বেগ প্রকাশ করুক না এখনও
নিজেদের অবস্থানেই অনড় শ্রমিক সংগঠনগুলি। ঘন
ঘন বন্‌ধ রাজ্যের শিল্পের ব্যাপক ক্ষতি করছে বলে
সোমবার ইন্ডিয়ান চেম্বার অব কমার্স (আই সি
সি)-এর এক সমীক্ষা রিপোর্ট প্রকাশ অনুষ্ঠানে মন্তব্য
করেছিলেন আই সি সি প্রেসিডেন্ট হর্ষ কুমার ঝা।

Figure 6.2: The text portion used in training file for all writers.

6.5 Classifier

We choose Support Vector Machine (SVM) as our classifier. The SVM looks for the optimal hyper-plane which maximizes the distance, the margin, between the nearest examples of both classes, named support vectors (SVs). In our experiments Gaussian kernel SVM outperformed other non-linear SVM kernels and linear SVM as well, hence we are reporting our recognition results based on Gaussian kernel only. The Gaussian kernel is of the form:

$$k(x,y) = exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \tag{6.1}$$

As mentioned earlier, in our experiment for 104 different writers, only about 53 words per writer were used for training. We got best optimized results when gamma parameter $\frac{1}{2\sigma^2}$ is set to 36.00 and 48.00 for directional feature and gradient feature, respectively. The

Table 6.1: WRITER IDENTIFICATION ACCURACY ON 104 TEST IMAGES.

Feature Used	Correctly identified	Misclassified
Directional Feature	94.23%	5.77%
Gradient Feature	95.19%	4.81%

penalty multiplier parameter is set to 20 for both feature type. Details of SVM can be found in [137] and [29]. Our problem is a multi-class problem, hence the 1:1 approach was used to solve the Multi-class svm.[†]

6.6 Results and Discussion

6.6.1 Writer identification accuracy

Here we show the writer identification accuracy of our scheme, after implementing majority voting technique for all character allograph present in a test image. In Table (6.1), we report accuracy of our system while using directional feature and gradient feature respectively. For evaluating each test image we did the following: (i) Let in a test image we get 80 character allographs by applying the method as discussed in Sub-section 6.2. (ii) We extract features from each of them and pass it to the classifier. (iii) The classifier decides the writer for each character allograph. (iv) Majority voting is performed amongst all classified character allographs. (v) Now, if amongst those 80 character allographs, writer 1 gets highest number of character allographs in its favor, we say that test image is written by writer 1. In case of a tie in majority voting we consider that as misclassification.

Results on Directional feature:- In our experiment with directional features, there were 6 misclassifications. We were unable to correctly identify the writer for test images from writer 41, 42, 49, 56, 71, and 83. For test image 41, 42 and 56 we noticed that the second top choice was the original writer whereas for test image 71 the third top choice was the original writer.

Results on Gradient feature:- In our experiment with gradient feature, there were 5 misclassifications. We were unable to correctly identify the writer for test images from writer 21, 41, 42, 56, and 71. Here also, for test image 41, 42 and 56 we noticed that the second top choice was the original writer whereas for test image 71 the third top choice was the original writer. We obtained an accuracy of 94.23% and 95.19% for directional and gradient features, respectively, even when training of our classifier was done with only 53 words per writer. This proves the reliability of our proposed system.

6.6.2 Distribution of Percentage of Identified Character Allographs Amongst Top Two Choices

Here we analyze the distribution of percentages of identified character allographs among the top two candidates of majority voting. This is done to give an idea of the differences between the top two choices. To illustrate, say in a test image there are 100 character allographs. Now suppose our classifier model assigns the highest number of character allographs (60) to writer "A" and second highest (20) to writer "B". So we can see that 60% of the total character allograph was assigned to the top choice. The very next second choice is

[†]Text changed/added.

having only 20% of the total character allograph, which is way behind the top choice. This distribution of character allographs between the top two choices is shown with the help of two different curves for all 104 test images. We noticed that, for all true hit points, the difference between the two curves is quite big. But for all false hit points, we observed a very small difference between the two curves. By a true (or false) hit point, we mean to say the test image with a correctly (or wrongly) identified writer. From this phenomenon it is clearly evident that for all true hit points there is a big difference of percentage of votes, between top two choices. See figure (6.3) and figure (6.4) for a pictorial illustrations of such distributions for directional and gradient features. Please note that in most of the false hit points the difference between the two curves is much less. It can be easily noted that the difference between the two curves in figure (6.4)is much more than the difference of curves in figure (6.3). Hence, we can conclude that our gradient-based features are more robust compared to our directional features.

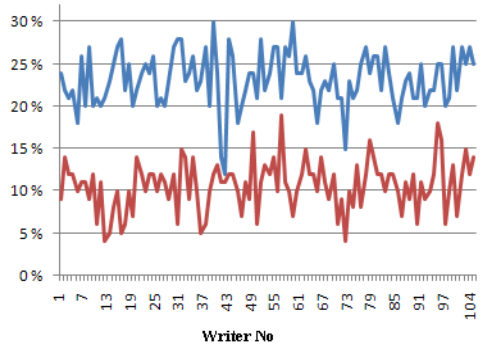

Figure 6.3: Distribution of percentage of character allograph between top two choices for directional feature. (Top blue curve-first choice), (Bottom red curve-second choice).

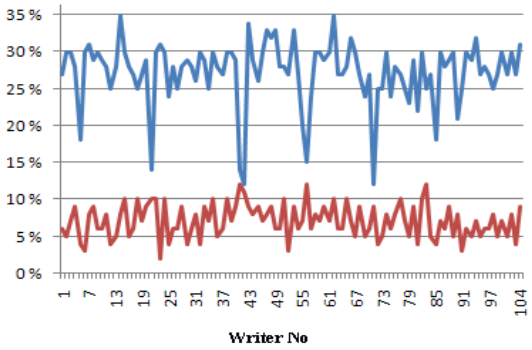

Figure 6.4: Distribution of percentage of character allograph between top two choices for gradient feature. (Top blue curve-first choice), (Bottom red curve-second choice)

6.6.3 Writer Identification Accuracy on Different Top Choices

Here we report the accuracy of our writer identification scheme when considering different choices of majority voting. It can be noted that we achieved 99.03% accuracy on Bengali

script with gradient features, when we considered the top three choices of our majority voting instead of the top one.

Table 6.2: Writer identification accuracy on different number of top choices of SVM.

No. of Top Choice	Directional Feature	Gradient Feature
1	94.23%	95.19%
2	97.11%	98.07%
3	98.07%	99.03%

6.6.4 Error Analysis

In Table 6.3, we report those writers whose test images are misclassified. We also mention corresponding features responsible for those misclassification. In the Table,'S' signifies that the particular writer was successfully identified by corresponding feature. We noticed that for most of those images there was a marginal win for the erroneous top choice. In most of those cases, after majority voting of character allograph, the original writer were either in the 2nd position or 3rd with very little difference from the top choice. We analyzed the reason and found that sometimes character allographs from two different classes were visually very similar.

6.6.5 Comparison with similar other works

Though there are many pieces of work on writer identification for non-Indic scripts, only one work [59] has been reported in the context of Indic script. Garain and Paquet [59] developed a writer identification system and evaluated their scheme on Roman and Bengali script. For Bengali script, they used a dataset of 40 writers, where each writer contributed two samples. One sample was used for training and other for testing. On an average, number of words in each of their sample was 200 or more. On Bengali script, they got 75% accuracy on first top choice amongst 40 writers. From our scheme, we obtained 95.19% accuracy when 104 writers are considered and number of word in each sample is much less (only about 50-60 words) compared to that of the number of words in samples of [59] (200 or more words in each sample).

Table 6.3: LIST OF MISCLASSIFIED WRITERS.

Original Writer	Misclassified as	
	Directional Feature	Gradient Feature
21	S	46
41	37	37
42	38	38
49	75	S
56	84	33
71	41	100
83	90	S

6.7 Conclusion

Here we propose a system for Bengali text independent writer identification using directional chain-code and gradient-based features. From the experiment on 104 writers we got promising results of 95.19% writer identification accuracy. We did not impose any rejection criteria in our classifier. We plan to do so in our future work.

Chapter 7

Text Independent Writer Identification for Oriya Script

Abstract-Automatic identification of an individual based on his/her handwriting characteristics is an important forensic tool. In a computational forensic scenario, presence of huge amount of text/information in a questioned document cannot be ensured. Lack of data threatens system reliability in such cases. We here propose a writer identification system for Oriya script which is capable of performing reasonably well even with small amount of text. Experiments with curvature feature are reported here, using Support Vector Machine (SVM) as classifier. We got promising results of 94.00% writer identification accuracy at first top choice and 99% when considering first three top choices.*

Keywords-: Writer Identification; Oriya Script; Curvature Feature; SVM.

7.1 Introduction

Writer identification utility could be an important tool in any computational forensic system. There are many pieces of work on writer identification [[27], [58], [93], [112], [114], [118], [119], [131], [132]]. Said et al. [112] developed a writer identification system which is text independent, they took a texture analysis based approach. Schomaker and Bulacu [118] proposed an offline writer identification system, using connected-component contours in uppercase handwritten samples. Later Bulacu and Schomaker [27] proposed texture level and allograph level feature-based writer identification scheme. Srihari et al. [132] have used a combination of global and local features. Though a large number of people in the world use Indic scripts, to the best of our knowledge, there are very few pieces of work on Indic scripts [[35], [59], [107]] in the context of writer identification. Garain et al. [59] proposed an AR coefficient feature-based writer identification system for 40 Bengali writers. They have used at least 200 words per writer for training and testing their system. In [35] a Gradient feature-based writer identification system is proposed for Bengali script which can perform well even when there are 50-70 words per writer. A writer identification system for Telegu script is proposed in [107]. In [107] the authors considered 5 samples from each of 22 writers; there they used structural information based features. In order to encounter adversary, like scarcity of data content in questioned documents Oriya script, we here propose a robust writer identification system.

7.2 Line And Character Segmentation

For line segmentation, at first, we divide the text into vertical stripes. Stripe width of a document is computed by statistical analysis of text height in the document [135]. Each of those stripes is processed to form Piece Wise Separating Lines (PSL) [135], and joining those PSLs we segmented the text lines. A histogram based approach was used to segment words in each text line. For word segmentation from a line, we compute vertical histogram

*Content of this chapter is mainly based on the article - "Text Independent Writer Identification for Oriya Script", Sukalpa Chanda, Katrin Franke and Umapada Pal, In Proc. 10th IAPR Document Analysis System, pp.369-273, DAS 2012.

of the line. In general the distance between two consecutive words of a line is bigger than the distance between two consecutive characters in a word. Taking the vertical histogram of the line and using a distance criteria [135] we segment words from lines.
In principle, when two or more characters in Oriya get connected one of the four following situations happens in most of the cases: (a) two consecutive characters create a large bottom reservoir; (b) the number of reservoirs and loops in a connected component will be greater than that of an isolated component; (c) two consecutive characters create a small top reservoir near mean line (d) the shape of the touching character will be more complex than isolated characters, (for details please see [135]). Computing different features obtained by the above observations we identify isolated and touching characters. If a component is detected as touching by the above algorithm then we segment the connected pattern to get its individual characters. For the segmentation of a touching pattern at first, the touching position is found. Next, based on the touching position, reservoir base-area points, topological and structural features the component is segmented to generate character allograph. Details about the method can be found in [135].

7.3 Feature Extraction

Oriya handwritten text is characterized by mainly round shaped characters/character allographs. But roundness in character allographs varies amongst different writers, even when they write the same text. Our curvature features are used as a descriptor to express this character allograph level dissimilarity present in the handwritings of different writers. Dimension of our curvature feature is 1176 which is later reduced to 392 using PCA.

7.3.1 Feature computation for Curvature feature

Curvature feature used in this paper has been calculated using bi-quadratic interpolation method as described in [125] and the procedure is as follows:
The curvature c at x_0 in a grey scale image is defined by

$$\text{c= y}^{''} \frac{}{\sqrt{(1+y^{'2})^3}} \tag{7.1}$$

where $y = g(x)$ is the equi-grey scale curve passing though x_0, $(x, y)$is the spatial co-ordinates of x_0 $y^{'}$ and $y^{''}$ are the first and second order derivative of y, respectively. The derivatives $y^{'}$ and $y^{''}$are derived from bi-quadratic interpolating surface for the grey scale values in the 8-neighbourhood of x_0 . The eight neighborhood of a pixel $(x_0)$is shown in figure (7.1). The pixel value of x_k is denoted by f_k. The bi-quadratic interpolated surface is given by

$$z = \begin{bmatrix} 1 & x & x^2 \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} 1 \\ y \\ y^2 \end{bmatrix} \tag{7.2}$$

Then the equi-grey curve passing through x_0 is given by

$$\begin{aligned} &(a_{22}x^2 + a_{12}x + a_{02})y^2 + (a_{21}x^2 + a_{11}x + a_{01})y \\ &+a_{02}x^2 + a_{10}x + a_{00} - f = 0 \end{aligned} \tag{7.3}$$

Differentiation of both sides of Eq. 7.3 by x we get

$$y' = \frac{\begin{aligned} &-\{(2a_{22}x + a_{12}x)y^2 + (2a_{21}x + a_{11})y + \\ &2a_{20}x + a_{10}x\} \end{aligned}}{\begin{aligned} &2y(a_{22}x^2 + a_{12}x + a_{02}) + a_{21}x^2 + a_{11}x \\ &+a_{01} \end{aligned}} \tag{7.4}$$

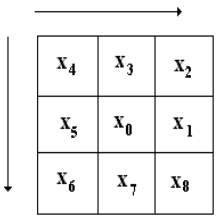

Figure 7.1: Neighbourhood of a pixel, x_o.

Substituting the co-ordinates (0,0) of x_0 to 7.4, the value of $y^{'}$ at x_0 is given by

$$y^{'} = -a_{10}/a_{01} \tag{7.5}$$

Similarly, the value of $y^{''}$at x_0 is given by

$$y^{''} = -2(a_{10}^2 a_{02} - a_{01}a_{10}a_{11} + a_{01}^2 a_{20})/a_{01}^3 \tag{7.6}$$

Solving the simultaneous linear equations 7.2 holding for 8-neighbour of x_0, the coefficients of the bi-quadratic surface are given by

$a_{10} = (f_1 - f_5)/2, a_{20} = (f_1 + f_5 - 2f_0)/2,$
$a_{01} = (f_3 - f_7)/2, a_{02} = (f_3 + f_7 - 2f_0)/2, a_{11} = (f_2 - f_8) - (f_4 - f_6)/4,$
The coefficients a_{10} and $a_{20}$are respectively, the first and the second order partial derivatives of $f(x, y)$with respect to x, $a_{01}$and a_{02} are similar partial derivatives with respect to y, and $a_{11}$is the derivative obtained with respect to x and y. Substituting Eqs. (7.5) and (7.6) to Eq. (7.1), the curvature is given by

$$c = -2(a_{10}^2 a_{02} - a_{01}a_{10}a_{11} + a_{01}^2 a_{20})/(a_{10}^2 + a_{01}^2)^{\frac{3}{2}} \tag{7.7}$$

By definition 7.7, curvature is indefinite if $a_{10} = a_{01} = 0$. When such situation occurs then we assume the curvature is zero in our algorithm.

To get the curvature feature the following steps are applied.

Step 1: The direction of gradient is quantized to 32 levels with $\pi/16$ intervals.

Step 2: The curvature $'C'$ computed by the mentioned formula in 7.8 is quantized into 3 levels using a threshold (t)(for concave, linear and convex regions). For concave region $c \leq -t$, for linear region $(-t < c < t)$ and for convex region $c \geq t$. We assume t as 0.12 in our experiment.

Step 3: The strength of the gradient is accumulated in each of the 32 directions and in each of the 3 curvatures levels of each block to get 49x49 local joint spectra of directions and curvatures.

Step 4: A spatial and directional resolution is made as follows. A smoothing filter [1 4 6 4 1] is used to get 16 directions from 32 directions. On this resultant image, another smoothing filter [1 2 1] is used to get 8 directions from 16 directions. Further more, we use a 31 x 31 two-dimensional Gaussian-like filter (See figure (7.2)) to get smoothed 7 x 7 blocks from 49 x 49 blocks (shown in figure (7.3)). So, we get 7x7x8 = 392 dimensional feature vector. Using curvature feature in 3 levels we get 392x3 =1176 dimensional features.

Step 5: Using principal component analysis we reduce this 1176 dimensional feature vector to a 392 dimensional feature vector and we fed this 392 dimensional feature vector to our classifier.

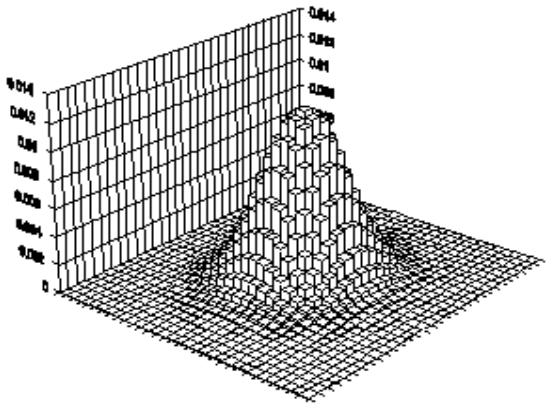

Figure 7.2: Example 31 x 31 two-dimensional Gaussian-like filter used for smoothing.

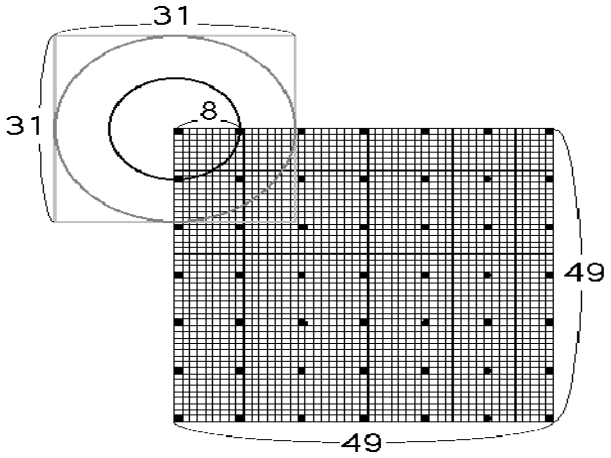

Figure 7.3: Illustration of getting 7 x 7 blocks from 49 x 49 blocks.

7.4 Classifier And Experimental Design

We choose Support Vector Machine (SVM) as our classifier. The SVM looks for the optimal hyper-plane which maximizes the distance, the margin, between the nearest examples of both classes, named support vectors (SVs). In our experiments Gaussian kernel SVM outperformed other non-linear SVM kernels and linear SVM as well, hence we are reporting our recognition results based on Gaussian kernel only. The Gaussian kernel is of the form:

$$k(x,y) = exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \tag{7.8}$$

As mentioned earlier, in our experiment for 100 different writers, in an average about 60-80 words per writer were used for training. We got best optimized results when gamma parameter $\frac{1}{2\sigma^2}$ is set to 0.05. The penalty multiplier parameter is set to 1. Details of SVM can be found in [137],[29].

For evaluating each test image we did the following: (i) Let in a test image we get N character allograph by applying the method as discussed in Section 7.2. (ii) We extract fea-

tures from each of them and pass it to the classifier.(iii)The classifier decides the writer for each character allograph. (iv)Majority voting is performed amongst all classified character allograph.(v)If amongst those N character allographs, writer 1 gets highest number of character allograph in its favor, we say that test image is written by writer 1. In case of a tie in majority voting we consider that as a rejection.

7.5 Dataset Details

Our dataset consists of two sets of handwriting from each of 100 writers. One set is used for training and other set for testing. Both set contains different text with varied number of words, from each writer. Our training (testing) dataset comprises of 60-80 (60-80) Oriya words in average. All data were scanned to 300 dpi in tiff file format. We mainly performed our experiment to investigate the following: (i) Robustness of curvature features to express discriminating characteristics of each individual writer. (ii) To identify dissimilar character allograph shapes amongst 100 different writers.(iii) To identify most similar character allograph shapes amongst 100 different writers.

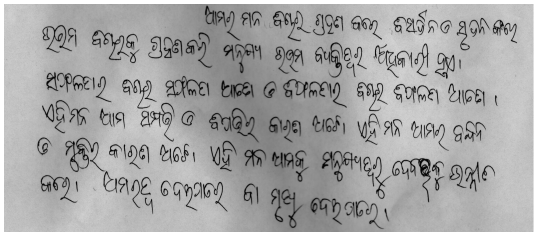

Figure 7.4: Example of a test file from one of our writers.

7.6 Results and Discussion

7.6.1 Writer Identification Accuracy

Here we show the writer identification accuracy of our scheme, after implementing majority voting technique for all character allograph present in a test image. In Table I, we report accuracy of our system using curvature feature. From the experiment on 100 writers, there were 5 misclassifications and one rejection.

7.6.2 Most dissimilar character allographs amongst 100 writers

Here we tried to investigate some character allograph shapes which actually help us in differentiating our 100 writers. We identified such character allograph based on two criteria, (i) We considered character allograph with a high confidence score [142] (with confidence score of at least 0.7) that were assigned to the right writer, we call them C_{hp} (ii) By looking for most frequent character shapes, those were correctly assigned to the right writers. We call them C_{cs} . We can conclude that those C_{hp} and C_{cs} largely contributed in discriminating Oriya writers. We noted that C_{hp} type character allograph not necessarily belongs to C_{cs} type character allograph or vice-versa. There was a lot of C_{cs} type character allograph those had a top choice confidence score of even less than 0.5. It is not mandatory that character allograph shapes once identified as C_{hp} type character allograph always had a high confidence score value. In figure 7.5 we show few examples of common character allograph shapes that helped us in discriminating amongst 100 writers.

Figure 7.5: Some common character allograph shapes which highly contributed in discriminating different writers.

7.6.3 Characteristics of similar shaped character allographs amongst 100 writers

We were also interested to find out characteristics of similar shaped character allograph generated by 100 writers that actually reduces our system accuracy. We can conclude that those shapes were most difficult to be assigned to the correct class/writers. We noticed that majority of character allograph those got wrongly assigned to incorrect writers, had a very low confidence score on the top choice. Figure 7.6 is a graph which shows the distribution of confidence score value amongst all such character allograph those were wrongly assigned to incorrect writers. We can see about 60% of wrongly assigned character allographs had a confidence score of 0.3 or less, whereas about only 5% of those wrongly assigned character allographs obtained a confidence score of 0.6 or more. In the error analysis section we will see why those few erroneously assigned character allographs had such high confidence score value.

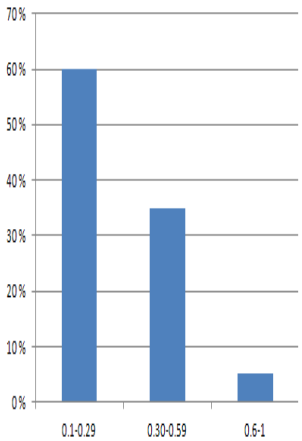

Figure 7.6: Confidence score distribution amongst erroneously classified character allograph.

7.6.4 Writer Identification Accuracy on Different Top Choices

Here we report the accuracy of our writer identification scheme when considering different choices of majority voting. It can be noted that we achieved 99.00% accuracy on with curvature feature, when we consider top three choices of our majority voting instead of the top one.

Table 7.1: WRITER IDENTIFICATION ACCURACY ON DIFFERENT NUMBER OF TOP CHOICES OF SVM.

No. of Top Choice	Accuracy
1	94.00%
2	97.00%
3	99.00%

7.6.5 Error Analysis

We considered those test images which are misclassified for further analysis. We noticed that for most of those images there was a marginal win for the erroneous top choice. In most of those cases, after majority voting of character allographs, the original writer were either in the 2nd or 3rd position. We analyzed the reason and found that character allographs were assigned to wrong classes due to mainly two reasons, (i) Sometimes character allograph from two different writers were visually very similar. (ii) Deformed character allographs was formed due to erroneous segmentation, where three or more character/character allograph forms a single character-component. With the help of figure 7.7 we show examples where two different writers produce very similar character allograph. Here the left character allographs in figure 7.7 is from the test image of writer 15 and were assigned to writer 17 with a confidence score of 0.8. We were surprised to see such high confidence score on erroneous classifications. We analyzed all the character allograph generated from the training file of writer 15 and writer 17. Unfortunately there were no similar character allographs in the training file of writer 15. But in the training file of writer 17 we found very similar characters (for an example please look into right hand character allographs in figure 7.7). So we can conclude that if our training process encounters character allograph of very similar shapes from different writers, character allograph of those writers might get misclassified during testing with high confidence score[†].

7.6.6 Comparison with similar other works

Though there are many pieces of work on writer identification for non-Indic scripts, only few pieces of work [[59], [35], [107]] have been reported in the context of Indic scripts. Garain and Paquet [59] developed a writer identification system and evaluated their scheme on Roman and Bengali script. For Bengali script, they used a dataset of 40 writers, where each writer contributed two samples. One sample was used for training and other for testing. On an average, number of words in each of their sample was 200 or more. On Bengali script, they got 75% accuracy on first top choice amongst 40 writers. Another work [35] on same Bengali script reports an accuracy of 95.19% with 104 writers in a constrained environment of only 50-60 words per writer. There gradient features along with SVM classifier were used. A system for Telegu text independent writer identification is proposed in [107]. They have achieved an accuracy of 98% but they have considered only 22 writers. Here we have considered 100 writers and obtained an accuracy of 94% considering only the top choice of our classifier.

[†]Sentence changed.

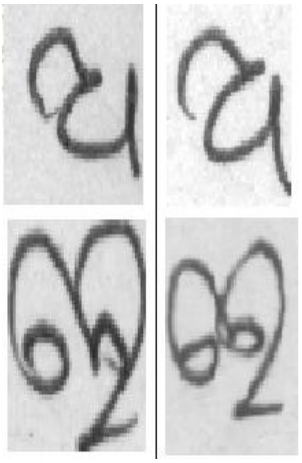

Figure 7.7: Four visually similar shaped character allograph: (left) from test image of writer 15, (right) from training image of writer 17.

7.7 Conclusion

Here we propose a system for Oriya text independent writer identification using directional chain-code and curvature-based features. From the experiment on 100 writers we got promising results of 94% writer identification accuracy. In future we would like to implement a L1-norm based Multiple Kernel SVM for its inherent feature selection/dimensionality reduction and classification capability, and compare with our present technique.

Chapter 8

Identification of Indic Scripts on Torn-Documents

Abstract- Questioned Document Examination processes often encompass analysis of torn documents. To aid a forensic expert, automatic classification of content type in torn documents might be useful. This helps a forensic expert to sort out similar document fragments from a pile of torn documents. One parameter of similarity could be the script of the text. In this chapter we propose a method to identify the script in document fragments. Torn documents are normaly charecterized by text with arbitrary orientation. We use Zernike moment based features that are rotation invariant together with Support Vector Machine (SVM) to classify the script type. Subsequently gradient features are used for comparative analysis of results between rotation dependent and rotation invariant feature type. We achieved an overall script-identification accuracy of 81.39% when dealing with 11 different scripts at character/connected-component level and 94.65% at word level. *

Keywords- Script Identification; Torn Document; Gaussian Kernel SVM; Computational Forensics.

8.1 Introduction

Questioned-document examination process often requires to analyze a heap of torn documents. In such cases an automated system can sort out similar document fragments, and narrow down the search space of a forensic expert. A notion of similarity between two document fragments could be the script present in those two documents. Script Identification technique can be used to sort out similar document fragments which might come from the same source. This can be used as a criterion for linking two or more different document fragments to a document page and/or same source of origin. Lot of research has been done on script identification already. Yet the present state of the art is in-sufficient to address the challenges of script identification in context of document fragments. The adversaries involved in script identification on torn documents are as follows: (i) Scarcity of text/data content. Please note that all images of Fig. (8.1) consist of very few text/words. (ii) Multiple orientation of text. (iii) Arbitrary background type for document fragments. In this article we intend to propose a script identification scheme based on Zernike moments/Gradient features for torn document fragments, which could be used as a part of an automatic questioned document examination system in context of Indian scripts. A brief review of some published research work on script identification is given in the following paragraph.
Among the pieces of earlier work on script identification, Spitz[129] developed a method to separate Han-based or Latin-based script. He used optical density distribution of characters and frequently occurring word shape characteristics. Jaeger, Ma, and Doermann [72] used KNN, SVM, weighted Euclidean distance, and Gaussian mixture model to identify English from Arabic, Chinese, Korean and Hindi scripts using cluster-based templates. An automatic script identification technique has been described by Hochberg, Kelly, Thomas

*Content of this chapter is based on the article - "Identification of Indic Scripts on Torn-Documents", Sukalpa Chanda, Katrin Franke and Umapada Pal, In Proc. 11th International Conference on Document Analysis and Recognition, pp. 713-717, ICDAR 2011.

and Kerns [69]. Using fractal-based texture features, Tan [134] described an automatic method for identification of Chinese, English, Greek, Russian, Malayalam and Persian text. All the above mentioned works deal with non-Indian script. Among Indian script, there are some related works. Pal, Sinha and Chaudhuri [102] proposed a generalized scheme for line-wise script identification from a single document containing twelve Indian scripts. Sinha, Pal and Chaudhuri [126] proposed a word-wise script identification scheme with a combination of Indian languages. Dhanya and Ramakrishnan [48] mentioned a Gabor-filter-based technique for word-wise segmentation from bi-lingual documents containing English and Tamil scripts; they have used classifiers like LSVM and K-NN. Patil and Subareddy [104] proposed a neural-network-based system for word-wise identification of English, Hindi and Kannada language scripts. Zhou, Lu and Tan [145] proposed a Bengali/English script-identification scheme using connected component analysis. In this paper a technique for script identification in torn documents is proposed that consist of Roman and all major Indic scripts. In the proposed method a study of two different feature types are conducted. A comparative analysis between a rotation invariant feature type and rotation dependent feature type is performed. For both cases, feature extraction is performed on each connected component/charactercomponent found in a document fragment. Later the extracted features were passed to a Support Vector Machine (SVM) classifier. Classification of a document fragment (words) to a particular script is done based on majority voting of each recognized character component of the document. Classification results are reported at three different levels: (i) connected component/character component level (ii) group of character-components /word level (iii) entire document fragment level.

Figure 8.1: Torn documents consisting of text in some Indic scripts in multiple orientations.

8.2 Methodology

Script identification for the whole torn document is dependent on a hierarchy based classification. At the bottom most level of this hierarchy is the connected component/character-component. In the middle level are words, which are formed by groups of character component/connected-component. At the highest level the whole torn document which is formed by a collection of words. Classification of connected/character-component present in a torn document is done first. Based on majority voting of script amongst character-component present in a word, we decide the script of the word. Finally considering majority voting amongst script of all words present in the torn document, we decide the script

type of the document. We noticed that arbitary orientation of text makes it difficult to define a 'word' for most of the scripts. In few scripts like Devnagari and Bengali the presence of a Head-line connects all characters together (See Fig. 8.2 for illustration). In such cases defining a word is easy due to the presence of head-line even with arbitrary orientation. But for Roman and some other Indic-scripts like 'Oriya', 'Tamil', 'Telegu' etc; characters sit beside each other in same horizontal line and form a word. In an arbitary orientation scenario this wont be possible as they are not always in a horizontal line. For e.g. consider the Oriya word in Fig. 8.2. As a consequence we need to deploy some pre-processing techniques, which will help us to define a group of character-components as a word. One might argue that we can directly apply any rotation invariant feature extraction scheme on the character-components, and based on majority voting on all character-components present in a torn document, we can decide the script type for the document. This wont work always in Indian subcontinent scenario. Being a multilingual country its very common in India to have multiple scripts in a single document page. It is quite possible that two scripts simultaneously occur in a document fragment of that document page. As a result we also need to identify the script at word level in a torn document. Then it can be used for some questioned document examination process in Indian sub-continent. To form word boundary in case of arbitrary oriented character-components for scripts like Oriya, Tamil, Telegu etc., we used a mathematical morphology based dilation operator. Details of our method are presented in Section 8.3.

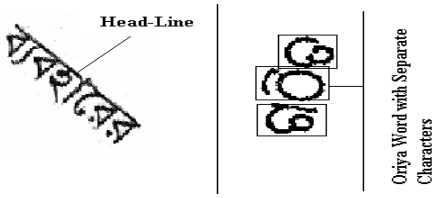

Figure 8.2: Word in a Bengali and Oriya script in an arbitrary orientation scenario.

8.3 Pre-processing

All input grayscale images (scanned with a resolution of 300 dpi) are converted to their binary equivalents. Next we perform the following tasks: (i) Perform character-component bloating. (This is done with the help of 'dilation', applying the five-times-dilation operation of mathematical morphology. The structuring element for the morphology is of size 7x7. See figure 8.3 for an illustration of dilation); (ii) It can be noted that the original input image consists of a word of three separate characters, using our dilation technique we fused them to get a word boundary.(iii) Perform connected component analysis and word-labeling. (iv) For each labelled word component, determine its direction of highest variation (extension) by implementing principal-component analysis (PCA) discussed in [120]. (v) Rotate each word component according to the direction of its first eigenvector. (Tasks iv and v are done solely to implement our rotation-dependent feature extraction method).

8.3.1 Pre-processing and Working Methodology for Zernike Moment Feature

Let $'B'_W$ and $'B'_C$ be the same copies of the binary image. In one binary image $'B'_W$, a region growing operation is performed on all connected components present in the image $'B'_W$. As a result, characters close to each other fuse and form a much bigger connected component which we can term as 'word-component'(See figure (8.3)b). Now a component labelling in $'B'_W$ gives us our desired word boundary in $'B'_W$. Other copy of the binary image $'B'_C$ is not processed by the region growing operation. We map the word boundaries

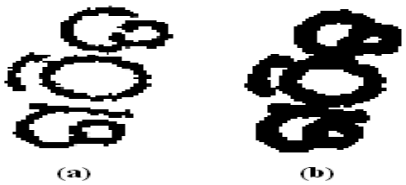

Figure 8.3: (Left)Oriya word with separate character, (Right) corresponding output after applying region growing method.

obtained from $'B'_W$ on the image $'B'_C$. We also calculate the average length (A_L) and width (A_W) of all character-components found in $'B'_C$. Considering the word boundary, we do component labelling inside a binary image $'B'_C$, (to get individual character-components present within the word boundary). Then each character-component with length $>= A_L$ and width $>= A_W$ is normalised to a square matrix. The size of this normalised matrix is considered with respect to Max (A_L, A_W). Zernike moment-based features are extracted from each of those size-normalised character-components and passed to a classifier. Classifier decides the script type for each character-component of the word.

8.3.2 Pre-processing and Working Methodology for Gradient Feature

Gradient features are not rotation invariant as Zernike moment-based features. So for extracting gradient features from text, we need to fix the orientation of the text. As mentioned earlier, after region growing operation characters in a word generally touch each other as shown in Fig. 8.3b and a word become a single connected component (word-component). We perform wordcomponent labeling in $'B'_W$. Inside the word component in $'B'_W$, we calculate the distribution of black pixels. A PCA-based method is deployed to detect the orientation of the word. Details about it can be found in [34]. Word component labeling in $'B'_W$ helps us to get word boundary from the binary image $'B'_C$ which is not processed by region growing operation. We copy the word area from the $'B'_C$ and fix the orientation of the word. After orientation of the word is fixed, a component labeling is performed within this word image. Individual character-components are processed for gradient feature extraction and extracted feature vector is passed on to the classifier. Classifier decides the script type for each character-component found in the word.

8.4 Feature Extraction

Considering the possible arbitrary orientation of text present in a torn document, we looked for a feature extraction scheme which is rotation and scale invariant. As a consequence we initially experimented with various moment-based features like Hu, Zernike etc.. We got best results with Zernike and we made all further experiments using Zernike moments-based features. Even though we got encouraging results with Zernike moments, we were curious to compare its efficacy in comparison to a rotation dependent feature extraction method. We had prior experience of fixing orientation of text present in such torn documents [34]. We used a similar morphology and PCA based approach to deduce the orientation of the text, thereafter rotating the piece of text to normal orientation we extracted gradient-based features. In the following two sub-sections we will narrate the feature extraction methodology used to obtain our Zernike moment-based features and Gradient features.

8.4.1 Zernike Moment Feature

Zernike moment features are rotation invariant in nature. Two dimensional Zernike moment can be computed using the formula:

$$A_{mn} = \frac{m+1}{\pi} \int_x \int_y f(x,y)[V_{mn}]^* dxdy$$

Where $x^2 + y^2 \leq 1$ and $m - n$ = even, $n \leq 1$
Here m = 0, 1, 2...∞ defines the order and $f(x,y)$ is the function being described and * denotes the complex conjugate. n is an integer implying the angular dependence. For a discrete image pixel $P(x,y)$, the integrals are changed to summation, and the above equation gets transformed to the following:

$$A_{mn} = \frac{m+1}{\pi} \sum_x \sum_y f(x,y)[V_{mn}]^* dxdy$$

$x^2 + y^2 \leq 1$

For our case the idea is to map the image of the size-normalized character-components to the unit disc using polar coordinates, where the centre of the image is the origin of the unit disc. Those pixels falling outside the unit disc are not used in our computation. We got best results with m=13 when basis functions with negative repetition are included, giving us a feature vector of dimension 105. Details about the feature can be found in [80].

8.4.2 Gradient Feature

To obtain 400-dimensional gradient features [101] we apply the following steps. (i) The input binary image is converted into a gray-scale image applying a 2 $\times$ 2 mean filtering 5 times. (ii) The resultant gray-scale image is then normalized so that the mean gray scale becomes zero with maximum value 1. (iii) Normalized image is now segmented into 9 $\times$ 9 blocks. (iv) A Roberts filter is applied on the image to obtain gradient image. The arc tangent of the gradient (direction of gradient) is quantized into 16 directions and the strength of the gradient is accumulated with each of the quantized directions. By strength of Gradient $f(x,y)$ we mean $f(x,y) = \sqrt{\Delta(v)^2 + \Delta(u)^2}$ and by direction of gradient $(\theta(x,y))$ we mean $\theta(x,y) = \tan^{-1} \frac{\Delta(v)}{\Delta(u)}$ where $\Delta u = g(x+1,y+1) - g(x,y)$ and $\Delta v = g(x+1,y) - g(x,y+1)$ and $g(x,y)$ is a gray scale at (x, y) point.(v) Histograms of the values of 16 quantized directions are computed in each of 9 $\times$ 9 blocks. (vi) 9 $\times$ 9 blocks are finally down sampled into 5 $\times$ 5 by a Gaussian filter. Thus, we get 5 $\times$ 5 $\times$ 16 = 400-dimensional feature vector.

8.5 Classifier

In our experiments, we have used a Support Vector Machine (SVM) as classifier. The Support Vector Machine (SVM) is defined for two-class problem and it looks for the optimal hyper-plane which maximizes the distance, the margin, between the nearest examples of both classes, named support vectors (SVs). Given a training database of M data: $\{x_m \mid m = 1, \cdots, M\}$ the linear SVM classifier is then defined as:

$$f(x) = \sum_j \alpha_j x_j \cdot x + b$$

where $\{x_j\}$ are the set of support vectors and the parameters α_j and b has been determined by solving a quadratic problem [29]. The linear SVM can be extended to various non-linear form, and details can be found in [137][29]. In our experiments we noted Gaussian kernel

SVM outperformed other non-linear SVM kernels, hence we are reporting our recognition results based on Gaussian kernel only. The Gaussian kernel is of the form:

$$k(x, y) = exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right) \tag{8.1}$$

For Zernike moment-based features, Gaussian Kernel gave highest accuracy when $\frac{1}{2\sigma^2}$ is set to 0.00006 and penalty multiplier is 3. Due to such low value of $\frac{1}{2\sigma^2}$ we can conclude that the Zernike moment-based features generated for our 11 class problem makes classification more linear in nature. For gradient features we noticed that Gaussian kernel gave highest accuracy when $\frac{1}{2\sigma^2}$ is set to 36.00 and the penalty multiplier is 3. The high value of $\frac{1}{2\sigma^2}$ for classification with gradient feature indicates that more non-linearity is involved in classification task with gradient feature.

8.6 Experimental Setup, Results And Discussions

We evaluated efficacy of both feature types separately. Classification accuracy for both feature types are recorded for all three following levels:(i) character-component (Level I) (ii) word/group of character-component, (Level II), and (iii) entire document fragment, (Level III). A brief view on our dataset can be found in sub-section 8.6.1. In subsection 8.6.2 we present the accuracy for all 11 classes of scripts (Bengali class-1 , Devnagari class-2 , Roman class-3 , Oriya class-4 , Gurumukhi class-5 , Gujarati class-6 , Telegu class-7 , Tamil class-8 Kannada class-9 , Malayalam class-10 and Urdu class-11) at character/connected-component level in a graphical format. The accuracies on other two levels (word and document) are reported in two sub-sections 8.6.3 and 8.6.4 respectively. Later in sub-section 8.6.5 we also present a confidence score distribution for both types of feature. By confidence score value of recognition we mean the probability estimation of the recognized class [142].

8.6.1 Dataset Details

To the best of our knowledge, there is no publicly available database suitable for our defined problem (torn documents with Indic scripts). We developed our own dataset to evaluate our proposed method. All document fragments were scanned with 300 dpi resolution. Utmost care is taken to ensure the presence of adversaries normally found in any torn document. Training dataset consist of 112 torn documents. The test dataset consists of 130 torn documents. From training and test dataset we obtained 7281 and 8130 connected/character-components, respectively. We considered 11 different scripts comprising of Bengali (Bangla), Devnagari, Oriya, Urdu, Malayalam, Gujarati, Telegu, Tamil, Kannada, Gurumukhi, and Roman. Normally a torn document with printed text will have similar orientation for all text present in the document. But to make our problem more challenging we intentionally prepared torn documents with multiple text orientation. Amongst all test images there were 10 document fragment images having text from multiple scripts. Those 10 document fragments were not considered during experimentation for document level script identification.

8.6.2 Accuracy at Connected/Character-Component level

At the character-component stage we calculated our accuracy in two different experimental setups for both types of features. (a) A five-fold cross-validation on the character-components of all scripts found in entire training dataset. (b) First training using entire training dataset and then classifying each character-component found in all test torn document images. Below is the graph, where we depict accuracy of our scheme when applied on test dataset for both feature types. It can be noted that the Gradient-based feature

slightly outperformed the Zernike moment-based features for every class. On our test dataset, the average accuracy at character-component/connected-component level with Zernike moment-based features were 71.03% while with Gradient-based features it was 81.13%. On our training dataset at character-component/connected-component level, a 5-fold cross validation gave an accuracy of 71.33% with Zernike moment-based features and 81.39% with gradient-based features.

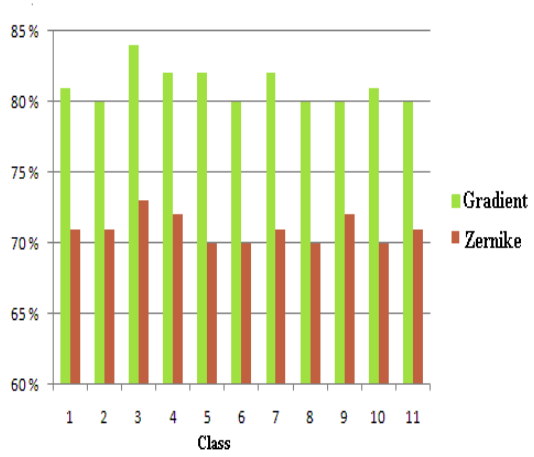

Figure 8.4: Percentage of accuracy at character-component/connected component level for both types of features.

8.6.3 Accuracy at word level

At word level the accuracy is calculated for all test images as follows: (a) Feature extraction is performed on each character-component found in a word.(b) Classification of each character-component is done.(c) Based on majority voting amongst all classified character-component the script type of the word is decided. (d) In case of a tie, we sum the respective confidence score of all recognized script types separately. The word is classified to the script type with maximum confidence score sum. We got an average accuracy of 94.65% with gradient features and 85.30% with Zernike features at word level. By average accuracy we mean to say the cumulative percentage accuracy of all scripts divided by 11 (the number of scripts used in our experiment).

8.6.4 Accuracy at document level

To calculate script identification accuracy at document level we only considered our test documents consisting of single script. At first the script of each word present in a test image was identified. Then based on majority voting amongst script type of words we conclude the script type for the document. In case of a tie, we consider it as a rejection. We obtained an accuracy of 96.7% and 98.33% at document level for Zernike moment and gradient features respectively with 0% rejection.

8.6.5 Confidence score distribution

Here we illustrate the distribution of confidence score of top-choice returned by our classifier, for both feature types. By confidence score, we mean to say the probability estimation of the recognized class [142]. The scores are taken during classification. We noticed that

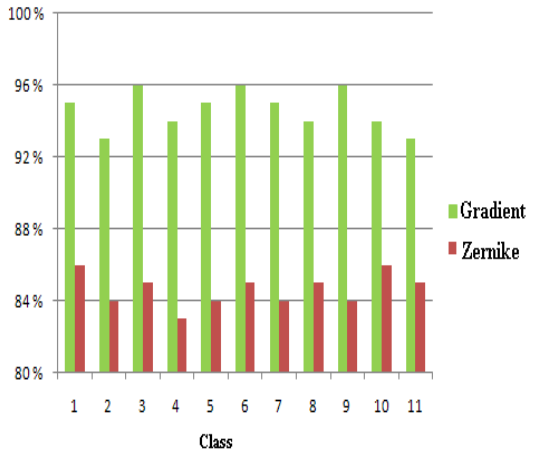

Figure 8.5: Accuracy at word level for both types of features.

majority of correct classification with Zernike features gave a confidence score in the range of 0.6-0.69 while with gradient features it is in the range of 0.8-0.89.

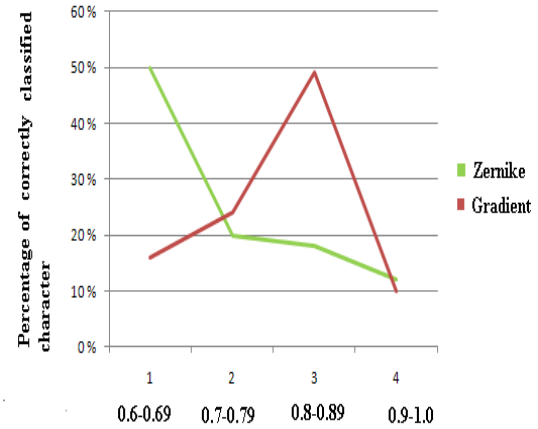

Figure 8.6: Distribution of confidence score in correct classification for both feature types.

8.7 Error Analysis

We analyzed the errors for both feature types. We noticed that for Zernike moment-based features, noisy images gave poor results. With gradient features, errors came mostly due to wrong orientation detection of text. This happened mostly with words when number of character component $<= 2$ and also with text found in the edge of torn documents. We noticed that most miss-classification occurred between Gujarati and Devnagari scripts. The reason is Gujarati characters looks very similar to a Devnagari character without headline on top.

8.8 Conclusion

In this article we proposed a scheme to identify 11 different scripts present in torn documents. We encountered adversaries like arbitrary orientation of text, scarcity of text in torn documents and got encouraging results using rotation dependent and independent features.

Chapter 9

Script Identification A Han & Roman Script Perspective

Abstract-All Han-based scripts (Chinese, Japanese, and Korean) possess similar visual characteristics.Hence system development for identification of Chinese, Japanese and Korean scripts from a single document page is quite challenging. It is noted that a Han-based document page might also have Roman script in them. A multi-script OCR system dealing with Chinese, Japanese, Korean, and Roman scripts, demands identification of scripts before execution of respective OCR modules. We propose a system to address this problem using directional features along with a Gaussian Kernel-based Support Vector Machine. We got promising results of 98.39% script identification accuracy at character level and 99.85% at block level, when no rejection was considered.*

Keywords:- Script Identification; Multi-script OCR; Document Analysis; SVM.

9.1 Introduction

Script Identification is a necessary pre-processing step for any multi-script OCR system. There are many pieces of work on script identification [[129][69][71][134][72][103][96][111]]. Using cluster based templates an automatic script identification technique has been described by Hochberg et al. [69]. Tan [134] described an automatic method for identification of Chinese, English, Greek, Russian, Malayalam, and Persian text using texture-based features. Jaeger et al. [72] used K-NN, SVM, weighted Euclidean distance, and Gaussian mixture model to identify English from Arabic, Chinese, Korean, and Hindi scripts. There are some pieces of work on Indic scripts also [96]. But surprisingly there are very few works on Han-based script identification [[129],[71], and [103]]. Among them, Spitz [129] developed a method for Han-based or Latin-based script separation. He used optical density distribution of characters and frequently occurring word shape characteristics for the purpose. Pan et al. [103] proposed a scheme to discriminate Chinese, Japanese, Korean, and Roman scripts using Gabor filter-based features. They extracted features on a 256x256 pixel window and used an ANN classifier for classification task. But in real life scenario, this approach could face some problems when we have characters from two different scripts within this window area. Hence a more logical approach would have been to identify script at a single character level. Here we propose a system to identify Chinese, Japanese, Korean and Roman script using chain-code histogram based directional features. The rest of the paper is arranged as follows. In Section 9.2, we describe our simple line and character segmentation process. In Section 9.3, we describe our features along with a logical explanation of its utility for our objective. We describe our experimental setup with some information on our dataset in Section 9.4. In Section 9.5, we give a brief discussion on our classifier. Result and discussions on various experiments are reported in Section 9.6.

*Content of this chapter is mainly based on the article - "Script Identification -A Han and Roman Script Perspective", Sukalpa Chanda, Umapada Pal, Katrin Franke and Fumitaka Kimura, In Proc. 20th International Conference on Pattern Recognition, pp.2708-2711, ICPR 2010.

9.2 Line And Character Segmentation

All document images are binarized and de-skewed before they are† processed for possible line and character segmentation [36]. The lines are segmented in the documents by finding the valleys of the histogram computed by counting the number of black pixels in each row. Then for each line, a vertical scanning of columns is executed and character segmentation is performed by following a technique as described in [36]. In few cases we observed wrong character segmentation due to presence of unwanted noise in the document image.

Figure 9.1: Line and Character segmentation results on Japanese (Top left), Roman (Bottom left), Chinese (Top right) and Korean (Bottom right) documents are shown.

9.3 Chain-code Histogram-Based Feature Extraction

During our initial experiments, we tried to extract Gabor filter-based features from each individual character. But we noticed that Gabor filter-based features were not capable enough to express dissimilarity at character level among our concerned script types. Directional features works as local descriptors of shapes in a character component. Though Roman characters are quite different in shape from any Han script-based character, but in Chinese, Japanese and Korean characters, shape dissimilarity is only found in certain small portions of those characters. Our chain-code histogram based directional feature is capable of expressing these tiny local/regional dissimilarities present in the characters of those three scripts. Feature computation is done as follows: At first we compute the bounding box of a character component. This bounding box is then divided into 7x7 blocks [101]. In each of these blocks the direction chain code for each contour point is noted and frequency of direction codes is computed. Here we use chain code of four directions only: directions 1 (horizontal); 2 (45 degree slanted); 3 (vertical) and 4 (135 degree slanted). See Fig.9.2 for illustration of four chain-code directions. We assume chain code of direction 1 and 5 are same. Also, we assume direction 2 and 6, 3 and 7, 4 and 8 are equivalent, because if we traverse from point 4 to point 8 we will have the same count as point 8 to point 4. Subsequently, in each block, we get an array of four integer values representing the frequencies of chain code in these four directions. These frequencies are used as feature. Thus, for 7x7 blocks we get 7x7x4=196 features. In order to reduce the feature dimension, after the histogram calculation in 7x7 blocks, the blocks are down sampled with Gaussian filter into 4x4 blocks. As a result we obtain 4x4x4=64 features for further classification. To normalize the features we determine the maximum value of the histograms from all the blocks and divide each of the above features by this maximum value to get the feature values between 0 and 1. A more detailed description about the feature can be found in [101].

†Text changed/added.

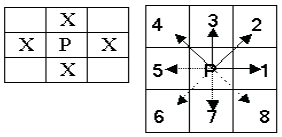

Figure 9.2: For a point "P" the direction code of its eight neighboring points is shown.

9.4 Dataset Details And Experimental Design

Our entire dataset comprises of 30 document images from each script. They were divided equally into training and testing set. Every image in our training and testing dataset were scanned to 300 dpi in tiff file format. From images of our training set we obtained 1750 Chinese, 1839 Japanese, 2130 Korean and 2018 Roman characters. From images in our test dataset, we obtained 3200 Chinese, 3415 Japanese, 2952 Korean and 3500 Roman characters. To get about the ideas of character size we also compute average height of characters. We noticed that average height of characters was from (25-40) pixels.
We mainly performed 3 different experiments. Which are as follows: (i) Character level script identification based on five-fold cross validation on the training dataset. (ii) Character level script identification based on training the classifier with training dataset, followed by evaluating the test dataset. (iii) Block level script identification without any rejection criteria. The first two experiments were performed to evaluate and support the effectiveness of our feature. The last was to make a comparative analysis of accuracy when more than one character is considered for deducing the script type. For the last experiment we follow the step as in our second experiment, but while deducing the script we consider a block of 1, 2, 3, 4, 5, and 6 or more characters and determine the script type of each block.

9.5 Classifier

In our experiment we noticed that Gaussian kernel SVM outperformed Neural Network and other SVM kernels in script classification, hence we choose SVM with Gaussian kernel. The Support Vector Machine (SVM) is defined for two-class problem and it looks for the optimal hyper-plane which maximizes the distance, the margin, between the nearest examples of both classes, named support vectors (SVs). Given a training database of M data: $\{x_m \mid m = 1, \cdots, M\}$ the linear SVM classifier is then defined as:

$$f(x) = \sum_j \alpha_j x_j \cdot x + b$$

where $\{x_j\}$ are the set of support vectors and the parameters $\{\alpha_j\}$ and b has been determined by solving a quadratic problem. The linear SVM can be extended to a non-linear classifier by replacing the inner product between the input vector x and the SVs,x_j, to a kernel function k defined as:

$$K(x, y) = \Phi(x) \cdot \Phi(y).$$

This Kernel function should be square integrable and verify the Mercer's Condition [29]. The Gaussian kernel is of the form:

$$k(x, y) = exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

Table 9.1: FIVE-FOLD CROSS VALIDATION ACCURACY ON TRAINING DATASET (WITH AVG. REJECTION CRITERIA OF 1.45%).

Chinese	Japanese	Koren	Roman
96.13%	95.63%	97.06%	98.60%
Average 96.85%			

Table 9.2: SCRIPT IDENTIFICATION ACCURACY ON TEST DATASET (WITH AVG. REJECTION CRITERIA OF 1.45%).

Chinese	Japanese	Koren	Roman
97.23%	94.64%	97.32%	98.63%
Average 96.95%			

Table 9.3: REJECTION VERSUS ACCURACY RESULT.

Rejection	Accuracy			
	C	J	K	R
1.45%	97.23%	94.64%	97.32%	98.63%
5.50%	98.93%	98.24%	98.92%	99.23%
10.00%	99.69%	99.65%	99.85%	99.94%

Besides optimizing the kernel parameters (such as gamma in a Gaussian kernel), one should consider trade-off parameter C (the penalty multiplier). This parameter indicates how severely errors have to be punished. We are not giving details of SVM due to its easy availability [[137],[29]]. We got best optimized results when gamma parameter $\frac{1}{2\sigma^2}$ is set to 48.00 and C is set to 10.

9.6 Results and Discussion

9.6.1 Script identification accuracy on single character

Here in Table 9.1, we show the character level script identification accuracy of our scheme, when a fivefold cross validation scheme is deployed on the training samples. Table 9.2 shows the character level script identification accuracy when our system is trained with the samples of training dataset and evaluated on the samples of testing dataset. We obtained an average accuracy of 96.85% while implementing five-fold cross validation on our training dataset and 96.95% when evaluating our test dataset. In both cases the rejection rate is 1.45%. Rejection is done based on confidence score of the classifier. If the script type of a character is not identified with a confidence score of greater than equal to 0.50; we consider that character as rejected sample. By confidence score, we mean to say the probability estimation of the recognized class [142]. Please note the standard deviation of average accuracy for two different experiments is 0.07 which is very small. This proves that the feature set used are quite robust. Also in Table 9.3 we report accuracy versus rejection rate of our system when evaluating our test dataset. Please note that with increased rate of rejection the accuracy of our system also increases. In Table 9.3, letters C, J, K and R denotes Chinese, Japanese, Korean and Roman script respectively.

9.6.2 Character level script identification accuracy with different confidence score

Here we analyze the distribution of all correctly script identified characters from our test dataset, with respect to their corresponding confidence score as given by the classifier model. It can be noted that some of the characters (about 30%) are recognized with a high confidence score (greater than or equal to 0.90), but most of the characters (about 50%) are having relatively lower confidence score of 0.50-0.60. This is because a large number of characters from Chinese and Japanese script are very similar. For those, though the classifier model returned correct class but corresponding confidence score was low. This phenomenon is illustrated in Fig.9.3.

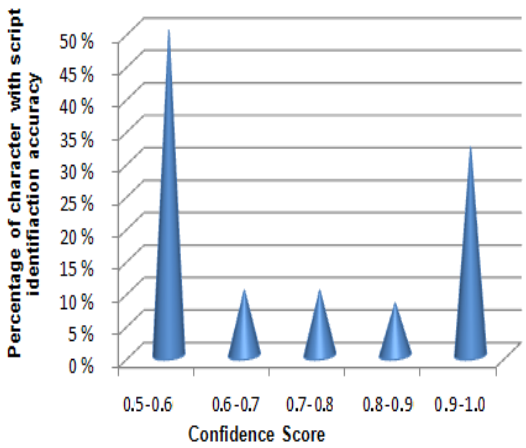

Figure 9.3: Distribution of script identified character with corresponding confidence score value.

9.6.3 Script identification accuracy on block level

Here we report the accuracy of our script identification scheme on block level. Following steps were followed in this experiment: (i) For all test images we perform necessary line and character segmentation, (ii) then features are extracted from all characters present in a block, (by block we mean to say a group of characters), (iii) feature value for individual character in a block is sent to classifier, (iv) classifier decides the script for each character in that block, (v) a majority voting scheme is deployed to deduce the script type for that block, (vi) in case of tie (when the classifier assigns equal number of characters to each script type in a block), then the script type for the block is determined in the following way: Now, say, in a block there are N characters. Now the classifier identifies N/2 characters as Japanese characters and the rest N/2 characters as Korean character. We sum the respective confidence score of all Japanese and Korean characters separately. If sum of confidence score of all Japanese characters is more than sum of confidence score of all Korean characters, then we assign Japanese script type to the block. A column chart given in Fig.9.4 shows that accuracy of our system rises as we increase the number of characters in a block. Please note that here we didnt impose any rejection criteria. Even when we compute character level accuracy in a block, we got an accuracy of 98.39%. This is much higher compared to the results discussed in sub-section 9.6.1, as a rejection criterion was imposed in experiments discussed in Section 9.6.1.

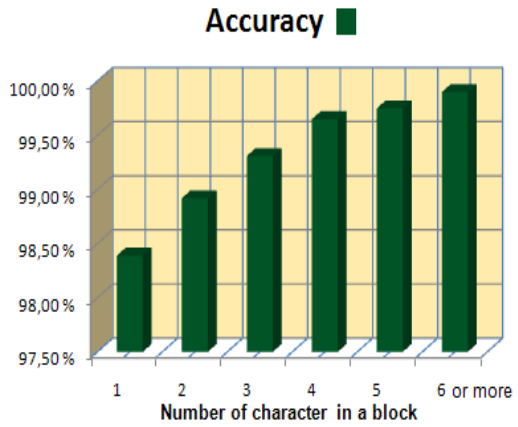

Figure 9.4: Accuracy with respect to number of character in a block.

9.6.4 Error Analysis

Analyzing the errors we noticed that most of the errors occurred due to deformed characters. Below are some examples of such erroneous Japanese (Kanji) characters. Due to bloated appearance, they were identified as Roman characters.

Figure 9.5: Example of three erroneous Kanji characters.

9.6.5 Comparison with similar other works

Though there are many pieces of work on script identification, not much of them are in the context of Han-based script identification. Ding et al. [71] developed a system for identification of oriental and occidental script. In average they obtained an accuracy of more than 99%. There they have considered Chinese, Japanese, Korean script as oriental script type. Pan et al. [103] proposed a block level script identification scheme of Chinese, Japanese, Korean and Roman scripts. They reported highest accuracy of 99.81% at block level of size 256x256 pixels. We digitized our images in 300 dpi and noticed that in our images, such a 256x256 window area consists of 6/7 characters on an average.We evaluate our system under similar circumstances as Pan et al. [103] and obtained an accuracy of 99.85%, which is higher than Pan et al. [103].

9.7 Conclusion

Here we propose a system to identify Chinese, Japanese, Korean and Roman scripts from a single document page. We obtained 96.95% accuracy at character level with a rejection rate of 1.45%. At block level we achieved 99.85% accuracy, when the number of characters in a block was 6 or more. We used a directional chain-code histogrambased feature for this

purpose and obtained encouraging results. In future we plan to make a detail analysis of our errors to improve our system.

Chapter 10

Font Identification In Context of an Indic Script

Abstract-Font can be used as a notion of similarity amongst multiple documents written in same script. We could automatically retrieve document images with specific font from a huge digital document repository. So Optical Font Recognition could be a useful pre-processing step in an automated questioned document analysis system for sorting documents with similar fonts. We propose a scheme to identify 10 different fonts for an Indic script (Bangla). Curvature-based features are extracted from segmented characters and are fed to a Support Vector Machine (SVM) classifier. The classifier determines the font type for each segmented character obtained from a document. Later, font identification for that document is executed on the basis of majority voting amongst 10 different fonts for all characters. Using a Multiple Kernel SVM classifier we obtained 98.5% accuracy from 400 test documents (40 documents for each font type). *

Keywords:- Font Identification; Computational Forensics; Document Analysis; Bangla Script; SVM; MKL-SVM.

10.1 Introduction

Font identification could be used as a pre-processing step in an automated questioned document analysis process. In a crime scene, a forensic expert might be only interested in analyzing documents those are type faced with a particular font. The pre-processing step will help the forensic expert to narrow down his search space for finding relevant evidence. Moreover identifying font of a document might help to determine its source of origin (since fonts are sometimes country specific).There are very few pieces of work on font identification [[148], [146], [81], [124], [90], [2]]. But mostly they dealt with Roman script fonts. Using typographical features font detection has been described by Zramdini et al. [148]. Zhu [146] described an automatic method for identification of 6 Chinese and 8 Roman fonts. He used a Gabor filter based texture analysis approach for discriminating fonts. Khoubyari et al. [81] reports a system which could identify 33 different fonts of Roman script. A Nearest neighbor classifier has been used by Manna et al. [90] for discriminating 4 different fonts in Roman script. Shi et al. [124] proposed a system to discriminate 9 Roman fonts. There they have used properties of the input page and also used graph matching results of recognized short words. A font identification system for Arabic script is being proposed in [2]. It can be noted that the present state-of the art addresses font identification from mainly Roman script perspective. There is no systematic study on font identification in context of any Indic script. Although the Indian subcontinent is home to a huge population with 11 official scripts. We propose a system to identify 10 different fonts for Bangla script, which is the 5th most popular script in the world and 2nd most popular script in India. Curvature based features with SVM and Multiple Kernel SVM are used here. The rest of the paper is arranged as follows. In Section 10.2, we describe line, word and character segmenta-

*Content of this chapter is mainly based on the article - "Font Identification - In Context of an Indic Script", Sukalpa Chanda, Umapada Pal and Katrin Franke , In Proc. 21st International Conference on Pattern Recognition, pp.1655-1658, ICPR 2012.

tion process. In Section 10.3, we describe our features along with a logical explanation of its utility for our objective. We describe our experimental setup with information on our dataset in Section 10.4. In Section 10.5, we give a brief discussion on our classifier. Result and discussions on various experiments are reported in Section 10.6. Finally, the article is concluded.

10.2 Line, Word and Character Segmentation

All document images are binarized and de-skewed before they are processed for possible line, word and character segmentation. The lines are segmented in the documents by finding the valleys of the histogram computed by counting the number of black pixels in each row. Then for each line, a column-wise vertical scanning is executed and word segmentation is carried out by considering the valley of histogram computed by the number of black pixels in each column. To segment individual characters of a word we consider only the middle zone. The basic approach here is to rub out the head line [37] so that the characters get topologically disconnected. To find the demarcation line of the characters a linear scanning in the vertical direction from the head line is initiated. If during a scan, one can reach the base line without touching any black pixel then this scan marks a boundary between two characters. For some kerned characters [37] a piecewise linear scanning method has been invoked. In few cases we observed wrong character segmentation due to presence of unwanted noise in the document image. Details about this can be found in [37].

10.3 Curvature Based Feature Extraction

In many Bangla characters/text intensity of roundness in character contour varies amongst different font types, even when the characters are exactly the same. Motivation behind using curvature features is that it can be used as a local shape descriptor to express this discrimination present in the characters of different fonts.

10.3.1 Feature computation for Curvature feature

Curvature features used in this paper has been calculated using bi-quadratic interpolation method. The details can be found in [125]. To get the curvature feature the following steps are applied.

Step 1: The direction of gradient is quantized to 32 levels with $\pi/16$ intervals.

Step 2: The curvature computed by the mentioned formula is quantized into 3 levels using a threshold (t)(for concave, linear and convex regions). For concave region $c \leq -t$, for linear region $(-t < c < t)$ and for convex region $c \geq t$. We assume t as 0.10 in our experiment.

Step 3: The strength of the gradient is accumulated in each of the 32 directions and in each of the 3 curvatures levels of each block to get 49x49 local joint spectra of directions and curvatures.

Step 4: A spatial and directional resolution is made as follows. A smoothing filter [1 4 6 4 1] is used to get 16 directions from 32 directions. On this resultant image, another smoothing filter [1 2 1] is used to get 8 directions from 16 directions. Further more, we use a 31 x 31 two-dimensional Gaussian-like filter to get smoothed 7 x 7 blocks from 49 x 49 blocks. So, we get 7x7x8 = 392 dimensional feature vector. Using curvature feature in 3 levels we get $392 \times 3 = 1176$ dimensional features.

10.4 Dataset Details and Experimental Design

Our entire dataset comprised of 50 document images from each font type (500 images in total). All images in our dataset were scanned to 300 dpi and stored in tiff format. Out of

these 500 files, 100 files were used for training (10 files from each font type). From those 100 training files we extracted features from 85,695 segmented characters. The rest of the 400 images were used for testing. The name of the ten fonts used in our experiments are as follows: Aponalohit, Kalpurush, Solaimanlipi, SiyamRupali, Vrinda, Nikosh, Bensen-handwriting, Bengali, Adorsholipi, Bensen. Examples of these fonts are shown in figure (10.1). All fonts are available in [1] for free download. It can be noted that "Kalpurush", and "Solaimanlipi" fonts are visually very similar.

Figure 10.1: Example of two words written in 10 different fonts, respective name of font in English appearing on bottom.

We mainly performed 2 different experiments. Which are as follows: (i) Character level font identification based on five-fold cross validation on the training dataset; (ii) Document level font identification based on training the classifier with the training dataset, followed by evaluating the test dataset by using a normal Gaussian Kernel SVM and a MKL-SVM (with 5 base kernels). Initially we were experimenting with only a Gaussian Kernel SVM. We noticed that in a few of our test document images the change in $\frac{1}{2\sigma^2}$ in Gaussian kernel resulted in accurate results, though for the rest of the test document images we were getting better accuracy at another $\frac{1}{2\sigma^2}$ value. This pursued us towards a multiple kernel architecture where the parameters will get adjusted according to the data. On using MKL-SVM we further improved our results.

10.5 Classifier

10.5.1 Support Vector Machine

The Support Vector Machine (SVM) is defined for two class problem and it looks for the optimal hyper-plane which maximizes the distance, the margin, between the nearest examples of both classes, named support vectors (SVs). Given a training database of M data: $\{x_m \mid m = 1, \cdots, M\}$ the linear SVM classifier is then defined as:

$$f(x) = \sum_j \alpha_j x_j \cdot x + b$$

where $\{x_j\}$ are the set of support vectors and the parameters α_j and b has been determined by solving a quadratic problem. The linear SVM can be extended to a non-linear classifier by replacing the inner product between the input vector x and the SVs, x_j, to a kernel

function k defined as: $K(x,y) =< \Phi(x), \Phi(y) >$. This Kernel function should be square integrable and verify the Mercers Condition [137]. The Gaussian kernel is of the form:

$$k(x,y) = exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$$

Besides optimizing the kernel parameters (such as gamma in a Gaussian kernel), one should consider the penalty multiplier parameter as well. This parameter indicates how severely errors have to be punished. We are not giving details of SVM due to its easy availability [[137],[29]]. We got best optimized results when gamma parameter $(1/2\sigma^2)$ is set to 0.05 and penalty multiplier parameter is set to 1.

10.5.2 Multiple Kernel SVM

During initial experimentation we noticed that optimal result is achieved when gamma parameter $(\frac{1}{2\sigma^2})$ of our SVM was set to 0.05, but on some miss-classified samples a different gamma parameters (1.00, 4.00 etc.,) were giving correct results. Unfortunately these different gamma values were affecting the overall accuracy by misclassifying some other samples, those were correctly classified when gamma value was set to 0.05. As a result a Multiple Kernel SVM framework was deployed to encounter the problem of automatic selection of kernel parameter according to the data. In a Multiple Kernel scenario a convex combination of some base Kernels forms the final Kernel in the following form:
$K(x_i, x_j) = \sum_k \; d_k \;\; K_k(x_i, x_j)$
The original formulation of MKL with L_1 norm regularization [84] leads to a dual which is not differentiable, this adversary is later encountered in [108], [94]. Solving an optimization problem, weights (d_k) for each base kernel are obtained. We deployed SMO-MKL framework for our experiment [138]. Our base kernels were Gaussian Kernel with gamma ($\frac{1}{2\sigma^2}$ in the formula) that was set to following values: 0.0005, 0.05, 1.00, 4.00, and 12.00, respectively. Details about MKL SVM can be found in [108][94][84][138].

10.6 Results and Discussions

10.6.1 Font identification accuracy on Five-Fold Cross Validation at Single Character Level

Here we show the average character level font identification accuracy of our scheme, when a fivefold cross validation scheme using a SVM/ MKL SVM is deployed on features of 85,695 segmented characters obtained from training samples. The average accuracy on five-fold cross validation on the training dataset was 85.65% when using a MKL-SVM. On same experimental setup a Gaussian kernel SVM achieved 85.00% accuracy.

10.6.2 Font Identification Accuracy on Document Level

Here we report the accuracy of our font identification scheme on document level. Our test dataset comprises of 400 images. We obtained 98.5% accuracy at document level when we used MKL-SVM, with a single kernel SVM the accuracy was 94.00%. Though a 4.5% increase is achieved at document level, it is not due to the fact that MKL-SVM hugely outperformed SVM at character level font identification in our test images. Rather we noticed that while performing majority voting, MKL-SVM marginally won on few characters over SVM in some test documents and hence the increase in accuracy. So we can say that in our case, performance of MKL-SVM and SVM were close to each other. Following steps were followed in this experiment: (i) For all test images we performed necessary line and character segmentation;(ii) then features are extracted from all segmented characters; (iii) feature value for individual character is sent to the classifier; (iv) the classifier decides the

font for each character; (v) a majority voting scheme was deployed to deduce the font type for that document image; (vi) in the case of tie (when the classifier assigns equal number of character to two font types in a document image), the font type for the document was determined in the following way: We sum the respective confidence scores of both font types separately. The font type with the maximum sum of confidence score is assigned as the font type for the document.

10.6.3 Weight Distribution on Base Kernels

We analyzed the distribution of weights for each base kernel. Here the multi-class problem in MKL-SVM is solved by taking a 1:1 class approach. Since there are 10 classes, all-together 45 optimization problems are solved. We analyzed the distribution of kernel weights obtained after all 45 optimizations were accomplished, the corresponding weights obtained for each base kernels are depicted in figure (10.2). It can be observed from the graph depicted in figure (10.2), that one of the base kernel with gamma value 0.05 got highest weights during almost all optimization process. This is justified, because on our experimentation with single Kernel SVM, we obtained the best accuracy when gamma value is set to 0.05.

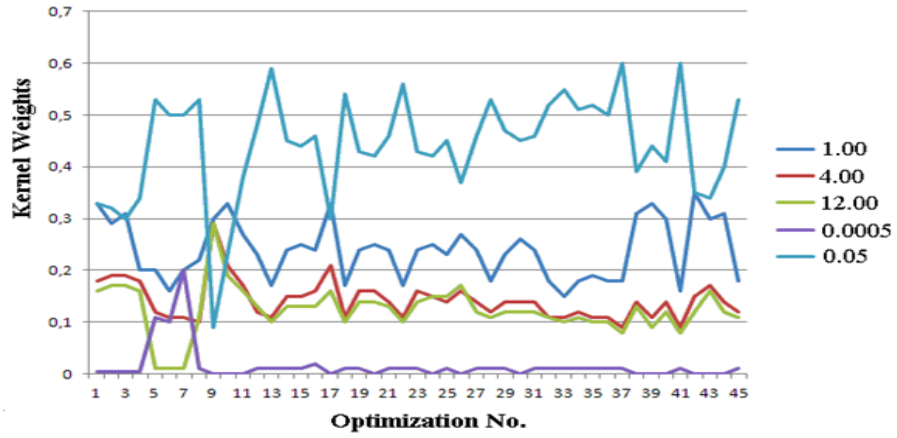

Figure 10.2: Weights of Kernels obtained after MKL optimization. Each line corresponds to every single base Kernel.

10.6.4 Distribution of Confidence Score

We analyzed also the distribution of top choice confidence score while using normal SVM as well as MKL-SVM in our test document images. It can be easily noted that about 75% of recognized characters in normal SVM had a confidence score between 0.25-0.5 but when MKL SVM is used around 55% of characters got a confidence score between 0.5-0.8. This phenomenon is depicted in figure (10.3).

10.6.5 Error Analysis

A large number of errors came due to confusion between ”Kalpurush”, and ”Solaimanlipi” at character level. It can be easily noted from the centre and right most image of two similar words in the top row of figure (10.1), that the fonts are visually alike to each other. Some of the errors occured due to the wrong busy zone detection in small length words. However this can be removed considering busy zone of the text line.

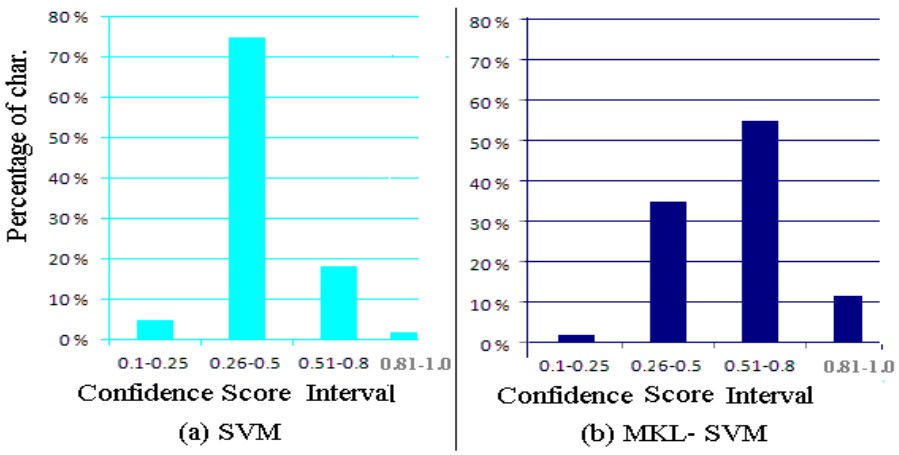

Figure 10.3: Distribution of percentage of characters within different confidence score interval: (a) SVM (b) MKL SVM.

10.6.6 Comparison with Similar Other Works

The present state-of the art addresses font identification in the context of mainly Roman script. No work on font identification for Indic scripts exists to the best of our knowledge. So we don't have the opportunity to compare results on same corpus. Yet to give a comparison of efficacy of other published work we present some information below. Zramdini and Ingold [148] obtained an average accuracy of 97% while dealing with 10 different fonts of Roman scripts. Zhu et al.[146] followed a Gabor filter-based texture analysis approach and obtained an accuracy of 99.1% when dealing with 6 Chinese and 8 Roman script fonts. Shi et.al [124] proposed a system for identification of nine Roman fonts, and they obtained an average accuracy of 85% approx. Abuhaiba [2] proposed a template based approach for font recognition in Arabic script, where he achieved an overall accuracy of 77.4% when 7.6% rejection rate is considered. We obtained an over-all accuracy of 98.5% when using MKL-SVM at document level on 400 test images when 10 Bangla fonts are considered.

10.7 Conclusion

Here we propose a system to identify 10 different fonts in Bangla script. We obtained 94.00% accuracy at document level when using a normal Gaussian Kernel SVM. After using a MKL SVM, we achieved 98.5% accuracy at document level. We used curvature based features as they act as a local shape descriptor showing differences in curvatures of various fonts. In future we plan to do font identification of other Indic scripts as well. Though for our experiments MKL-SVM is little bit superior to SVM in terms of accuracy, we found MKL-SVM more time consuming compared to normal SVM.

Bibliography

[1] http://www.omicronlab.com/bangla-fonts.html. 83

[2] ABUHAIBA, I. Arabic font recognition based on templates. *The International Arab Journal of Information Technology 1* (2001), 33–39. 30, 81, 86

[3] AL-MAADEED, S. Text-dependent writer identification for arabic handwriting. *Journal of Electrical and Computer Engineering* (2012), 1–8. 29

[4] AN, C., BAIRD, H. S., AND XIU, P. Iterated document content classification. In *International Conference on Document Analysis and Recognition* (2007), pp. 252–256. 23, 37, 44, 45

[5] BANERJI, R. *The Origin of the Bengali Script*. Calcutta University Press, Calcutta,India, 1919. 12

[6] BANERJI, R. *A Descriptive Study of the Modern Bengali Script*. Lambert Academic Publishing, Saarbrucken, Germany, 2011. 11

[7] BEN MOUSSA, S., ZAHOUR, A., BENABDELHAFID, A., AND ALIMI, A. M. New features using fractal multi-dimensions for generalized arabic font recognition. *Pattern Recognition Letter 31*, 5 (2010), 361–371. 30

[8] BENSEFIA, A., NOSARY, A., PAQUET, T., AND HEUTTE, L. Writer identification by writer's invariants. In *International Workshop on Frontiers in Handwriting Recognition* (2002), pp. 274–279. 28

[9] BENSEFIA, A., PAQUET, T., AND HEUTTE, L. Information retrieval based writer identification. In *In Proc. 7th International Conference on Document Analysis and Recognition* (2003), pp. 946–950. 28

[10] BENSEFIA, A., PAQUET, T., AND HEUTTE, L. Handwriting analysis for writer verification. In *International Workshop on Frontiers in Handwriting Recognition* (2004), pp. 196–201. 28

[11] BENSEFIA, A., PAQUET, T., AND HEUTTE, L. A writer identification and verification system. *Pattern Recognition Letters 26*, 13 (2005), 2080–2092. 28

[12] BHARDWAJ, A., REDDY, M., SETLUR, S., GOVINDARAJU, V., AND SITARAM, R. Latent dirichlet allocation based writer identification in offline handwriting. In *In Proc. Document Analysis Systems* (2010), pp. 357–362. 28

[13] BLOG. http://science.howstuffworks.com/handwriting-analysis.html, last retrieved on 15.4.2014. 12, 13

[14] BLOG. http://www.nha-handwriting.org.uk/handwriting/what-is-handwriting, last retrieved on 20.6.2014. 12

[15] BLOG. Torn document examination example picture last retreived on 18.04.2014 from http://www.wired.com/politics/security/magazine/16-02/ff_stasi?currentpage=all. 1

[16] BLOG, W. Discussion on pseudoscience and forensics last retreived on 1.8.2014 from http://www.enotes.com/pseudoscience-forensics-reference/pseudoscience-forensics. 13

[17] BLOG, W. A historical perspective on graphological references and validation studies last retreived on 26.2.2015 from http://bazaarmodel.net/phorum/read.php?1,3951. 12

[18] BLOG, W. Is handwriting-analysis a science retreived on 1.10.2014 from http://www.straightdope.com/columns/read/2447/is-handwriting-analysis-legit-science. 13

[19] BLOG, W. Notes on handwriting analysis last retreived on 1.8.2014 from http://www.enotes.com/handwriting-analysis-reference/handwriting-analysis. 13

[20] BLOGSPOT, W. http://crsouza.blogspot.in/2010/03/kernel-functions-for-machine-learning.html, last retrieved on 15.4.2014. 18, 20

[21] BOUGHORBEL, S., TAREL, J.-P., AND BOUJEMAA, N. Generalized histogram intersection kernel for image recognition. In *In Proc. International Conference on Image Processing* (2005), pp. 161–164. 20

[22] BOUSQUET, O., BOUCHERON, S., AND LUGOSI, G. Introduction to statistical learning theory, 2006. 14

[23] BRINK, A., SMIT, J., BULACU, M., AND SCHOMAKER, L. R. B. Writer identification using directional ink-trace width measurements. *Pattern Recognition 45*, 1 (2012), 162–171. 28

[24] BUKHARI, S. S., SHAFAIT, F., AND BREUEL, T. M. Segmentation of curled textlines using active contours. In *Document Analysis Systems* (2008), pp. 270–277. 25

[25] BULACU, M., AND SCHOMAKER, L. Writer style from oriented edge fragments. In *International Conference on Computer Analysis of Images and Patterns* (2003), pp. 460–469. 28

[26] BULACU, M., AND SCHOMAKER, L. A comparison of clustering methods for writer identification and verification. In *International Conference on Document Analysis and Recognition* (2005), pp. 1275–1279. 28

[27] BULACU, M., AND SCHOMAKER, L. Text-independent writer identification and verification using textural and allographic features. *IEEE Transactions on PAMI 29* (2007), 701–718. 27, 28, 47, 55

[28] BULACU, M. L. *Statistical Pattern Recognition for Automatic Writer Identification and Verification*. Ph.D. thesis, Groningen, The Netherlands, 2008. 13

[29] BURGES, C. J. A tutorial on support vector machines for pattern recognition. In *Data mining and knowledge discover,2* (1998), pp. 1–43. 41, 50, 58, 67, 75, 76, 84

[30] BUSCH, A., BOLES, W., AND SRIDHARAN, S. Texture for script identification. *IEEE Transactions on PAMI 27* (2005), 1720–1732. 26

[31] CAMPBELL, C. http://videolectures.net/epsrcws08_campbell_isvm. 17, 18

[32] C.FRANCKS, DELISI, L., FISHER, S., LAVAL, S., RUE, J., J.F.STEIN, AND A.P.MONACO. Confirmatory evidence for linkage of relative hand skill to 2p12-q11. *American Journal of Human Genetics 22* (2003), 499–502. 13

[33] CHAKRAVARTI, S. N. Development of the bengali alphabet from the 5th century a.d. to the end of the muhammadan rule. *Journal of the Royal Asiatic Society of Bengal 4* (1938), 351–391. 11

[34] CHANDA, S., FRANKE, K., AND PAL, U. Structural handwritten and machine print classification for sparse content and arbitrary oriented document fragments. In *ACM Symposium on Applied Computing 2010* (2010), pp. 18–22. 42, 66

[35] CHANDA, S., FRANKE, K., PAL, U., AND WAKABAYASHI, T. Text independent writer identification for bengali script. In *International Conference on Pattern Recognition* (2010), pp. 2005–2008. 55, 61

[36] CHANDA, S., PAL, U., AND KIMURA, F. Identification of japanese and english script from a single document page. In *In Proc. International Conf. on Information Technology* (2007), pp. 656–661. 74

[37] CHAUDHURI, B. B., AND PAL, U. A complete printed bangla ocr system. *Pattern Recognition 31*, 5 (1998), 531–549. 82

[38] CHOWDHURY, S. P., MANDAL, S., DAS, A. K., AND CHANDA, B. Segmentation of text and graphics from document images. In *International Conference on Document Analysis and Recognition* (2007), pp. 619–623. 24, 37, 44, 45

[39] C.L.TAN, LEONG, P., AND S.HE. Language identification in multi-lingual documents. In *International Symposium on Intelligent Multimedia and Distance Education* (1999), pp. 59–64. 26

[40] COGGINS, J. M. *A framework for texture analysis based on spatial filtering*. Ph.D. thesis, East Lansing, MI, USA, 1983. 9

[41] CRISTIANINI, N. Kernel methods for general pattern analysis. 19

[42] CRISTIANINI, N., AND SHAWE-TAYLOR, J. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, UK, 2001. 19

[43] CRYSTAL, D. *The Cambridge Encyclopaedia of English Language*. Cambridge University Press, Cambridge,UK, 1995. 11

[44] DASH, N. S. The bengali script and the unicode. *Print Out 2*, 8 (2011), 1–16. 12

[45] D.DOERMANN, AND TOMBRE, K. *Handbook of Document Image Processing and Recognition*. Springer, Germany, 2014. 12

[46] DESIGNER WALL, W. Discussion on western typography last retreived on 1.08.2014 from http://webdesignerwall.com/general/brief-history-of-western-typography/comment-page-2?replytocom=58736. 12

[47] DHANYA, D., AND RAMAKRISHNA, A. G. script identification in printed bilingual documents. In *Document Analysis System* (2002), pp. 13–24. 27

[48] DHANYA, D., RAMAKRISHNA, A. G., AND PATI, P. B. Script identification in printed bilingual documents. *Sadhana 27* (2002), 73–82. 27, 64

[49] DING, X., CHEN, L., AND WU, T. Character independent font recognition on a single chinese character. *IEEE Trans. Pattern Analysis and Machine Intelligence 29*, 2 (2007), 195–204. 30

[50] DIRINGER, D. *The Alphabet: A Key to the History of Mankind*. Hutchinsons Scientific and Technical Publications, London,UK, 1948. 11

[51] DTREG. http://www.dtreg.com/svm.htm. 19

[52] ELGAMMAL, A. M., AND ISMAIL, M. A. Techniques for language identification for hybrid arabic-english document images. In *International Conference on Document Analysis and Recognition* (2001), pp. 1100–1104. 26

[53] F. ARGENTI, L. A., AND BENELLI, G. Fast algorithms for texture analysis using co-occurrence matrices. In *IEEE Proceedings on Radar and Signal Processing* (1990), pp. 443–448. 23

[54] FISHER, J. L., HINDS, S. C., AND DAMATO, D. P. A rule-based system for document image segmentation. In *International Conference on Pattern Recognition* (1990), pp. 567–572. 24

[55] FLETCHER, L. A., AND KASTURI, R. A robust algorithm for text string separation frommixed text/graphics images. *IEEE Trans. Pattern Analysis and Machine Intelligence 10(6)* (1988), 910–918. 23, 37

[56] FRANKE, K. *The influence of physical and biomechanical processes on the ink trace - Methodological foundations for the forensic analysis of signatures*. Ph.D. thesis, Groningen, The Netherlands, 2005. 9, 12

[57] FRANKE, K. Analysis of authentic signatures and forgeries. In *International workshop in Computaional Forensics* (2009), pp. 150–164. 13

[58] FRANKE, K., BNNEMEYER, O., AND SY, T. Writer identification using ink texture analysis. In *International Workshop on Frontiers of Handwriting Recognition* (2002), pp. 268–273. 27, 47, 55

[59] GARAIN, U., AND PAQUET, T. Off-line multi-script writer identification using AR coefficients? In *International Conf. on Document Analysis and Recognition* (2009), pp. 991–995. 29, 47, 52, 55, 61

[60] GHOSH, D., DUBE, T., AND SHIVAPRASAD, A. P. Script recognition - a review. *IEEE Trans. Pattern Anal. Mach. Intell. 32*, 12 (2010), 2142–2161. 11, 21, 25

[61] GUO, J. K., AND MA, M. Y. Separating handwritten material from machine printed text using hidden markov models. In *International Conference on Document Analysis and Recognition* (2001), pp. 439–443. 25, 38, 45, 46

[62] GUPTA, S., AND NAMBOODIRI, A. Text dependent writer verification using boosting. In *In Proc.International Conference on Frontiers in Handwriting Recognition* (2008). 29

[63] HAN, C., AND FAN, K. Skeleton generation of engineering drawings via contour matching. In *Pattern Recognition* (1994), vol. 27, pp. 261–275. 23

[64] HARALICK, R. M., SHANMUGAM, K., AND DINSTEIN, I. Textural features for image classification. 610–621. 23

[65] HE, Z., YOU, X., AND TANG, Y. Y. Writer identification using global wavelet-based features. *Neurocomputing 71*, 10-12 (2008), 1832–1841. 28

[66] HELLI, B., AND MOGHADDAM, M. E. A text-independent persian writer identification based on feature relation graph (frg). *Pattern Recognition 43*, 6 (2010), 2199–2209. 27

[67] HERTEL, C., AND BUNKE, H. A set of novel features for writer identification. In *In Proc. 4th International Conf. Audio-and Video-Based Biometrie Person Authentication* (2003), pp. 679–687. 28

[68] HOANG, T. V., AND TABBONE, S. Text extraction from graphical document images using sparse representation. In *Document Analysis Systems* (2010), pp. 143–150. 25

[69] HOCHBERG, J., P KELLY, T. T., AND KERNS, L. Automatic script identification from document images using cluster-based templates. *IEEE Trans. on PAMI 19* (1997), 176–181. 26, 64, 73

[70] INTERTEL. http://www.intertel.co.za/questioned-document-examination-forensics-handwriting-analysis.html, last retrieved on 18.4.2014. 3

[71] J. DING, L. L., AND SUEN, C. Y. Classification of oriental and european scripts by using characteristic features. In *International Convention on Document Analysis and Recognition* (1997), pp. 1023–1027. 73, 78

[72] JAEGER, S., MA, H., AND DOERMANN, D. Identifying script on word-level with informational confidence. In *In Proc. 8th International Conf. on Document Analysis and Recognition* (2005), pp. 416–420. 27, 63, 73

[73] JAIN, A. K., AND BHATTACHARJEE, S. Text segmentation using gabor filters for automatic document processing. In *Machine Vision and Applications* (1992), pp. 169–184. 23

[74] JAKKULA, V. Tutorial on support vector machine last retrieved on 18.09.2014 from http://www.ccs.neu.edu/course/cs5100f11/resources/jakkula.pdf. 15

[75] JANG, S. I., JEONG, S. H., AND NAM, Y.-S. Classification of machine-printed and handwritten addresses on korean mail piece images using geometric features. In *International Conference on Pattern Recognition* (2004), pp. 383–386. 25, 38

[76] JAWAHAR, C., KUMAR, M. P., AND KIRAN, S. A bilingual ocr for hindi-telugu documents and its applications. In *International Conference on Document Analysis and Recognition* (2003), pp. 408–412. 26

[77] KANDAN, R., REDDY, N., ARVIND, K., AND RAMAKRISHNAN, A. A robust two level classification algorithm for text localization in documents. 25, 45, 46

[78] KANDAN, R., REDDY, N. K., ARVIND, K., AND RAMAKRISHNAN, A. A robust two level classification algorithm for text localization in documents. In *International Symposium on Visual Computing*. 38

[79] KEYSERS, D., SHAFAIT, F., AND BREUEL, T. M. Document image zone classification - a simple high-performance approach. In *International Conference on Computer Vision Theory and Applications* (2007), pp. 44–51. 24

[80] KHOTANZAD, A., AND HONG, Y. H. Invariant image recognition by zernike moments. *IEEE Transactions on PAMI 12*, 5 (1990), 489–497. 67

[81] KHOUBYARI, S., AND HULL, J. Font and function word identification in document recognition. *Computer Vision and Image Understanding 63* (1996), 66–74. 29, 81

[82] K.K.CHIN. http://svr-www.eng.cam.ac.uk/ kkc21/thesis_main/node9.html. 14

[83] KUHNKE, K., SIMONCINI, L., AND KOVACS-V, Z. M. A system for machine-written and hand-written character distinction. pp. 811–814. 24, 38

[84] LANCKRIET, G. R. G., CRISTIANINI, N., BARTLETT, P., GHAOUI, L. E., AND JORDAN, M. I. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research 59* (2004), 27–72. 84

[85] LEE, S., AND KIM, J. Techniques for language identification for hybrid arabic-english document images. In *International Conference on Document Analysis and Recognition* (1995), pp. 28–33. 26

[86] LETTNER, M., AND SABLATNIG, R. Higher order mrf for foreground-background separation in multi-spectral images of historical manuscripts. In *Document Analysis Systems* (2010), pp. 317–324. 23

[87] LI, Y., ZHENG, Y., DOERMANN, D., AND JAEGER, S. Script-independent text line segmentation in freestyle handwritten documents. *IEEE Trans. Pattern Analysis and Machine Intelligence*. 24

[88] LU, S., TAN, C., AND HUANG, W. Language identification in degraded and distorted document images. In *Document Analysis System* (2006), pp. 232–242. 26

[89] MA, H., AND DOERMANN, D. Gabor filter based multi-class classifier for scanned document images. In *In Proc. 7th International Conf. on Document Analysis and Recognition* (2003), pp. 968–972. 27

[90] MANNA, S., COLLA, A., AND SPERDUTI, A. Optical font recognition for multi-font ocr and document processing. In *International Workshop on Database and Expert Systems Applications* (1999). 29, 81

[91] MAO, S., AND KANUNGO, T. Empirical performance evaluation methodology and its application to page segmentation algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence 23*, 3 (2001), 242–256. 24

[92] MARTI, U.-V., AND BUNKE, H. The iam-database: an english sentence database for offline handwriting recognition. *International Journal of Document Analysis and Recognition 5*, 1 (2002), 39–46. 28

[93] MARTI, U.-V., MESSERLI, R., AND BUNKE, H. Writer identification using text line based features. In *ICDAR* (2001), pp. 101–105. 27, 28, 47, 55

[94] M.KLOFT, BREFELD, U., SONNENBURG, S., LASKOV, P., K.-R.MULLER, AND ZIEN, A. Efficient and accurate lp-norm multiple kernel learning. In *In Proc. Neural Information Processing System* (2009), pp. 997–1005. 84

[95] MOORE, A. W. http://www.autonlab.org/tutorials/svm15.pdf. 16

[96] PAL, U. Automatic script identification: A survey. *Vivek* (2006), 26–35. 73

[97] PAL, U., AND CHAUDHURI, B. B. Script line separation from indian multi-script documents. In *International Conf. on Document Analysis and Recognition* (1999), pp. 406–409. 26

[98] PAL, U., AND CHAUDHURI, B. B. Identification of different script lines from multi-script documents. *Image and Vision Computing 20* (2002), 945–954. 26

[99] PAL, U., AND CHAUDHURY, B. Machine-printed and hand-written text lines identification. In *Pattern Recognition Letters* (2001), vol. 22, pp. 431–441. 24, 38

[100] PAL, U., AND DATTA, S. Segmentation of bangla unconstrained handwritten text. In *International Conf. on Document Analysis and Recognition* (2003), pp. 1128–1132. 47, 48

[101] PAL, U., SHARMA, N., WAKABAYASHI, T., AND KIMURA, F. Handwritten numeral recognition of six popular indian scripts. In *International Conference on Document Analysis and Recognition* (2007), pp. 749–753. 41, 48, 67, 74

[102] PAL, U., SINHA, S., AND CHAUDHURI, B. B. Multi-script line identification from indian documents. In *International Conf. on Document Analysis and Recognition* (2003), pp. 880–884. 26, 64

[103] PAN, W. M., SUEN, C. Y., AND BUI, T. D. Script identification using steerable gabor filters. In *In Proc. 8th International Conf. on Document Analysis and Recognition* (2005), pp. 883–887. 27, 73, 78

[104] PATIL, S. B., AND SUBAREDDY, N. V. Neural network based system for script identification in indian scripts. *Sadhana 27* (2002), 83–97. 26, 64

[105] PATRICIO, M., AND MARAVALL, D. Segmentation of text and graphics/images using the gray-level histogram fourier transform. In *Joint IAPR International Workshops on Advances in Pattern Recognition,LNCS 1876* (2000), pp. 757–766. 24, 38, 45

[106] PENG, X., SETLUR, S., GOVINDARAJU, V., AND SITARAM, R. Using a boosted tree classifier for text segmentation in hand-annotated documents. *Pattern Recognition Letters 33*, 7 (2012), 943–950. 25

[107] PURKAIT, P., KUMAR, R., AND CHANDA, B. Writer identification for handwritten telugu documents using directional morphological features. In *In Proc.International Conference on Frontiers in Handwriting Recognition* (2010), pp. 658–663. 29, 55, 61

[108] RAKOTOMAMONJY, A., BACH, F., GRANDVALET, Y., AND CANU, S. Simplemkl. *Journal of Machine Learning Research 9* (2008), 2491–2521. 84

[109] RAVEAUX, R., BURIE, J.-C., AND OGIER, J.-M. A colour text/graphics separation based on a graph representation. In *International Conference on Pattern Recognition* (2008), pp. 1–4. 23

[110] ROY, P. P., LLADOS, J., AND PAL, U. Text/graphics separation in color maps. In *International Conference on Computing: Theory and Applications* (2007), pp. 545–551. 24

[111] S, L., AND TAN, C. Script and language identification in noisy and degraded document images. *IEEE Trans. Pattern Analysis and Machine Intelligence 30*, 1 (2008), 14–24. 73

[112] SAID, H., TAN, T., AND BAKER, K. Personal identification based on handwriting. *Pattern Recognition 33*, 1 (2000), 149–160. 27, 47, 55

[113] SAITOH, T., TACHIKAWA, M., AND YAMAAI, T. Document image segmentation and text area ordering. In *International Conference on Document Analysis and Recognition* (1993), pp. 323–329. 24

[114] SCHLAPBACH, A., AND BUNKE, H. Using HMM-based recognizers for writer identification and verification. In *International Workshop on Frontiers of Handwriting Recognition* (2004), pp. 167–172. 27, 47, 55

[115] SCHLAPBACH, A., KILCHHERR, V., AND BUNKE, H. Improving writer identification by means of feature selection and extraction. In *8th International Conference on Document Analysis and Recognition* (2005), pp. 131–135. 28

[116] SCHLAPBACH, A., LIWICKI, M., AND BUNKE, H. A writer identification system for on-line whiteboard data. *Pattern Recognition 41*, 7 (2008), 2381–2397. 28

[117] SCHÖLKOPF, B., AND SMOLA, A. J. *Learning with Kernels*. MIT Press, USA, 2002. 15

[118] SCHOMAKER, L., AND BULACU, M. Automatic writer identification using connected-component contours and edge-based features of uppercase western script. *IEEE Transactions on PAMI 26*, 6 (2004), 787–798. 27, 28, 47, 55

[119] SCHOMAKER, L., FRANKE, K., AND BULACU, M. Using codebooks of fragmented connected-component contours in forensic and historic writer identification. *Pattern Recognition Letters 28* (2007), 719–727. 27, 47, 55

[120] SEOK LEE, Y., SUH KOO, H., AND SUNG JEONG, C. A straight-line detection using principal component analysis. *Pattern Recognition Letters* (2006), 1744–1754. 39, 40, 65

[121] SHAFAIT, F., KEYSERS, D., AND BREUEL, T. M. Pixel-accurate representation and evaluation of page segmentation in document images. In *International Conference on Pattern Recognition* (2006), pp. 872–875. 24

[122] SHAFAIT, F., KEYSERS, D., AND BREUEL, T. M. Performance evaluation and benchmarking of six-page segmentation algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence 30*, 6 (2008), 941–954. 24, 37

[123] SHAFAIT, F., VAN BEUSEKOM, J., KEYSERS, D., AND BREUEL, T. M. Background variability modeling for statistical layout analysis. In *International Conference on Pattern Recognition* (2008), pp. 1–4. 23

[124] SHI, H., AND PAVLIDIS, T. Font recognition and contextual processing for more accurate text recognition. In *International Conference on Document Analysis and Recognition* (1997), pp. 1945–1948. 29, 81, 86

[125] SHI, M., FUJISAWA, Y., WAKABAYASHI, T., AND KIMURA, F. Handwritten numeral recognition using gradient and curvature of grayscale image. *Pattern Recognition 35* (2000), 2051–2059. 56, 82

[126] SINHA, S., PAL, U., AND CHAUDHURI, B. B. *Word-wise Identification from Indian documents*. Lecture Notes on Computer Science LNCS-3136, 2004. 26, 64

[127] SLIMANE, F., KANOUN, S., ALIMI, A. M., INGOLD, R., AND HENNEBERT, J. Gaussian mixture models for arabic font recognition. In *International Conference on Pattern Recognition* (2010), pp. 2174–2177. 30

[128] SLIMANE, F., KANOUN, S., HENNEBERT, J., ALIMI, A. M., AND INGOLD, R. A study on font-family and font-size recognition applied to arabic word images at ultra-low resolution. *Pattern Recognition Letters 34*, 2 (2013), 209–218. 30

[129] SPITZ, A. L. Determination of the script and language content of document images. *IEEE Trans on Pattern Analysis and machine Intelligence 19* (1997), 235–245. 26, 63, 73

[130] SREERAJ, M., AND IDICULA, S. A survey on writer identification scheme. *International Journal of Computer Applications 26*, 2 (2011), 2333. 27

[131] SRIHARI, S., CHA, S., ARORA, H., AND LEE, S. Individuality of handwriting. *Journal of Forensic Sciences 47*, 4 (2002), 1–17. 27, 29, 47, 55

[132] SRIHARI, S. N., BEAL, M., BANDI, K., SHAH, V., AND KRISHNAMURTHY, P. A statistical model for writer verification. In *International Conference on Document Analysis and Recognition*, pp. 1105–1109. 27, 47, 55

[133] THUMWARIN, P., AND MATSUURA, T. On-line writer recognition for thai based on velocity of barycenter of pen-point movement. In *International Conference on Image Processing* (2004), pp. 889–892. 28

[134] T.N.TAN. Rotation invariant texture features and their use in automatic script identification. *IEEE Trans on PAMI 20* (1998), 751–756. 27, 64, 73

[135] TRIPATHY, N., AND PAL, U. Handwriting segmentation of unconstrained oriya text. In *International Workshop on Frontiers in Handwriting Recognition* (2004), pp. 306–311. 55, 56

[136] TUCERYAN, M., AND JAIN, A. Texture analysis. *Handbook of Pattern Recognition and Computer Vision, 2nd Edition*. 9, 10

[137] VAPNIK, V. *The nature of statistical learning theory*. Springer-Verlag, 1995. 41, 50, 58, 67, 76, 84

[138] VISHWANATHAN, S. V. N., SUN, Z., THEERA-AMPORNPUNT, N., AND VARMA, M. Multiple kernel learning and the smo algorithm. In *In Proc. Neural Information Processing System* (2010), pp. 2361–2369. 84

[139] VON LUXBURG, U., AND SCHOLKOPF, B. Statistical learning theory: Models, concepts, and results. *Handbook of the History of Logic 10* (2011). 14, 15

[140] WIKIPEDIA. http://en.wikipedia.org/wiki/ugly_duckling_theorem. 20

[141] WOOD, S., YAO, X., KRISHNAMURTHI, K., AND DANG, L. Language identification for printed text independent of segmentation. In *International Conference on Image Processing* (1995), pp. 428–431. 26

[142] WU, T., LIN, C., AND WENG, R. C. Probability estimates for multi class classification by pair wise coupling. 59, 68, 69, 76

[143] ZHANG, B., AND SRIHARI, S. N. Analysis of handwriting individuality using word features. In *International Conference on Document Analysis and Recognition* (2003), pp. 1142–1146. 29

[144] ZHENG, Y., LI, H., AND DOERMANN, D. Machine printed text and handwriting identification in noisy document images. *IEEE Trans. Pattern Analysis and Machine Intelligence 26*. 25, 38, 45, 46

[145] ZHOU, L., LU, Y., AND TAN, C. L. Bangla/english script identification based on analysis of connected component profiles. In *In International Workshop on Document Analysis System,(DAS 2006)* (2006), pp. 243–254. 26, 64

[146] ZHU, Y., TAN, T., AND WANG, Y. Optical font recognition using typographical features. *IEEE Transactions on PAMI 23* (2001), 1192–1200. 29, 81, 86

[147] ZOIS, E. N., AND ANASTASSOPOULOS, V. Morphological waveform coding for writer identification. *Pattern Recognition 33*, 3 (2000), 385–398. 27, 29, 47

[148] ZRAMDINI, A., AND INGOLD, R. Optical font recognition using typographical features. *IEEE Transactions on PAMI 20* (1998), 887–882. 29, 81, 86

Appendix A

Additional Information

Writer Profile:- All the writers came from a varied socio-economic background and profession and were 18-65 years old. There was also no gender biased-ness. Example of a training and testing image in Bengali script from one of our writer is shown in Fig. (A.1) and Fig. (A.2).

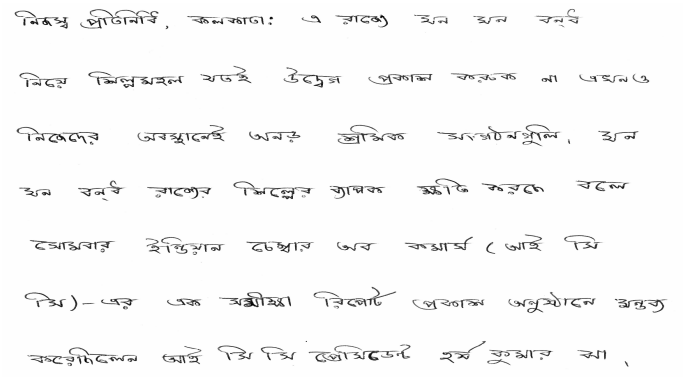

Figure A.1: Training image from one of our writer.

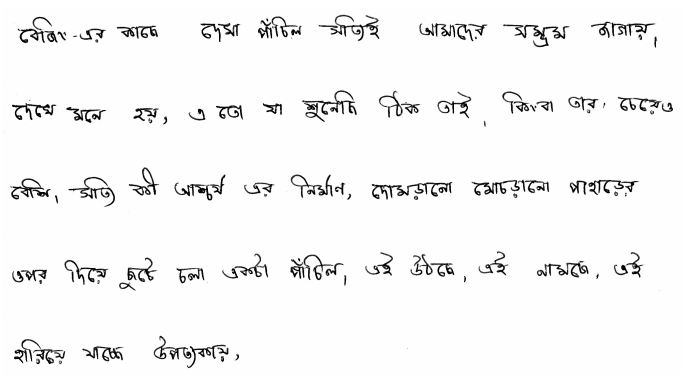

Figure A.2: Corresponding test image from the same writer.

Font Samples:- Samples of document with similar text written in two different fonts are shown here in Fig. (A.3).

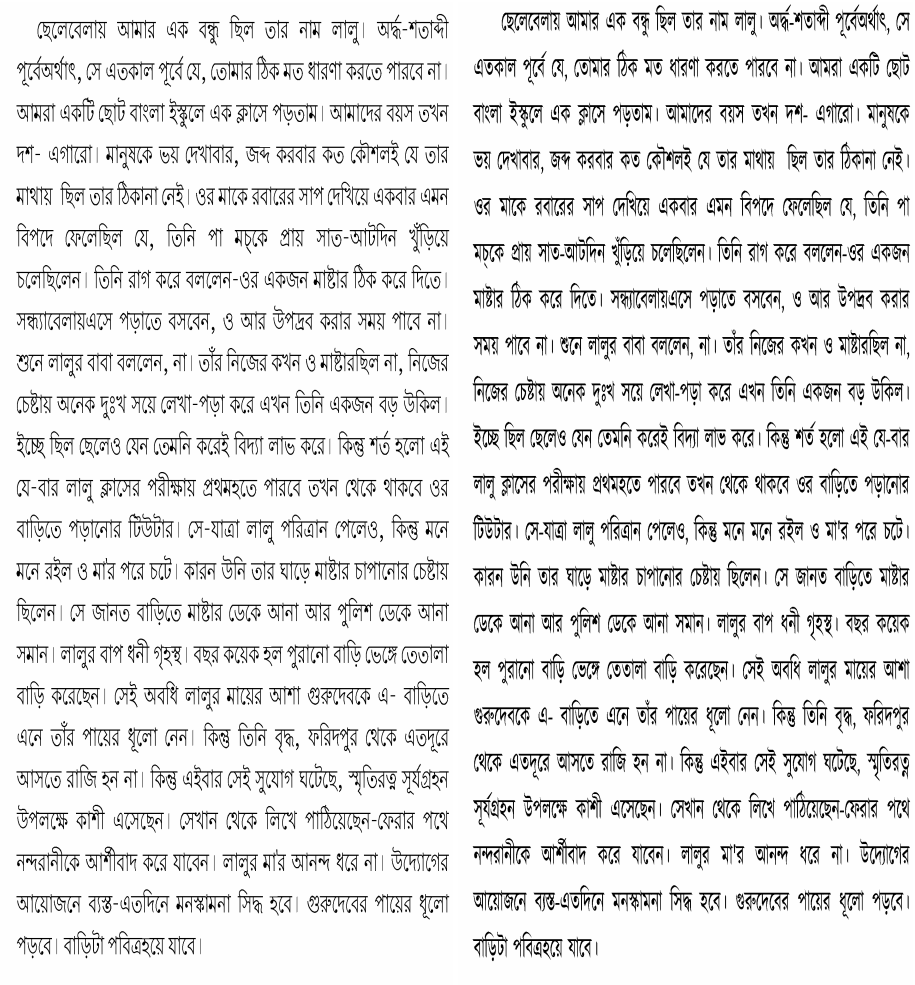

Figure A.3: Example of two different document images in our corpus in two different fonts.