An Approach for Script Identification from Official Indic Scripts

Thesis submitted

by

Sk Md Obaidullah

DOCTOR OF PHILOSOPHY (Engineering)

Department of Computer Science & Engineering
Faculty Council of Engineering & Technology
Jadavpur University
Kolkata-700032
India

December, 2016

An Approach for Script Identification from Official Indic Scripts

Thesis submitted by

Sk Md Obaidullah

DOCTOR OF PHILOSOPHY (Engineering)

under the supervision of

Prof. Kaushik Roy

Department of Computer Science
West Bengal State University
Kolkata-700126, India

and

Dr. Nibaran Das

Department of Computer Science & Engineering
Jadavpur University
Kolkata-700032, India

December, 2016

INDEX NO.44/14/E

1. **Title of the Thesis**: An Approach for Script Identification from Official Indic Scripts

2. **Name, Designation & Institute of the supervisors**

   i. **Prof. Kaushik Roy**
   Professor,
   Department of Computer Science
   West Bengal State University
   Kolkata-700126

   ii. **Dr. Nibaran Das**
   Assistant Professor,
   Department of Computer Science and Engineering,
   Jadavpur University
   Kolkata-700032

3. **List of publications:**

   a. Journal Publications

   i. **Sk Md Obaidullah,** K C Santosh, C Halder, N Das, K Roy "Automatic Indic script identification from handwritten documents – page, block, line and word-level approach", in Journal of Machine Learning and Cybernetics (JMLC), Springer, doi: 10.1007/s13042-017-0702-8, 2017
   ii. **Sk Md Obaidullah,** C Halder, K C Santosh, N Das, K Roy, "PHDIndic_11: Page-level handwritten document image dataset of 11 official Indic scripts for script identification", in Multimedia Tools and Applications (MTAP), Springer, doi:10.1007/s11042-017-4373-y, 2017
   iii. **Sk Md Obaidullah,** C Goswami, K C Santosh, C Halder, N Das, K Roy "Separating Indic scripts with 'matra' as a precursor for effective handwritten script identification in multi-script documents", in International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI), World Scientific, vol. 31, no. 4, 1753003 (17 pages), 2017
   iv. **Sk Md Obaidullah,** K C Santosh, C Halder, N Das, K Roy, "Word-Level Multi-Script Indic Document Image Dataset and Baseline Results on Script Identification", in International Journal of Computer Vision and Image Processing (IJCVIP), IGI Global, vol. 7, no. 2, pp. 81-94, 2017

v. **Sk Md Obaidullah,** C Halder, K. C. Santosh, N Das, K Roy, "Automatic Line-level Script Identification from Handwritten Document Images – A region-wise classification framework for Indian subcontinent", in Malaysian Journal of Computer Science (MJCS), 2016 (in press)

vi. **Sk Md Obaidullah,** C Halder, N Das, K Roy, "A New Dataset of Word-level Offline Handwritten Numeral Images from Four Official Indic Scripts and Its Benchmarking using Image Transform Fusion", in International Journal of Intelligent Engineering Informatics (IJIEI), Inderscience, vol. 4, no. 1, pp. 1-20, 2016

vii. **Sk Md Obaidullah,** C Halder, N Das, K Roy**,** "Bangla and Oriya Script Lines Identification from Handwritten Document Images in Tri-script Scenario", in International Journal of Service Science, Management, Engineering, and Technology (IJSSMET), IGI Global, vol. 7, issue 1, article 3, pp. 43-60, 2016

viii. **Sk Md Obaidullah,** C Halder, N Das, K Roy, "Numeral Script Identification from Handwritten Document Images", in Procedia Computer Science Journal, Elsevier**,** vol. 54C (2015), pp. 585-594, 2015

ix. **Sk Md Obaidullah,** N Das, K Roy, "Convolution Based Technique for Indic Script Identification from Handwritten Document Images", in International Journal of Image, Graphics and Signal Processing, MECS Publisher, vol. 7 no. 5, pp. 49-57, 2015

x. **Sk Md Obaidullah,** A Mondal, N Das, K Roy, "Script Identification from Printed Indian Document Images and Performance Evaluation using Different Classifiers", in Applied Computational Intelligence and Soft Computing, Hindawi Publishing Corporation, vol. 2014, Article ID 896128, 2014

xi. **Sk Md Obaidullah,** S K Das, K Roy, "A System for Handwritten Script Identification From Indian Document", in Journal of Pattern Recognition Research (JPRR), vol. 8, no. 1, 2013

xii. **Sk Md Obaidullah,** K C Santosh, N Das, C Halder, K Roy, "Handwritten Indic Script Identification in Multi-script Document Images: A Survey", in International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI), World Scientific, 2017 **(revised version submitted)**

b. Conference Publications

i. **Sk Md Obaidullah,** K C Santosh, C Halder, N Das, K Roy "Word-level thirteen official Indic languages database for script identification in multi-script documents", in International Conference on Recent Trends in Image Processing and Pattern Recognition (RTIP2R 2016), Karnataka, India, 2016

ii. **Sk Md Obaidullah,** C Goswami, K C Santosh, C Halder, N Das, K Roy, "Separating Indic scripts with `matra' -- a precursor to script identification in multi-script documents", in IAPR International Conference on Computer Vision & Image Processing (CVIP 2016), Roorkee, India, 2016

iii. **Sk Md Obaidullah,** C Halder, N Das, K Roy**,** "PWDB_13: A Corpus of Word-level Printed Document Images for Thirteen Official Indic Scripts", in 4th International Conference on Frontiers on Intelligent Computing - Theory and Applications (FICTA 2015), Durgapur, India, 2015

iv. **Sk Md Obaidullah,** C Halder, N Das, K Roy**,** "Visual Analytics based Technique for Handwritten Indic Script Identification – A Greedy Heuristic Feature Fusion Framework", in 4th International Conference on Frontiers on Intelligent Computing - Theory and Applications (FICTA 2015), National Institute of Technology, Durgapur, India, 2015

v. **Sk Md Obaidullah,** C Halder, N Das, K Roy**,** "An Approach for Automatic Indic Script Identification from Handwritten Document Images", 2nd Doctoral Symposium on Applied Computation and Security Systems (ACSS 2015), Calcutta, India, 2015

vi. **Sk Md Obaidullah,** C Halder, N Das, K Roy**,** "Handwritten Indic Script Identification from Document Images - A Statistical Comparison of Different Attribute Selection Techniques in Multi-classifier Environment", in 2nd International Conference on Computer and Communication Technologies (IC3T 2015), Hyderabad, India, 2015

vii. **Sk Md Obaidullah,** C Halder, N Das, K Roy**,** "A Corpus of Word-level Offline Handwritten Numeral Images from Official Indic Scripts", in 2nd International Conference on Computer and Communication Technologies (IC3T 2015), Hyderabad, India, 2015

viii. **Sk Md Obaidullah,** C Halder, N Das, K Roy, "Indic Script Identification from Handwritten Document Images – An Unconstrained Block-level Approach", in 2nd International Conference on Recent Trends in Information Systems (ReTIS 2015), Jadavpur, Kolkata, India, 2015

ix. **Sk Md Obaidullah,** R Karim, S Shaikh, C Halder, N Das, K Roy**,** "Transform Based Approach for Indic Script Identification

from Handwritten Document Images", in 3rd International Conference on Signal Processing, Communications and Networking (ICSCN 2015), Chennai, India, pp. 1-7, 2015

x. **Sk Md Obaidullah,** Z Rahaman, N Das, K Roy, "Development of Document Image Database for Offline Handwritten Indic Script Identification - A State-of-the-art", in International Conference on Pattern Recognition and Multimedia Signal Processing (ICPRMSP 2015), Tamilnadu, 2015

xi. **Sk Md Obaidullah,** N Das, K Roy, "Gabor Filter Based Technique for Offline Indic Script Identification from Handwritten Document Images", in International Conference on Devices, Circuits and Communication (ICDCCom 2014), Mesra, Ranchi, India, 2014.

xii. **Sk Md Obaidullah,** N Das, K Roy, "Offline Handwritten Script Identification from Eastern Indian Document Images using Logistic Model Tree", in 2014 International Conference on Intelligent Computing, Communication and Devices (ICCD 2014), Bhubaneswar, India, 2014

xiii. **Sk Md Obaidullah,** A Mondal, K Roy, "Structural Feature Based Approach for Script Identification from Printed Indian Document", in International Conference on Signal Processing and Integrated Network (SPIN 2014), Noida, India, 2014.

xiv. **Sk Md Obaidullah,** K Roy, N Das, "Comparison of Different Classifiers for Script Identification from Handwritten Document", in International Conference on Signal Processing, Computing and Control (ISPCC 2013), Shimla, India, 2013

xv. K Roy, S K Das, **Sk Md Obaidullah**, "Script Identification from Handwritten Document", in IEEE National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIGP 2011), Hubli, Karnataka, India, 2011

4. List of Patents: None

5. List of Presentations in International / National Conference

i. **Sk Md Obaidullah,** K C Santosh, C Halder, N Das, K Roy "Word-level thirteen official Indic languages database for script identification in multi-script documents", in International Conference on Recent Trends in Image Processing and Pattern Recognition (RTIP2R 2016), Karnataka, India, 2016

ii. **Sk Md Obaidullah,** C Goswami, K C Santosh, C Halder, N Das, K Roy, "Separating Indic scripts with `matra' -- a precursor to script identification in multi-script documents", in IAPR International

Conference on Computer Vision & Image Processing (CVIP 2016), Roorkee, India, 2016

iii. **Sk Md Obaidullah,** C Halder, N Das, K Roy, "Indic Script Identification from Handwritten Document Images – An Unconstrained Block-level Approach", in 2nd International Conference on Recent Trends in Information Systems (ReTIS 2015), Jadavpur, Kolkata, India, 2015

iv. **Sk Md Obaidullah,** C Halder, N Das, K Roy, "An Approach for Automatic Indic Script Identification from Handwritten Document Images", 2nd Doctoral Symposium on Applied Computation and Security Systems (ACSS 2015), Calcutta, India, 2015

v. **Sk Md Obaidullah,** R Karim, S Shaikh, C Halder, N Das, K Roy**,** "Transform Based Approach for Indic Script Identification from Handwritten Document Images", in 3rd International Conference on Signal Processing, Communications and Networking (ICSCN 2015), Chennai, India, pp. 1-7, 2015

vi. **Sk Md Obaidullah,** N Das, K Roy, "Gabor Filter Based Technique for Offline Indic Script Identification from Handwritten Document Images", in International Conference on Devices, Circuits and Communication (ICDCCom 2014), Mesra, Ranchi, India, 2014.

vii. **Sk Md Obaidullah,** N Das, K Roy, "Offline Handwritten Script Identification from Eastern Indian Document Images using Logistic Model Tree", in 2014 International Conference on Intelligent Computing, Communication and Devices (ICCD 2014), Bhubaneswar, India, 2014

viii. **Sk Md Obaidullah,** A Mondal, K Roy, "Structural Feature Based Approach for Script Identification from Printed Indian Document", in International Conference on Signal Processing and Integrated Network (SPIN 2014), Noida, India, 2014.

ix. **Sk Md Obaidullah,** K Roy, N Das, "Comparison of Different Classifiers for Script Identification from Handwritten Document", in International Conference on Signal Processing, Computing and Control (ISPCC 2013), Shimla, India, 2013

x. K Roy, S K Das, **Sk Md Obaidullah**, "Script Identification from Handwritten Document", in IEEE National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIGP 2011), Hubli, Karnataka, India, 2011

Department of Computer Science and Engineering
Jadavpur University
Kolkata-700032

Certificate from the Supervisors

*This is to certify that the thesis entitled "**An Approach for Script Identification from Official Indic Scripts**" submitted by **Sk Md Obaidullah,** who got his name registered on 2nd November 2014 for the award of Ph.D. (Engg.) degree of Jadavpur University, is absolutely based upon his own work under the supervision of Prof. Kaushik Roy, Department of Computer Science, West Bengal State University and Dr. Nibaran Das, Department of Computer Science & Engineering, Jadavpur University and that neither his thesis nor any part of the thesis has been submitted for any degree/diploma or any other academic award anywhere before.*

1……
(Prof. Kaushik Roy)

2…….....
(Dr. Nibaran Das)

Dedicated to my beloved parents,

Abdul Matin

and

Sofia Begum

Acknowledgements

First and foremost, praises and thanks to the Almighty, the most beneficent and the merciful for his shower of blessings in course of this thesis work. You have given me the power to believe in myself and to pursue my dreams.

I would like to take this opportunity to express my sincere gratitude to Prof. Kaushik Roy, Professor, Dept. of Computer Science, West Bengal State University, Kolkata and Dr. Nibaran Das, Assistant Professor, Dept. of Computer Science & Engg., Jadavpur University, Kolkata, my supervisors, for their guidance, constant support and monitoring throughout the course of this work. They were not just my teachers but also my friends and philosophers. Their inspiration was a real asset to me during this thesis work. I would like to convey my thankfulness to Mrs. Moumita Roy and Mrs. Shyamali Das for their cordial support whenever I had visited my supervisor's house. In this special moment I remember the sweet girl Kalpita, master Nandish, master Kalpit and convey my affectionate love to them.

I sincerely thank Prof. K. C. Santosh, University of South Dakota, USA, for his valuable direction which helped me to have the vision of the right approach to focus the area that needed much attention.

I would like to thank Prof. Mita Nasipuri, Professor, Dept. of Computer Science & Engg., Jadavpur University, Kolkata, for her support and blessings. I am thankful to Prof. Jaya Sil, Professor, Dept. of Computer Science & Tech., IIEST, Kolkata, for her valuable comments to improve the quality of my work.

I am thankful to Prof. Ujjwal Maulik, Head, Dept. of Computer Sc. & Engg., Jadavpur University for allowing me to carry out the research work at Jadavpur University. I am also thankful to Prof. Debesh Das, Ex. Head, Dept. of Computer Sc. & Engg., Jadavpur University for allowing me to registrar at Jadavpur University.

I am thankful to Hon'ble Vice Chancellor of Aliah University, Prof. Abu Taleb Khan for his encouragement and support to carry out the work beside my regular teching and administrative activites at Aliah University. I would like to thank Ex. Vice Chancellor of Aliah University, Prof. Samsul Alam for his encouragement. I am also thankful to my colleagues at Aliah University for their cooperation. Especially, I would

like to remember Dr. Zeenat Rehena, for her positive words and inspiration. I am thankful to my students at Aliah University for their love and affection.

I am thankful to Ms. Tithi Mitra Chowdhury and Mr. Sandipan Chowdhuri of Jadavpur University for their immense help for my manuscript correction. I am also thankful to Mr. Chayan Halder, Mr. Himadri Mukherjee, Ms. Ankita Dhar, Ms. Payel Rakshit of West Bengal State University for their help and support during the course of my work.

Today, I remember all my respected teachers starting from my nursery school upto university, for their blessings, without which I couldn't have seen this day. I am thankful to them. I am also thankful to my friends for their encouragement. Especially I remember Sutapa for her encouragement and support.

In this work, I had to collect lot of handwritten samples from different people of different parts of India. I am very much thankful to all those persons who had contributed their handwriting voluntarily to prepare the dataset used in this thesis work.

I am thankful to my beloved brother Asad and my uncle Mr. Abdur Rakib for their immense help and constant support. I am also thankful to my in-laws, especially Mr. Rezoan Ali for their blessings, constant support and encouragements.

I am indebted to my parents, Mr. Abdul Matin, and Mrs. Sofia Begum, for their constant inspiration and love during my work. I remember and acknowledge their sacrifice to grow me up. Whatever, I am today, is just because of them.

I am really thankful to my two little darlings, my five year old son Saadat and one year old daughter Sara. I was not able to give them quality time as a dad but still they didn't mind, Saadat happily told me always "papa once you complete your work then we will play". Finally, I would like to extend my heartfelt gratitude to my wife Sumi for her constant support, encouragement, patience and love during this thesis work.

Jadavpur University — Sk Md Obaidullah

December, 2016.

Contents

List of Tables

List of Figures

CHAPTER ONE

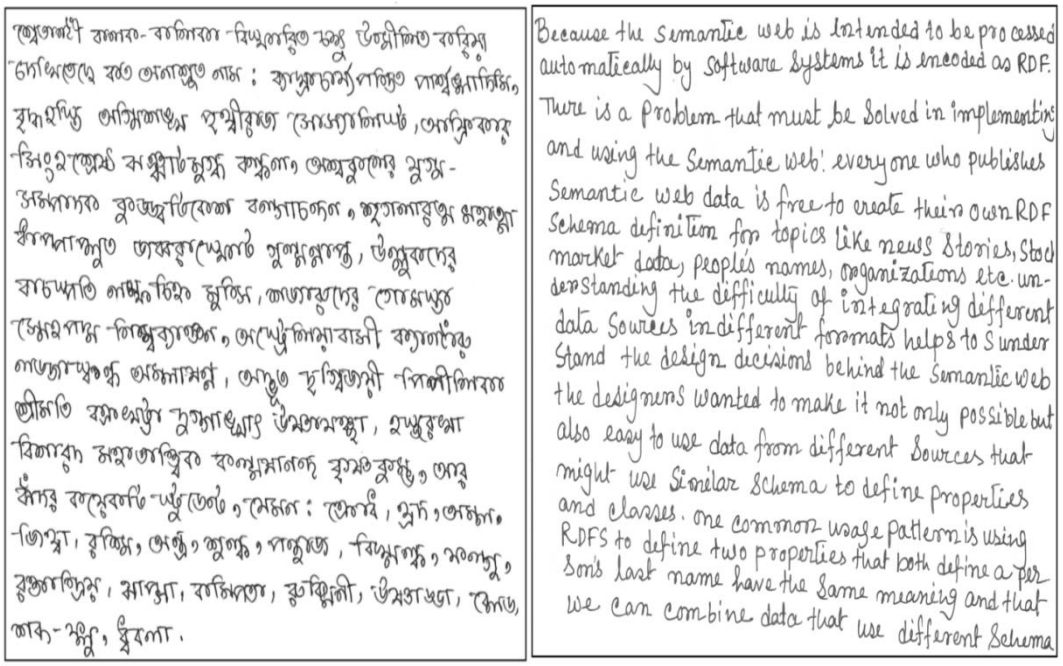
INTRODUCTION

Document image processing is one of the emerging areas of research where different techniques are applied to document images in order to obtain an editable text. The main objective of this thesis is to identify the scripts from Indic multi-script document images. This work can further be used to automate the optical character recognition system in a multi-script environment. The introductory chapter explains the background, highlights on the existing works, and describes the objective, contributions & structure of the thesis. The following section provides the preamble where the background of the addressed problem has been briefed.

1.1 PREAMBLE

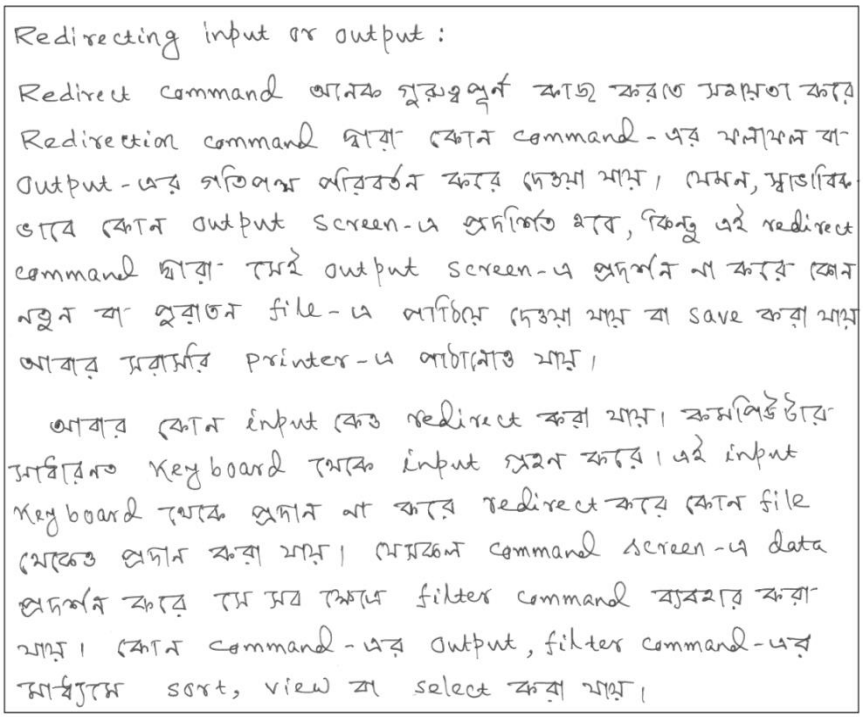
The dream of making a 'paperless world' will become a reality if an overwhelming volume of physical documents can be converted into its digital form. Researchers are working towards achieving this goal by developing several techniques for automatic processing of text document images. The initial step for developing such an auto processing system is digitization of the document files. Digitized text documents have several advantages, like indexing and sorting of large volumes of data, for efficient search operation and retrieval. Digitization of text documents can even ensure their preservation since digital documents will be protected from any kind of damage, degradation, the later being a common scenario in physical documents. In the past, researchers world wide have exploited this possibility of digitization in an attempt to develop an image-to-alphanumeric text conversion system. Such a system is well renowned – popularly known as Optical Character Recognizer (OCR) [1] [2]. The history of character recognition dates back to the year 1870 when the retina scanner

system was invented by Carey [1], which was a photocell based image transmission system. In the late 1960's, soon after the invention of the digital computer, scientists realized the necessity of OCR for document processing system. As per record, the first commercialized OCR was developed by IBM to read the special font of IBM machines [1]. Practitioners all around the world have, since then, been intrigued by this emerging field of research, which encompasses innumerable multi-faceted applications. The field is maturing day by day, by encapsulating smart capabilities in the system like, ability to handle complex documents which may contain text, graphics, mathematical symbols, historical documents with degraded quality and noise, color images etc. Smart ready to use commercial systems have been developed, whose applications include reading aids for blinds and automatic postal document sorter to name a few. Researchers belonging to the OCR community are now focusing on the development of efficient techniques for computerized document processing systems. However, in a multi-script country like India (having 11 scripts and 22 languages) [3], the prerequisite for these techniques is an adequate knowledge of the particular script from which the language has been originated. Thus development of a script identification system is essential in terms of the concerned research task. In our day to day life, we come across various multi-script documents such as postal documents, filled up pre-printed application forms, railway reservation forms, etc. Figure 1.1 shows examples of few such multi-script documents, where part (a) shows a single document written using a single script (Bangla and Roman as an example), (b) shows a single document written using multiple scripts (combination of Bangla and Roman scripts) and (c) shows two real life multi-script postal document images (in the first image, the text has been written using Bangla and Roman scripts, and in the second image both scripts have been utilised to write the address block). The first image showcases Inter-document script identification, whereas the latter one falls under the category of Intra-document script identification. As it is evident from the provided images, script identification is an essential module before feeding the document image to language/script specific OCR.



Because the semantic web is intended to be processed automatically by software systems it is encoded as RDF. There is a problem that must be solved in implementing and using the Semantic Web: everyone who publishes Semantic web data is free to create their own RDF Schema definition for topics like news stories, Stock market data, peoples names, organizations etc. understanding the difficulty of integrating different data sources in different formats helps to Sunder Stand the design decisions behind the Semantic web the designers wanted to make it not only possible but also easy to use data from different sources that might use Similar Schema to define properties and classes. one common usage pattern is using RDFS to define two properties that both define a person's last name have the same meaning and that we can combine data that use different Schema

(a)

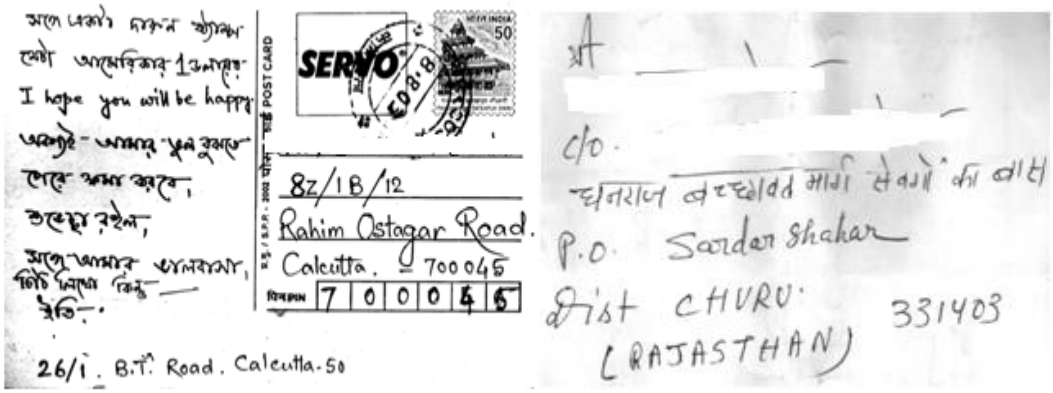


Redirecting input or output :

Redirect command আমাদের গুরুত্বপূর্ণ কাজ করতে সহায়তা করে। Redirection command দ্বারা কোন command-এর সম্পাদন বা output-এর গতিপথ পরিবর্তন করে দেওয়া যায়। যেমন, স্বাভাবিক ভাবে কোন output screen-এ প্রদর্শিত হবে, কিন্তু এই redirect command দ্বারা সেই output screen-এ প্রদর্শন না করে কোন নতুন বা পুরাতন file-এ পাঠিয়ে দেওয়া যায় বা save করা যায় আবার সরাসরি printer-এ পাঠানোও যায়।

আবার কোন input কেও redirect করা যায়। কমপিউটার সাধারণত Keyboard থেকে input গ্রহন করে। এই input Keyboard থেকে প্রদান না করে redirect করে কোন file থেকেও প্রদান করা যায়। সেক্ষেত্রে command screen-এ data প্রদর্শন করে সে সব ক্ষেত্রে filter command ব্যবহার করা যায়। কোন command-এর output, filter command-এর মাধ্যমে sort, view বা select করা যায়।

(b)



(c)

Figure 1.1 Examples of multi-script documents (a) Single document written in a single script (Bangla and Roman script images are shown) (b) Single document written in different scripts (a single handwritten page contains both Bangla and Roman texts) [4] (c) Two multi-script postal document images are shown (in the first image, text is written using Bangla and Roman script and in the second one the address block has been written using Roman and Bangla scripts) [5]

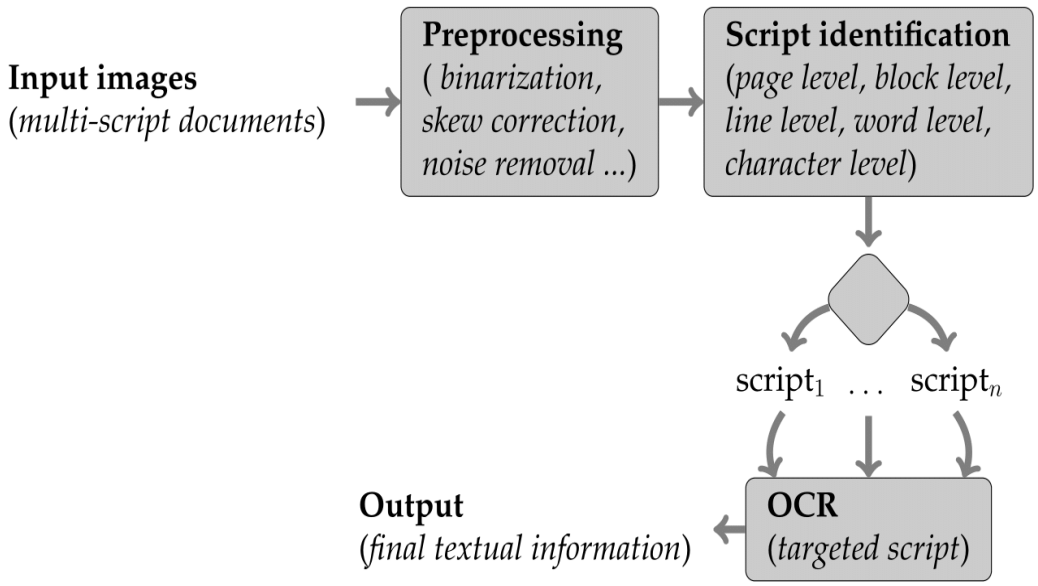


Figure 1.2 Block diagram of a multi-script document processing system showing different modules

A block diagram of a multi-script document processing system is depicted in Figure 1.2. Initially, various multi-script document images are provided as input data. Then, basic pre-processing operations like noise removal, foreground-background separation, skew

detection and correction, segmentation are performed. The next step performs script identification at page/block/line/word/character level, where, specific script type is produced as an output. Then script dependent OCR is called from OCR bank and final textual information is generated.

Figure 1.3 A map showing different scripts for different states [6]

1.2 SCRIPTS AND LANGUAGES OF INDIA

A script can be described as a set of graphemes which are used to write a single language or a class of languages. Sometime languages and scripts are synonymous. Examples of such languages/scripts are Oriya, Tamil, Telugu, Urdu, Gujarati, Kannada. On the other hand, scripts like Devanagari, Bangla, Roman are used by more than one languages. For example, Devanagari script is used by the languages like Bodo, Konkani, Marathi, Maithili, Nepali, Sanskrit, Sindhi, Hindi etc., Roman script is used by English and Santali languages, Bangla script is used to write Bangla, Assamese and Manipuri

languages. Figure 1.3 shows a map where different state names are written using different state specific scripts. Table 1.1 portraits on official Indic languages and scripts [3] [6] [7] [8] [9] [10] with information about the language family, use of the script of the particular language, different locations in India, where such languages/scripts are used as a communication medium and approximate number of the population under the particular language/script.

Table 1.1 Official languages of India as per 8th schedule of the constitution and different scripts used to write them [3] [11]

Language	**Belonging family**	**Script used**	**Communication medium of major Indian states**	**Population (Million)**
1. Hindi	Indo-European	Devanagari	Uttar Pradesh, Himachal Pradesh, Uttaranchal, Delhi, Rajasthan, Punjab, Madhya Pradesh and northern Bihar	182
2. Marathi			Maharashtra	68.1
3. Konkani			Goa, Karnataka, Kerala, Dadra and Nagar Haveli. Parts of Maharashtra	7.6
4. Sanskrit			It uses a liturgical language	0.03
5. Sindhi			Uttar Pradesh, Delhi, Rajasthan, Madhya Pradesh, Gujarat, Maharashtra, Andhra Pradesh, Tamil Nadu, Orissa, Bihar	21.4
6. Nepali			Parts of West Bengal, Sikkim, Arunachal Pradesh, Manipur, Nagaland, Meghalaya, Tripura, Mizoram, Assam, Bihar, Himachal Pradesh, Uttaranchal, Uttar Pradesh, Haryana	13.9
7. Maithili			Bihar	34.7
8. Bodo	Sino-Tibetan		Parts of Assam, Manipur, Meghalaya and Darjeeling, West Bengal	0.5

9. Bangla	Indo-European	Bangla	West Bengal, Tripura, Bihar, Parts of Jharkhand, Meghalaya, Assam, Nagaland, Mizoram	181
10. Assamese			Assam, Arunachal Pradesh, Meghalaya, West Bengal	16.8
11. Manipuri	Sino-Tibetan		Manipur, Karimganji and Cachar of Assam, West and North Tripura districts, Nagaland, West Bengal, Uttar Pradesh	13.7
12. Telugu	Dravidian	Telugu	Andhra Pradesh	69.8
13. Tamil		Tamil	Tamil Nadu	65.7
14. Urdu	Indo-European	Urdu/ Perso-Arabic	Uttar Pradesh, Uttaranchal, Delhi, Rajasthan, Punjab, Madhya Pradesh and northern Bihar, West Bengal	60.6
15. Kashmiri			Jammu and Kashmir	5.6
16. Gujarati		Gujarati	Gujarat, Rajasthan, Madhya Pradesh, Maharashtra, Karnataka	46.5
17. Malayalam	Dravidian	Malayalam	Kerala, Laccadive Islands	35.9
18. Oriya	Indo-European	Oriya	Orissa, Assam, Parts of Jharkhand, Chhattisgarh, West Bengal, Andhra Pradesh	31.7
19. Kannada	Dravidian	Kannada	Tamil Nadu, Andhra Pradesh and Maharashtra	3.63
20. Punjabi	Indo-European	Gurumukhi	Punjabi	1.05
21. Dogri		Gurumukhi/ Devanagari	Area between Chenab and Ravi rivers in Jammu and Kashmir, Chandigarh	3.8
22. Santali	Austro-Asiatic	Roman	Assam, Mizoram, Tripura Bihar, Orissa, West Bengal	6.2

1.2.1 CHARACTERISTICS

It is already mentioned that, in India, there are 23 different languages (including English) and 11 official scripts (including Roman) are used to write them. These scripts vary from one another in visual and structural appearances. Some of the key observations about these scripts are as follows:

- Presence of 'matra' or 'shirorekha', a horizontal line on the upper part of the words or sentence connecting more than one character resulting into a larger connected component. Examples of 'matra'-based scripts are Bangla and Devanagari. Figure 1.4 shows the presence of 'matra' or 'shirorekha' in Bangla and Devanagari scripts. The same is absent in Urdu and Oriya scripts [12].
- At a first glance, Devanagari and Gurumukhi scripts look almost similar. But the characters in Devanagari are more circular in nature compared to Gurumukhi and Bangla. Whereas Gurumukhi script contains a number of half length vertical lines which are a prominent distinguishable feature from the other two [8] [13] [14]. Gujarati script has also visual similarity like Devanagari, but the number of loops is more in the former one.
- Oriya and Malayalam scripts have components of a more circular shape than others [7].
- Urdu script contains maximum dot ('.') like small components [7] as shown in Figure 1.5. This script looks quite unlike than other Indic scripts. Many characters of Urdu contain directional strokes with orientation 75^0.
- Roman scripts contain many vertical, horizontal and slanting (45^0) strokes.
- The visual appearances of south Indian scripts are quite similar, compared to that of northan India. Most of the characters in Telugu and Kannada scripts look similar to each other. Considering Malayalam and Tamil scripts, the former has more round shape characters compared to the latter one. Out of 51 character set in Malayalam, in 27% cases presence of straight lines have been found. But for Tamil script, out of 36 characters, almost in 72% cases straight lines have been found. Many characters in Tamil contain Roman 'T' like a shape (as shown in Figure 1.6).
- Malayalam (or Tamil) and Telugu (or Kannada) scripts can be distinguished by the direction of their concavities. In Malayalam (or Tamil), concavities for most of the characters are present downwards, whereas the same lies upwards for Telugu (or Kannada) (as shown in Figure 1.7). Another graphic characteristic of Telugu (or Kannada) script is the presence of a head mark above few characters which is

known as ‘talakattu’ or ‘talekattu’ [15]. There is a slight difference between the Telugu and Kannada script based on the position of the head mark.

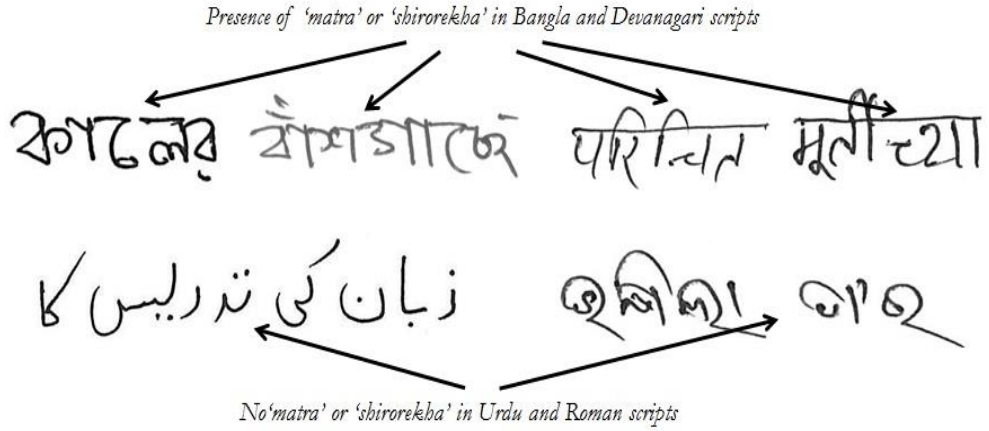


Figure 1.4 Presence of ‘matra’ or ‘shirorekha’ in Banagla and Devanagari scripts, the same is absent in Urdu and Oriya scripts

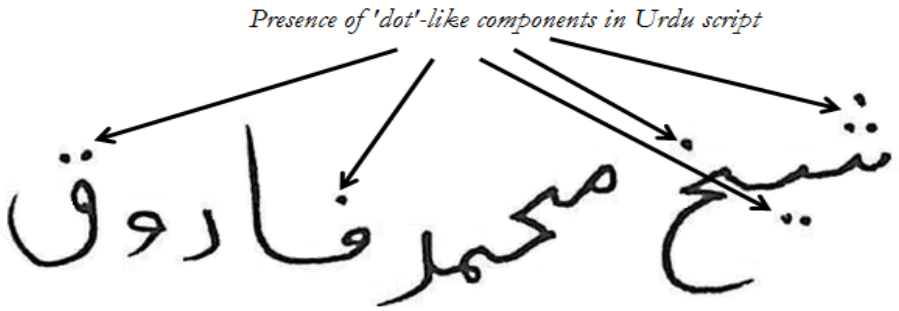


Figure 1.5 Presence of ‘dot’ symbol on top and bottom position of most of the Urdu script characters

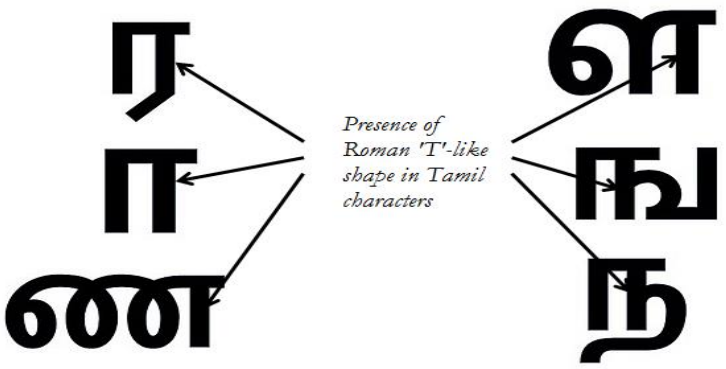


Figure 1.6 Few characters from the Tamil script where Roman ‘T’ like shape is found

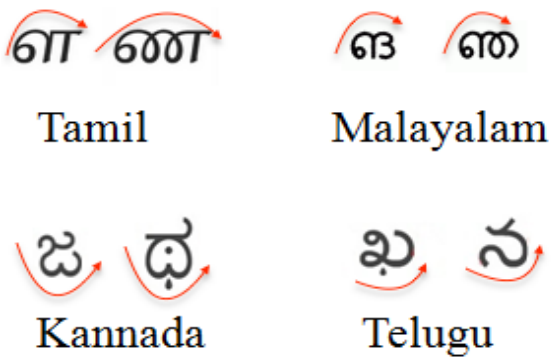


Figure 1.7 Direction of concavities for South Indian scripts. Malayalam (or Tamil) characters have downward concavities and Telugu (or Kannada) characters have upwards concavities

In order to understand the origin of Indic scripts, Figure 1.8 provides a tree diagram with the origin and examples of different scripts. We will restrict our discussion only in the Alphabetic writing system as it is our topic of interest in this thesis.

ALPHABETIC WRITING SYSTEM

An alphabet is a set of basic writing symbols which represent phonemes of a spoken language. The word alphabet is derived from the Greek. This system can be categorized into three major categories, namely Abjad, Abugida and True Alphabetic. Abjad is a very old writing system where one symbol per consonant is present. Demarcation of vowels is absent in this type of system. Some abjads, like Arabic and Hebrew, have markings for vowels as well. However, they use them only for special purposes, such as for teaching. Many scripts derived from abjads have been extended with vowel symbols and later become full alphabets [9]. Urdu, which is a popular script in many South Asian countries, is also used in many places of India. It falls under the catagory of Abjad writing system. Unlike Abjad, in Abugida, vowels are present along with the consonants. This system has several features like, vowel representation after consonant, initial vowel representation, inherent vowels, without vowels etc. The largest single group of Abugida is the Brahmic family of scripts, which is classified into three major categories namely Gupta, Kadamba and Grantha. All existing Indic scripts are descendants of the Brahmic alphabet. Today, they are used in most of the languages of

South Asia and mainland Southeast Asia with the exception of Malaysia and Vietnam. Southern Indic scripts fall under the Gupta family. They are primarily used in South India, Sri Lanka and Southeast Asia. On the other hand, North Indic scripts fall under the category of Kadamba and Grantha families. They are primarily used in Northern India, Nepal, Tibet and Bhutan. South Indic letters are generally round in shape, North Indic less so, with an exception of Oriya script. Most North Indic scripts have a horizontal line at the top known as 'matra' or 'shirorekha', with Gujarati and Oriya script as exceptions. South Indic scripts do not have any 'matra' or 'shirorekha'.

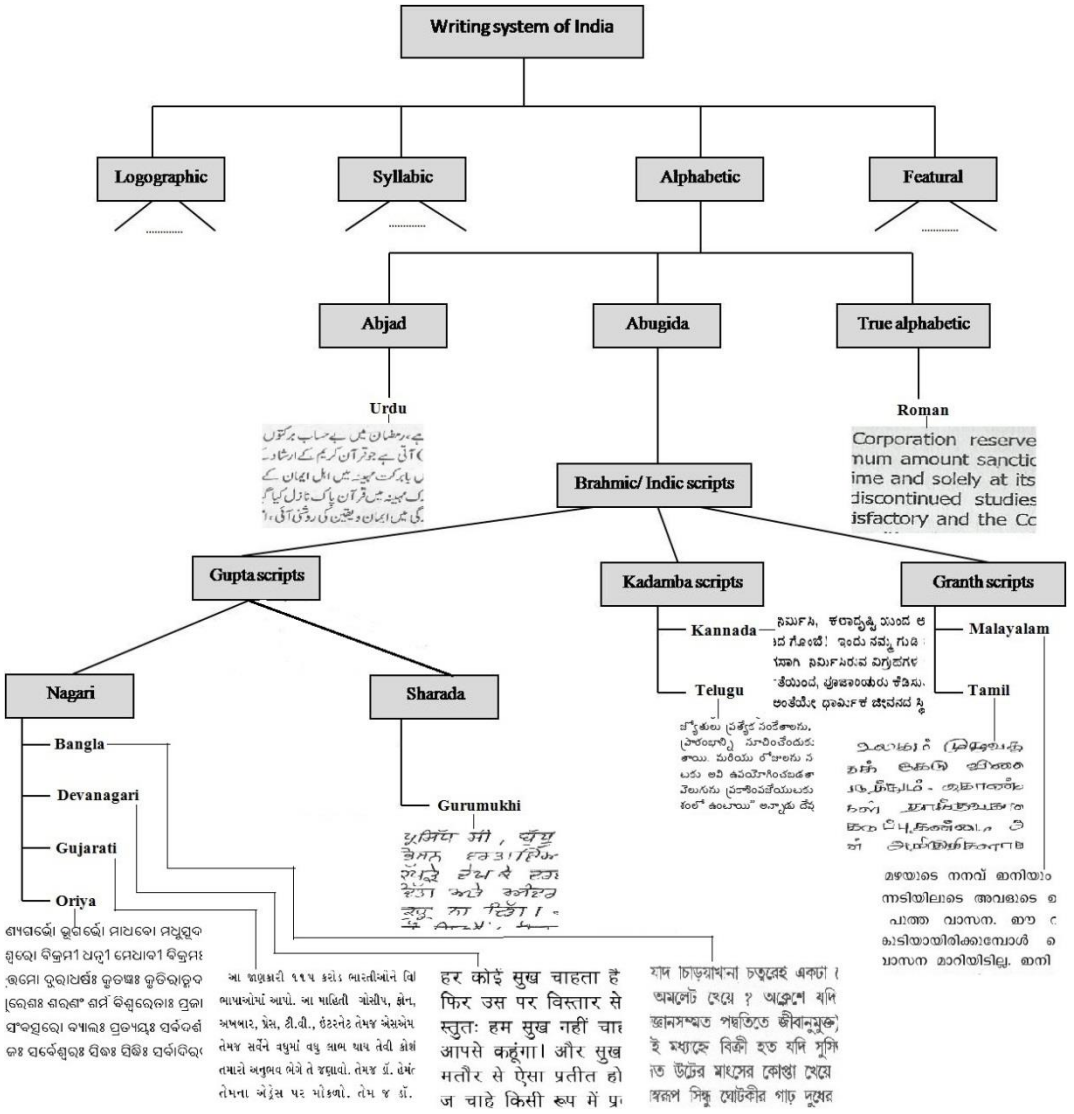


Figure 1.8 Writing System of Indic Scripts [9]

1.3 SCRIPT IDENTIFICATION TECHNIQUES

Figure 1.9 shows a tree diagram of high level categorization of different script identification techniques. In general, script identification techniques can be divided into two main categories based on raw data/image acquisition: offline and online. In offline system, inputs are provided in the form of images, whereas in the online catagory, inputs are considered to be ordered sequence of points. In case of online system, additional information regarding the stroke direction can be captured as one of the important feature values which is not available on the offline system, where a pre-captured image is provided as an input. Because of this additional information online script identification is apparently easier as compared to offline.

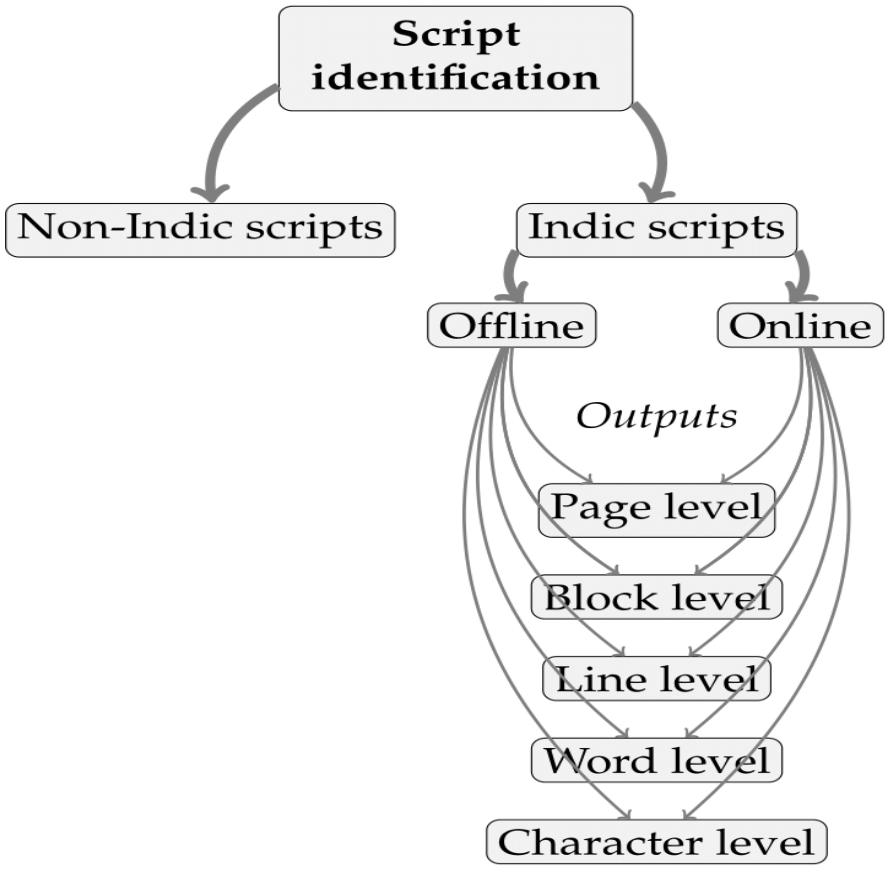


Figure 1.9 Catagorization of different script identification techniques

We further divide all the offline/online script identification techniques into five major categories, based on the segmentation scheme followed prior to the feature extraction. These are: (i) Page level script identification (ii) Block level script identification (iii) Line level script identification (iv) Word level script identification and (v) Character level script identification. Table 1.2 presents a script-wise distribution of handwritten script identification techniques. These works have been discussed in the following section.

Table 1.2 Related works on handwritten script identification (script wise distribution)

Script Name	**Methods**
1. Devanagari	Zhu *et al.* [16], Basu *et al.* [17], Singhal *et al.* [18], Hangarge and Dhandra [19], Rajput and Anita [20], Roy *et al.* [21], Sarkar *et al.* [22], Chanda *et al.* [23], Hangarge *et al.* [24], Singh *et al.* [25], Pardeshi *et al.* [26], Singh *et al.* [27], Singh *et al.* [28]
2. Bangla	Basu *et al.* [17], Singhal *et al.* [18], Hangarge and Dhandra [19], Kanoun *et al.* [29], Rajput and Anita [20], Zhou *et al.* [30], Roy *et al.* [31], Roy *et al.* [21], Sarkar *et al.* [22], Chanda *et al.* [23], Pardeshi *et al.* [26], Singh *et al.* [27], Singh *et al.* [28]
3. Roman	Hochberg *et al.* [32], Zhu *et al.* [16], Basu *et al.* [17], Singhal *et al.* [18], Rajput and Anita [20], Zhou *et al.* [30], Benjelil *et al.* [33], Roy *et al.* [31], Roy *et al.* [34], Roy *et al.* [21], Sarkar *et al.* [22], Roy and Pal [5], Chanda *et al.* [23], Hangarge *et al.* [24], Singh *et al.* [25], Pardeshi *et al.* [26], Singh *et al.* [27], Singh *et al.* [28]
4. Oriya	Roy and Pal [5], Chanda *et al.* [23], Pardeshi *et al.* [26], Singh *et al.* [27], Singh *et al.* [28]
5. Urdu	Basu *et al.* [17], Chanda *et al.* [23], Hangarge and Dhandra [19], Pardeshi *et al.* [26]
6. Tamil	Rajput and Anita [20], Chanda *et al.* [23], Hangarge *et al.* [24], Pardeshi *et al.* [26], Singh *et al.* [27]
7. Telugu	Singhal *et al.* [18], Rajput and Anita [20], Chanda *et al.* [23], Hangarge *et al.* [24], Pardeshi *et al.* [26], Singh *et al.* [27], Singh *et al.* [28]
8. Kannada	Rajput and Anita [20], Chanda *et al.* [23], Hangarge *et al.* [24], Pardeshi *et al.* [26]
9. Malayalam	Chanda *et al.* [23], Rajput and Anita [20], Hangarge *et al.* [24], Pardeshi *et al.* [26], Singh *et al.* [28]
10. Gurumukhi	Rajput and Anita [20], Chanda *et al.* [23], Rani *et al.* [14], Pardeshi *et al.* [26], Singh *et al.* [27]
11. Gujarati	Chanda *et al.* [23], Pardeshi *et al.* [26]

1.3.1 OFFLINE SCRIPT IDENTIFICATION TECHNIQUES

Offline script identification techniques have been discussed in the following section. These works have been categorized into different levels based on the segmentation scheme adopted before computing the actual features. These levels are namely: Page, Block, Line, Word and Character level.

PAGE-LEVEL SCRIPT IDENTIFICATION

The Page level approach ensures fast feature computation as it is completely segmentation free. The whole document page is considered as input, and then feature extraction techniques are applied to all the pages. Depending upon the features type, if needed component analysis is to be performed and feature values are computed for each component and then the average is obtained. In some cases, without analyzing components individually, the whole document is considered globally and the pages are converted into the frequency domain to compute different feature values. The reported works based on Page level script identification have been discussed in this section.

Hochberg *et al.* [32] perfomed connected component analysis for identifying six scripts namely Arabic, Chinese, Cyrillic, Devanagari, Japanese and Latin. Components were filtered based on pixel count in the bounding box. The components, which had bounding box height and width less than 3 pixels, were classified as small components. The total area of the bounding boxes was also considered whose size was considered as below 30 pixels. Following the similar approach, long and thin components were also identified. Once these components were identified, the mean and standard deviation of the bounding box height and width were measured. In the second phase of filtering, unusually large components were removed. Finally, using connected components and visual observations, a feature set was generated. This feature set included relative Y centroid, X centroid, number of white holes, sphericity, aspect ratio, etc. Finally the linear discriminant function (LDF) classifier was used for identifying a particular script. The classifier had been tested through writer sensitive cross validation. Using the same

set of features, neural network based classifier had also been used, even though reported results confirm that LDF performs best.

Zhu *et al.* [16] proposed a scheme based on shape codebook for identifying eight scripts namely Arabic, Chinese, Roman, Hindi, Japanese, Korean, Russian and Thai. In their work, a shape codebook had been constructed based on geometrically invariant feature types and indexed them based on structure of the codes. All the traditional script identification techniques mainly focus on finding sophisticated features or finding features from visual analysis of the document image. However, in this reported work, they tried to identify differences between texts collectively using the statistics of a large variety of generic, geometrically invariant feature types instead of selecting class specific features. After constructing the codebook, contour features were extracted by using a two step procedure. At first, edges using the Canny edge detector [35] were computed, which give precise localization and unique response to text content. Secondly, contour segments were grouped by connected components and fit them locally into line segments using an algorithm that broke a line segment into two parts, only when the deviation crossed certain threshold. Then, within each connected component, every triplet of connected line segments that started from the current segment was extracted. Then the dissimilarity measure had been computed. The overall dissimilarity between two contour features was quantified by the weighted sum of the distances in length and orientation. Finally, using multi-class SVM classifier an average successful classification rate of 95.6% had been achieved.

In a recent work, performed by Singh *et al.* [27], a texture based approach using modified log-Gabor filter to distinguish eight different scripts namely Bangla, Devanagari, Gurumukhi, Oriya, Tamil, Telugu, Urdu and Roman. During feature computation, 5 scales (ns=1, 2, 3, 4, and 5) and 6 orientations (no= 0^0, 30^0, 60^0, 90^0, 120^0, and 150^0) had been considered. Then, each filter was convolved with the input image to generate 30 different response matrices for the particular image. These response matrices were then converted into final feature vector generating a 30x240 dimensional feature vector, where the total number of document pages was 240.

Different classifiers namely Naïve Bayes, Simple logistic, MLP, SVM, Random forest, Bagging and Multi class classifiers were tested and Simple logistic showed an average accuracy of 95.57% for the concerned dataset. In this work, no justification is provided for using specific scales and orientation values. Actually, the process of designing the response matrices is dependent on the particular image processing application in hand. Besides scaling and orientations, there are other parameters to be considered such as: the minimum and maximum frequencies, the filter bandwidth, the number of orientations and the angular bandwidth. Another point is, here the authors had selected 160 (20 from each script) pages as training and 80 (10 from each script) for testing. But the confusion matrix shows outcome on the whole data set, i.e. 240 document pages, not on the test data set. It is not clear whether the authors have followed cross-validation or train-test splitting. It would have been more accurate if the authors had reported the outcome of their methods on the test dataset.

BLOCK-LEVEL SCRIPT IDENTIFICATION

In the block level script identification techniques, normally blocks of predefined size are extracted from the document images. This size can vary from 64 x 64, 128 x 128 to 512 x 512. Sometimes these extracted blocks of sub images require padding of white pixels as during the block extraction phase, in some blocks, some characters are attached to the boundary of the blocks. From the extracted blocks of sub images, feature values are computed.

Kanoun *et al.* [29] proposed a hybrid scheme for script identification from Arabic and Latin document images. Their work was designed for both printed and handwritten documents. In this approach, they collected morphology features based on global analysis of the text blocks. They also collected some local features based on geometrical analysis at line level and component level. During morphological analysis, they extracted connected components of text block and localized a reference line for each text line. They used some extractors for extracting morphological features like diacritic dots, occlusions, and 'alif' character. They considered other connected components as

traces. Diacritic dots and 'alif' character extractors were calculated based on some heuristic threshold. The threshold was fixed after carrying out some tests on their text blocks data set. Occlusion extractor was calculated based on interior contour detection of connected components. For each text line, they have detected diacritic dots and occlusion position (bottom or up) by comparison of coordinates between the last components and a reference line. Using the aforementioned methods, they obtained features like, diacritics, dots, numbers and their positions, occlusions number and their position, 'alif' characters and trace number. During geometrical analysis, measurement of the physical structure and textual entity had been carried out. They have obtained features like pixel density, eccentricity, spheroid on text lines and connected components. Finally, using KNN classifier they obtained a successful classification rate of 88% for Arabic handwritten text and 98% for Latin handwritten text.

In another work, Singhal *et al.* [18] used rotation invariant texture feature using multi-channel Gabor filtering and gray level co-occurrence matrix for feature extraction. In this way, variations in writing style, character size, interline and inter-word spacing problems could be tackled. During the pre-processing stage they performed denoising, thinning and pruning using basic morphological operations. After that, connectivity and linking process had been carried out for adjustment of the broken components. The text size normalization process had also been performed through adjustment of the text height, inter-word spacing and left-right justification. Then features were extracted using multi-channel Gabor filtering and gray level co-occurrence matrix. Finally, they used a multi-prototype classifier, which was a combination of K-means clustering, fuzzy C-means clustering and Probabilistic clustering methods. They reported an individual result of 90% for Devanagari, 86.6% for Bangla, 96.7% for Telugu and 93.3% for Latin script with an average of 91.64% overall accuracy rate.

Zhou *et al.* [30] proposed a line level script identification technique for Bangla and Roman languages using connected component analysis. At first, connected component labelling had been carried out. Then, they selected meaningful connected component based on pixel area value. In this way, absolutely very small element deletion, relatively

small elements deletion and relatively large element deletion were performed. Subsequently, they extracted the topmost profile and the bottom most profile of the finally remained connected components, respectively, i.e. the topmost pixels and the lowest pixels of vertical columns of the components. Finally, considering about 1200 images, they reported a successful classification rate of 95%.

In another work, Hangarge *et al.* [19] reported feature extraction from text blocks of size 128 x 128 images based on 13 global spatial features. Visual observation was an important tool for identifying several features from document images. From Devanagari, Roman and Urdu scripts they extracted features based on observations like 'matra' or 'shirorekha', which are present in Devanagari but not in Urdu and Roman. In the Roman script, presence of vertical strokes is more than horizontal strokes as compared to the other two scripts. Urdu scripts have a strong baseline as well as right diagonal strokes. It also has less number of holes compared to the other two scripts. Different stroke density based features like vertical stroke density, horizontal stroke density, etc. were also considered as features in their work. In morphological features, they computed horizontal openings, bottom hat, and top hat transformation for identifying three scripts. Finally, using KNN classifier, they reported a success rate of 99.2% for bi-script documents and 88.6% for tri-script documents.

Rajput *et al.* [20] proposed a scheme for script identification considering discrete cosine transform and wavelet based features. They have considered eight Indic scripts namely Roman, Devanagari, Kannada, Tamil, Bangla, Telugu, Punjabi, and Malayalam. Firstly, input images were manually broken into 512 x 512 size blocks. Then feature vectors were computed using DCT and DWT. They have considered Roman, Hindi and one regional language for tri-script classification purposes. Using KNN classifier they reported an average tri-script classification rate of 96.4%. This work can be improved by avoiding manual segmentation, which is a time consuming task.

Basu *et al.* [17] proposed a novel framework considering four scripts namely Latin, Devanagari, Bangla and Urdu for identifying the script of the numeric postal code from an address block of a multi-script postal document. Firstly, they localized the postal address block from the entire postal document with localized address block region using the Hough transform based method. The isolated handwritten digit pattern was then extracted. Then, the above four scripts were grouped into 25 clusters based on similar shaped digit pattern. A script independent unified pattern classifier was used to classify the numeric postal codes into one of these 25 clusters. Taking this classification result, a rule based script identification engine was designed to find the script of the numeric postal code. As feature extractor, the authors have used a quad-tree based image partitioning scheme from the numeric digit pattern. The reported average identification accuracy over a ten-fold cross validation of results for the SVM based 25 class unified pattern classifier was reported as 92.03%. For address block detection, Hough transform based method had been used here whose efficiency is dependent on the quality of the input image. So, if real life postal documents contain some arbitrary noise, then some denoising scheme needs to be applied over them for optimum results. Another issue in this work is that, the authors considered a defined address block region, which may not be a real life scenario for all types of postal documents.

LINE- LEVEL SCRIPT IDENTIFICATION

In line level script identification, a document image can contain more than one script, but it requires the same script on a single line. So, line segmentation is to be performed before computing the actual feature values. In order to explore the Indic scripts, topological, stroke based and structural features of the script are analyzed.

Moussa *et al.* [36] used fractal based feature for script identification from Arabic and Latin scripts. Their scheme worked for both handwritten and printed document images. In this scheme, firstly, they performed morphological transformation of line text images. Then they computed features based on fractal analysis from both of the original 2-D images and vertical, horizontal projection profiles. Finally, they obtained 12 features based on multi dimensional fractal analysis. They tested this proposed system

for 1000 prototypes with various typefaces, scripts styles and sizes. The accuracy rate was reported to be 96.64% using KNN classifier and 98.72% using RBF classifier. Lower computational cost as well as faster processing are the primary advantages of fractal based feature.

Rajput and Anita [37] proposed a scheme based on Gabor filter for identifying unknown script from a bi-script document. Eight Indic languages, namely Roman, Devanagari, Kannada, Tamil, Bangla, Telugu, Punjabi and Malayalam were considered in their scheme. Firstly, they have created a Gabor filter bank by considering six different orientations and three different frequencies to obtain 18 filters. They convolved the input image with the created Gabor filter Bank. For each output image they extracted cosine part and computed the standard deviation (18 features). They extracted the sine component and computed the standard deviation (18 features). Finally, they computed the standard deviation of the entire output image (18 features). This formed a feature vector of length 54. Finally, using KNN classifier they reported 100% success rate from bilingual scripts i.e. Roman mixed with any of the other regional languages.

WORD-LEVEL SCRIPT IDENTIFICATION

Word level script identification is a very common approach compared to the other approaches. It is basically a segmentation based approach. Firstly, lines are segmented, and then words are segmented before feature computation. Line and word segmentation from handwritten documents is itself a major research challenge. Unlike printed documents, handwritten documents do not follow standard intra-space gap between two consecutive lines and between two consecutive words. This is due to the different writing styles, which is one of the most common natures in handwritten document images. Sometimes, lines and words are not skewed properly, which makes the segmentation task more challenging. For line segmentation and word segmentation technique, there are some works [38] [39] [40] [41] [42] available in literature.

Roy *et al.* [21] proposed a method for script identification from postal documents considering Bangla/Devanagari and English languages. In the work towards Indian Postal Automation development [43], firstly, documents skews were detected and corrected. Then non-text parts were segmented from the document using run length smoothing algorithm (RLSA). Next, using a piecewise projection method the destination address block (DAB) was segmented into lines and then lines into words. During feature selection they have considered shirorekha based, water reservoir based, and presence of small components based features. The idea of water reservoir technique works based on the idea of storing some water in different shape reservoirs. Here, if water is poured from top and bottom of the characters/numeral, the cavity regions of the characters/numerals where water will be stored are considered as reservoirs. Here, by top (bottom) reservoirs, it is meant that the reservoirs are obtained when water is poured from top (bottom). (Here, water pouring from bottom means the water pouring from top after rotating the component by 180°). Though this feature is efficient in terms of performance, there is a huge computational cost associated with it. Finally, using Tree classifier, they reported a classification result of 89%. But it was also claimed that if small words are ignored, then success rate will increase to 93%.

In another work, Roy *et al.* [31] used fractal based, busy-zone based, water reservoir based, presence of small components and topological features to classify between Bangla and English language. Fractal and busy-zone based features perform well to distinguish 'matra' based scripts from their counterpart. Water reservoir features have already been discussed in the earlier paragraph. Here, MLP was used for classification and a success rate of 97.62% was acheived. Roy *et al.* [5] used almost the same set of features for identifying Bangla and Oriya languages and reported a successful classification rate of 97.69%. Roy *et al.* in [34] proposed a script identification scheme considering Roman and Persian scripts. In this work, they have considered a set of 12 features based on fractal dimension, position of small component and topology based features. Finally, using KNN classifier they reported a successful identification rate of 99.20%.

Benjelil *et al.* [33] reported a work considering Arabic and Roman script using Steerable Pyramid (SP) based feature. The SP is a linear multi-scale, multi-orientation image decomposition, that provides a useful front-end for image-processing and computer vision applications. The SP can capture the variation of a texture in both intensity and orientation. Initially, the image was separated into low and high pass sub bands, using low pass and high pass filters. The low pass sub bands was then divided into a set of oriented band pass sub bands and lower pass sub bands. This lower pass sub band was sub sampled by a factor of 2 in the X and Y directions. The recursive (pyramid) construction of a pyramid was achieved by inserting a copy of the shaded portion of the diagram at the location of the solid circle. The basic functions of the steerable pyramid were directional derivative operators that came in different sizes and orientations.

Sarkar *et al.* [22] proposed a word level script identification technique from Bangla and Devanagari handwritten texts mixed with Roman script. In this work, they first extracted the text lines and words from document pages using a script independent neighboring component analysis technique [44]. During the feature extraction stage, horizontalness, segmentation related and foreground-background transition related features were considered. Horizontalness property is directly related to 'matra'/'shirorekha' which is presented in Bangla and Devanagari but do not in Roman. The feature was extracted by calculating the row wise sum of continuous black pixels. In segmentation based feature they considered a number of 'matra' pixels and number of segmentation point pixels. In the foreground-background transition feature, they observed that the horizontal pixel density varies in different regions. Considering this, they computed the changeover of foreground and background pixels as a feature for classifying 'matra'-based scripts from their counterpart.

Singh *et al.* [25] reported a technique, which automatically identified the script of handwritten words from a document page, written in *Devanagari* script mixed with *Roman* script. 39 distinctive features (8 topological and 31 convex hull based features)

were extracted and classification was done using MLP classifier with 3-fold cross validation. An average accuracy rate of 99.54% was reported by the authors.

An application of automatic classification of content type in torn documents was proposed by Chanda *et al.* [23] based on script of the text. For classification they used rotation invariant Zernike moment based feature with SVM classifier. Along with that, gradient features were also computed for a comparative analysis between rotation dependent and independent aspects. Finally, they reported an average eleven script accuracy of 81.39% at the component level and 94.65% at the word level.

Dhandra and Hangarge [45] reported a word level script identification technique from three Indic languages, namely Kannada, Roman and Devanagari text words and numerals. They carried out their work in two phases. The first phase reported the script identification of text words using morphological filters and regional descriptors based features of these three major Indic languages. In the second phase Kannada and Roman handwritten numeral script identification was carried out. Stroke density and pixel density, aspect ratio, eccentricity and extent were used as pertinent features for their work. An overall result reported is: average 96.05% accuracy for word script identification and an average 99% accuracy for numeral script identification. In a recent work, Hangarge *et al.* [24] proposed a word level script identification technique considering Roman, Devanagari and four south Indian scripts namely Kannada, Telugu, Tamil and Malayalam. Their primary investigation was capturing of diagonal edge based shape information by applying 1D and 2D DCT, which they reported as directional DCT based features. Firstly, the input word image matrix was considered and normalized into a square matrix by padding zeros. Then 1-D and 2-D DCT were computed for each of the (N-2) upper and lower diagonals (assuming the matrix size is an N x N) and their standard deviations were computed to reduce the size of feature vectors. Conventional DCT values were also computed by dividing the whole word image into four zones and their standard deviation was performed. Altogether a feature vector of size ten comprising of six features from the directional DCT and four features from conventional DCT was constructed. A bi-script and tri-script

identification accuracy of 96.95% and 96.42% respectively were reported bu the authors.

Pardeshi *et al.* [26] reported a technique based on different image transform based method to identify 11 Indic scripts namely Roman, Devanagari, Urdu, Kannada, Oriya, Gujarati, Bangla, Gurumukhi, Tamil, Telugu and Malayalam. Radon Transform, DWT, Statistical filters, DCT methods were used for feature extraction and SVM, KNN were used as classifier. Experimentation was carried out on 28100 word images and an average bi-script and tri-script accuracy were reported to be 98% and 96% accordingly. This work has the limitation in terms of execution time compared to other local or script dependent features which runs very fast.

Singh *et al.* [28] proposed a word level script identification technique from seven Indic scripts namely Bangla, Devanagari, Gurumukhi, Malayalam, Oriya, Telugu and Roman that used elliptical and polygon approximation based techniques. Out of total 82 features, 12 features were obtained from maximum inscribed ellipse, where, the ellipse fitting was done on the word images and some local values were computed. Other 32 features were obtained in the similar way, where, the whole word images were divided into four regions and again ellipse fitting was done on each of the word segments. Now each segment rendered 8 features generating a total of 32 features. Other 14 features were obtained from four concentric elliptical regions. During polygon approximation, they applied distance threshold and fit-split methods to generate remaining 24 features. The authors had prepared a dataset of 7000 words which were extracted manually from handwritten pages. Using 5-fold cross validation, on an average accuracy of 95.35% had been reported. The following Table 1.3 summarizes the offline handwritten identification techniques.

Table 1.3 Summarization of the methods for Offline Script Identification from Handwritten or Handwritten-Printed mixed document Images of Indic scripts/languages

Work	Method		Language/ Script	Dataset Size	Avg. Acc. (%)
	Features	Classifier			
Page-level					
Hochberg *et al.* [32]	Connected component analysis	LDA, MLP	Arabic, Chinese, Cyrillic, Devanagari, Japanese, Latin	496	88.00
Zhu *et al.* [16]	Translation, scale based descriptor, shape codebook	SVM	Arabic, Chinese, Roman, Hindi, Japanese, Korean, Russian, Thai	1512	95.60
Singh *et al.* [27]	Modified log-gabor filter based feature	Simple logistic	Bangla, Devanagari, Gurumukh, Oriya, Tamil, Telugu, Urdu, Roman	240	95.35
Block-level					
Kanoun *et al.* [29]	Morphological analysis, geometrical analysis based features	KNN	Arabic (A), Latin (L)	735	A: 88.00 L: 98.00
Singhal *et al.* [18]	Multi-channel Gabor filter and GLCM	Multi-prototype	Devanagari (D), Bangla (B), Telugu (T), Latin (L)	480	D: 90.00 B: 86.60 T: 96.70 L: 93.30
Zhou *et al.* [30]	Connected component analysis	Rule based	Bangla, Roman	1200	95.00
Basu *et al.* [17]	Similar shaped digit pattern	SVM	Latin, Devanagari, Bangla, Urdu	100	92.03
Hangarge and Dhandra [19]	Stroke density, pixel density, morphological transformation	KNN	Roman, Devanagari, Urdu	300	Bi: 99.2 Tri: 88.6
Rajput and Anita [20]	DCT and Wavelet based features	KNN	Roman, Devanagari, Kannada, Tamil, Bangla, Telagu, Punjabi, Malayalam	800	Tri: 96.4
Line-level					

Moussa *et al.* [36]	Fractal based features	KNN, RBF	Arabic (A), Latin (L)	1000	KNN A: 93.30 L: 96.00 RBF A: 97.30 L: 98.60
Rajput and Anita [37]	Gabor filter	KNN	Roman, Devanagari, Kannada, Tamil, Bangla, Telagu, Punjabi, Malayalam	800	Bi: 100%
Word-level					
Roy *et al.* [21]	Component analysis	Tree based	Bangla/ Devanagari, Roman	2342	89.00
Roy *et al.* [31]	Component analysis and topological features	MLP	Bangla, Roman	4342	97.62
Roy and Pal [5]	Component, zone analysis and topological feature	MLP	Roman, Oriya	2500	97.69
Benjelil *et al.* [33]	Steerable pyramid	KNN	Arabic (A), Latin (L)	400	A: 97.00 L: 96.00
Roy *et al.* [34]	Fractal, component analysis and topological features	MLP	Persian, Roman	5000	99.20
Sarkar *et al.* [22]	Foreground-background transition	MLP	Bangla and Roman (B-R), Devanagari and Roman (D-R)	3200	B-R: 99.29 D-R: 98.43

Chanda *et al.* [23]	Zernike moment based feature (rotation invarient)	SVM	Roman, Devanagari, Urdu, Kannada, Oriya, Gujarati, Bangla, Gurumukh, Tamil, Telugu, Malayalam	240 page words	94.65
Hangarge *et al.* [24]	Directional DCT	KNN, LDA	Roman, Devanagari, Kannada, Telugu, Tamil, Malayalam	9000	Bi: 96.95 Tri: 96.4 Mul: 85.7
Singh *et al.* [25]	Topological and convex hull based	MLP	Devanagari, Roman	100 page words	99.54
Pardeshi *et al.* [26]	Radon transform, discrete wavelet transform, statistical filter, DCT	SVM, KNN	Roman, Devanagari, Urdu, Kannada, Oriya, Gujarati, Bangla, Gurumukh, Tamil, Telugu, Malayalam	28100	Bi: 98.00 Tri: 96.0
Singh *et al.* [28]	Elliptical and polygon approximation based feature	MLP	Bangla, Devanagari, Gurumukh, Oriya, Malayalam, Telugu, Roman	7000	95.35

Table 1.4 is a summarized version of Table 1.3. It is evident from Table 1.4 that number of works at: *word level* > *block level* > *page level* > *line level* > *character level*. So, script identification at page, line and character level need to be explored. With respect to the number of scripts considered till date, we have found that only word level work has been performed considering all official Indic scripts. In other level, no work considering all official Indic scripts has been reported so far. From the study we have noticed that among the classifiers have been chosen, MLP and KNN are preferable classifier irrespective of the level of work.

Table 1.4 Distribution of different works at different level

Document level	Reported works	Remarks
Page level	Hochberg *et al.* [32], Zhu *et al.* [16], Singh *et al.* [27]	Total number of page level works reported here is three. In most of the cases (>66%) MLP was used as a classifier. Highest number of scripts considered at page level is eight (Bangla, Devanagari, Gurumukhi, Oriya, Tamil, Telugu, Urdu, Roman) by Singh *et al.* [27].
Block level	Kanoun *et al.* [29], Singhal *et al.* [18], Zhou *et al.* [30], Basu *et al.* [17], Hangarge and Dhandra [19], Rajput and Anita [20]	Number of block level works reported is six. KNN is the mostly used classifier at this level of work. Rajput and Anita [20] considered highest number of scripts at block level (Roman, Devanagari, Kannada, Tamil, Bangla, Telagu, Punjabi, Malayalam)
Line level	Moussa *et al.* [36], Rajput and Anita [37]	Total two works are reported at line level. Similar to block level, KNN is the mostly used classifier at line level. Rajput and Anita [37] considered highest number of scripts at line level (Roman, Devanagari, Kannada, Tamil, Bangla, Telagu, Punjabi, Malayalam).
Word level	Roy *et al.* [21], Roy *et al.* [31], Roy and Pal [5], Benjelil *et al.* [33], Roy *et al.* [34], Chanda *et al.* [23], Sarkar *et al.* [22], Hangarge *et al.* [24], Singh *et al.* [25], Pardeshi *et al.* [26], Singh *et al.* [28]	At word level highest number of works is carried out (total eleven). Chanda *et al.* [23] and Pardeshi *et al.* [26] considered all the eleven Indic scripts (Roman, Devanagari, Urdu, Kannada, Oriya, Gujarati, Bangla, Gurumukhi, Tamil, Telugu, Malayalam) in their work. MLP and KNN are the commonly used classifier for word level work.

NON-INDIC SCRIPTS:

Among the works on non-Indic scripts, Spitz [46] developed a method to separate Han or Latin based scripts. Optical density distribution of characters and frequently occurring words shape characteristics had been used for this purpose. Using cluster based templates, an automatic script identification technique had been described by Hochberg *et al.* [47]. Ding *et al.* [48] proposed a method for separating the two classes

of scripts: European (comprising Roman and Cyrillic scripts) and Oriental (comprising Chinese, Japanese and Korean scripts). Using fractal based texture features, Tan [49] described an automatic method for identification of Chinese, Roman, Greek, Russian, Malayalam and Persian printed text. Chanda *et al.* [50] proposed a system for Roman and Thai script identification using the SVM classifier. All the above pieces of work deal with non-Indic scripts and are based solely on offline documents. Lee and Kim [51] proposed a scheme for online multi-lingual cursive handwritten language identification, based on hidden markov model (HMM). They have considered Hangul and Roman handwritten text documents as individually or in combination.

1.4 CHALLENGES

Several important issues are to be considered while designing multi-script handwritten OCR system for a multi-script country like India. It can be found that most of the work has been done using three popular scripts namely Devanagari, Bangla and Roman. To analyze the reason behind this, we can observe Table 1.1 where, approximately 328.23, 211.50 and 334.20 million Indian people are reported to use Devanagari, Bangla and Roman scripts respectively. Considerable numbers of works have been found on Urdu and South Indian scripts. Surprisingly, reported works on Gujarati script is very less, as yet, although 46.50 million people are using this script. Another notable problem is the unavailability of the handwritten database for all Indic scripts. If more dataset are available, the system can be effectively tested to produce robust and reliable results. Another issue in handwritten script identification is working on the line, word or character level due to the segmentation challenge. Line, word or character segmentation, from handwritten document is itself a challenging research area. Researchers are trying hard to develop algorithms and techniques for word, line and character segmentation with optimum accuracy. This problem arises due to different unavoidable factors like: variations of writing style for different people, the presence of skew on line or word level and some time at character level also, uneven spacing between words or line, etc. while considering handwritten documents. In a nutshell, we summarize the following key challanges related to Indic script identification problem:

- **BENCHMARK DATABASE FOR ALL OFFICIAL INDIC SCRIPTS**

The first and foremost requirement is availability of standard dataset. To the best of our knowledge, till date, no handwritten dataset has been developed for all official Indic scripts. The task is really challenging because of the geographical distribution of Indian populations from north to south (Kashmir to Kanyakumari) and east to west (Tripura to Gujarat). To cater to the versatility of the database, more people with diversified age, education, culture etc. have to be involved. So attention from the OCR research community is expected to resolve the matter as early as possible.

- **SCRIPT IDENTIFICATION FROM ALL OFFICIAL INDIC SCRIPTS**

The work of script identification till date was mostly on scripts like Devanagari, Bangla and Roman. It is also to be noted that, till date no work has been reported considering all official Indic scripts at page, block and line level. Even though few works have been reported at word level, their reported accurecies are significantly low.

- **MULTI-LEVEL SCRIPT IDENTIFICATION**

So far different authors have reported different works at page, block, line, word and character level. But if a single document is considered at different levels, then how this segmentation affects the script identification performance is yet to be studied. In general, in terms of information contents page-level documents contains more information compared to block, line, word and character level.

- **NUMERAL SCRIPT IDENTIFICATION**

Numeric script identification also helps in automatic sorting of postal documents in Indian multi-script scenario. So, this problem (numeral script identification) also needs attention.

- **OPTIMIZATION ISSUES RELATED TO SCRIPT IDENTIFICATION PROBLEM**

Performance of script identification solely depends on the particular feature chosen. Feature selection is an important issue while studying which feature/features are more suitable for script identification. This issue needs to be addressed.

1.5 RESEARCH MOTIVATION

The main motivation of this thesis can be pointed out as follows:

- India is a multi-lingual, multi-script country (23 languages, 11 scripts including English and Roman)
- An official document may be written by any of these languages creating multi-script documents
- Multi-script documents may be categorized into:
  - -Multiple documents written with multiple scripts
  - -Single document written with multiple script
- Script identification is a prerequisite for choosing a particular OCR from an OCR bank for a target language/script.
- We are also motivated to develop benchmark dataset for all official Indic scripts.

1.6 OBJECTIVE

The objective of this thesis can be summarized as follows:

- Preparing document image dataset for official Indic scripts and reporting benchmark results for script identification at different levels i.e. at page, block, line or word level.
- Designing of script identification techniques for different Indic scripts and evaluating their performance using different classifiers.
- Addressing the challenges associated with handwritten script identification techniques.
- Addressing the issue of multi-level script identification, i.e. script identification from the same documents which are considered at page, block, line and word-level.

1.7 CONTRIBUTION

The above list of challenges motivated us to consider Indic script identification as a research problem of this thesis and to propose novel feature/features, dataset, benchmark results. We have also tried to improve the performance of script identification to achieve better accuracy and used low dimensional features which are fast to compute. The principal contributions of the thesis are as follows:

- Properties of different Indic scripts, their origin, and demographic distribution are studied. Different works on printed and handwritten script identification are studied, their applicability, limitations are pointed out.
- Without publicly available datasets, specifically in handwritten document recognition (HDR), we cannot make a fair and/or reliable comparison between the methods. Considering HDR, Indic script's document identification is still in its early stage compared to others such as Roman and Arabic. In this work we proposed benchmark Indic script dataset on printed and handwritten documents for all the eleven official Indic scripts. Not only that, we also proposed handwritten numeral image dataset from four popular Indic scripts.
- We have proposed some novel features and studied their effectiveness for Indic script identification. We have achieved promising script identification accuracy especially in handwritten scenario.
- We also have tried to keep the feature dimension as low as possible so that training time get reduced while building the model.
- The issue of numeral Indic script identification is addressed in this thesis. Numeral script identification helps in different application like: automatic sorting of postal documents, document catagorization based on handwritten roll numbers in different scripts.
- We have studied the effect of segmentation at page, block, line and word level on the performance of script identification.

- Performance of different feature combinations has also been studied for the current problem.

1.8 ORGANIZATION OF THE THESIS

Script identification from printed and handwritten document written using different official Indic scripts is proposed in this thesis. The rest of the chapters have been organized as follows:

Chapter 2 discusses dataset development. Dataset is the most crucial part of any pattern recognition tasks. Without publicly available dataset we cannot make fair comparison of our techniques. In this chapter, we describe the data collection methodology, convention, preparation of printed and handwritten dataset.

Development of different methods and techniques for the Indic script identification problem has been discussed in Chapter 3. Script identification techniques are broadly classified into two types: (i) Script dependent techniques and (ii) Script independent techniques. Fusion of different script independent features is discussed. The use of different classifiers and experimental strategies are also discussed in this chapter.

Script identification systems are broadly classified into two types based on the nature of the input documents: (i) Printed script identification and (ii) Handwritten script identification. In Chapter 4, we studied the performance of printed script identification, especially at page and word level for all official scripts.

In Chapter 5, we describe the handwritten script identification scheme. Handwritten script identification is more challenging compared to printed one due to several reasons: versatility of writing style, variation in inter-line, inter-word spacing, character sizes for different users across the globe. Script identification approaches which will be suitable for printed documents may sometime generate upsetting results for

handwritten cases. That is why script identification from handwritten document images is still an open challenge.

Chapter 6 concludes the thesis discussing the overall summary and scope of the future work. Besides listing the conclusion of the present work, this chapter discusses about future direction of research. Not only that, we also mention few of the limitations of the present work.

All the references are listed at end of the thesis.

CHAPTER TWO

DEVELOPMENT OF DATASETS

The progress of Indic script identification is still in an early stage because of inadequacy of benchmark datasets. When we started this work, we didn't find any publicly available dataset that covers the entire domain of Indic scripts, i.e. all the eleven official Indic scripts. Printed dataset can be prepared from several readily available sources. But preparing handwritten dataset is a real challenge. The main reason is the huge demographic distribution of Indian population and the spread over of different languages across different regions. One has to travel extensively across different regions of India to collect handwritten samples. To fill this gap we propose some benchmark dataset of official Indic scripts. This chapter focuses on handwritten Indic dataset development issues like: review on the existing dataset, motivation, challanges and dataset preparation.

2.1 CONTEXT

Without publicly available datasets, specifically in handwritten document recognition (HDR), we cannot make a fair and/or reliable comparison between the methods. Considering HDR, Indic script's document identification is still in its early stages compared to others, such as Roman and Arabic. In document image analysis (DIA), HDR can be considered as one of the challenging areas and it includes applications such as segmentation, script identification and writer verification. Researchers have found that, the script identification (from multi-script documents) has made a real impact in a country like India, where as per the 8th schedule of the constitution, 22 official languages (excluding English, which is also very popular in India) [3] are used for verbal communication and 11 scripts are used to write those languages. This, in

general puts a burden on optical character recognizer (OCR), since OCR is script specific or script dependent. Therefore, to explore the possibility of recognizing a script of a page without any prior knowledge, one of the solution is to develop a script identification system so that one can use it as a precursor to the script specific OCR. In this chapter, we present three datasets: (i) *PHDIndic_11* [52], that is composed of 11 official Indic scripts (having a fairly large amount of text pages, text lines, words/sub-words of all scripts) to be used for an automatic script identification from multi-script documents and (ii) Word-level printed dataset of 11 official Indic scripts (from 13 languages) [53] and (iii) *Numeral_db* [54], a handwritten numeral dataset from four popular Indic scripts: Bangla, Devanagari, Roman and Urdu.

2.2 RELATED WORK

An overview of the available datasets developed till date by different researchers emphasizing Indic scripts is shown in Table 2.1. There are several popular Roman/Latin script datasets. Roman and Latin are normally used interchangeably. 'Latin alphabet' is generally used to portray the alphabet used to write Latin in classical times, even as 'Roman alphabet' is usually used to depict the adaptation of the Latin alphabet to write languages like: English and French. Throughout this chapter we will use Roman as the official script name. *NIST* [55] includes 810000 characters and digits, 91500 text and phrases of running English text. *CENPRAMI* [56] contains 17000 digits extracted from images of 3400 postal zip codes. *CEDAR* [57] contains 14000 city and state names, 5000 postal zip codes and 49000 isolated characters and digits. *MNIST* [58] contains 70000 Roman digits. *IAM-database* is a popular Roman script dataset developed by Marti and Bunke [59] [60], which contains 1539 pages, 5685 sentences, 13353 lines and 115320 words. These datasets can be used for various application related to offline handwritten text identification from document images. An automatic word segmentation scheme was developed by Zimmermann and Bunke [61], to extract those words. All images are provided in .png file format and all the pre-processing information is also provided in .xml file format [62]. *ICDAR 2009* handwritten

segmentation contest dataset [63] contains page-level handwritten document images of about 300 pages.

Bhattacharya and Chaudhuri [64], in the year 2005 reported handwritten isolated numeral dataset of Devanagari, Bangla and Oriya scripts. The dataset consists of 22556 Devanagari numerals written by 1049 people, 23392 Bangla numerals written by 1106 people and 5970 Oriya numerals written by 356 people. A large Bangla numeral dataset was reported by Chaudhuri [65] in the year 2006. It contains about 8348 online numeral strings and 23392 offline isolated numerals. Many postal documents are considered for this dataset generation.

CENPARMI-U, a fairly large Urdu dataset was developed by Saqheer *et al.* [66] in the year 2009, which includes isolated digits, numeral strings with/without decimal points, five special symbols, 44 isolated characters and 57 Urdu words.

CMATERdb1 [4], was developed by Sarkar *et al.* in the Centre for Microprocessor Applications for Training Education and Research, Jadavpur University, Kolkata, in the year 2012, which contains 150 page-level document images. Out of 150 handwritten document pages, 100 pages are written purely in Bangla script and rest of the 50 pages are written in Bangla text mixed with English words. Their ground truth labeling is done by using Bangla script with blue color and Roman script by red color. All the image files are saved in .bmp file format.

In [67], Nethravathi *et al.* reported a dataset in the year 2010, as a part of Tamil and Kannada handwriting identification work. It is a versatile dataset having about 100000 words from 600 different subjects.

KHTD [68], a Kannada script dataset consisting of 204 documents, 4298 lines, and 26115 words distributed over document, line and word level was developed by Aleai *et al.* in the year 2011. About 51 writers with varying age group contributed to build the KHTD.

UHSD, an offline sentence dataset of Urdu handwritten documents along with pre-processing and segmentation techniques was reported by Raza *et al.* [69]. Around 200 native writers contributed to build this dataset.

QUWI or Qatar University Writer Identification dataset [70] is an Arabic and Roman sentence level handwritten dataset built by Raza *et al.* in the year 2013. It consists of 4068 handwritten documents contributed by 1017 volunteers of different ages, nationalities, genders and education levels.

In the year 2012, a character level Devanagari script dataset, both for alphabets and numerals was developed by Dongre and Mankar [71]. Almost 750 writers contributed to build this dataset.

PBOK [72], a page level dataset of four different scripts: Persian, Bangla, Oriya and Kannada were developed by Alaei *et al.* The dataset contains total 707 text pages, 12565 text lines, 104541 words and 423980 characters. A total of 436 individuals have contributed in developing the dataset. Two types of ground truths, based on pixel information and content information, were generated for the PBOK dataset.

The CVL dataset [73], a public Roman script dataset for writer retrieval, writer identification and word spotting, was developed by Diem *et al.* in the year 2013. It consists of 2163 handwritten forms contributed by 311 different writers from English and German languages. Both the languages follow Roman script with minor variation. Document images are stored at 300 dpi RGB image format.

Tamil-DB [74], developed by Thadchanamoorthy *et al.* in the year 2013, is a popular and very useful handwritten city name dataset written in Tamil script. Almost 500 writers contributed to build this dataset. It was primarily developed for postal automation system.

Das *et al.* [75], in the year 2014, reported a benchmark image dataset of isolated Bangla handwritten compound characters. Altogether, 55278 isolated character images, belonging to 199 different pattern shapes are included in this dataset. The authors

reported benchmark identification accuracy of 79.35% on the test database consisting of 171 character classes. In a recent work [76], the same author has reported a Bangla character dataset which consists of 59892 characters (including compound characters).

Table 2.1 Handwritten script datasets (mainly Indic) reported till date

Dataset & year	Scripts	Level	Volume
1. NIST [55], 1992	Roman	Character, Digit, Phrase	810000 characters and digits, 91500 text and phrases
2. CENPREMI [56], 1992	Roman	Digit	17000 isolated digits
3. CEDAR [57], 1994	Roman	Word, Character, Digit	14000 city and state names, 5000 zip code, 49000 isolated characters
4. MNIST [58], 1998	Roman	Digit	70000 characters
5. IAM-database [59] [60], 2002	Roman	Page, Line, Sentence, Word	1539 pages, 5685 sentences, 13353 lines, 115320 words
6. ISICal numeral dataset [64], 2005	Bangla, Devanagari, Oriya	Character	22556 Devanagari, 23392 Bangla and 5970 Oriya numerals
7. Bangla-Numeral-DB [65], 2006	Bangla	String, Character	8348 online numerals and 23392 offline isolated numerals
8. ICDAR [63], 2009	Roman	Page	300 text pages
9. CENPARMI-U [66], 2009	Urdu	Word, Character, Digit	18000 words
10. CMATERdb1 [4], 2010	Bangla and Roman	Page	150 pages
11. Tamil-Kannada-DB [67], 2010	Tamil, Kannada	Word	About 100000 words
12. KHTD [68], 2011	Kannada	Page, Line, Word	204 documents, 4298 lines, 26115 words
13. UHSD [69], 2012	Urdu	Sentence	400 forms
14. QUWI [70], 2012	Roman, Arabic	Sentence	4068 forms
15. Devanagari-DB [71], 2012	Devanagari	Character, Digit	20305 characters, 5137 digits
16. PBOK [72], 2012	Persian, Bangla, Oriya, Kannada	Page	707 text pages

17. CVL [73], 2013	Roman	Sentence	2163 forms
18. Tamil-DB [74], 2013	Tamil	Word	26500 city names
19. Compound characters [75], 2014	Bangla	Character	55278 isolated compound characters

2.3 OUR CONTRIBUTION

It is evident from Table 2.1 that so far the dataset development efforts were focused on scripts like Roman and few other Indic scripts, mainly at character and digit level. Few page-level dataset have been reported, but they were unable to cover the whole domain of Indic scripts. Development of a page-level dataset covering a fairly large number of Indic scripts is still lacking. To bridge this gap we propose *PHDIndic_11,* a new handwritten dataset [52], having document images from all official Indic scripts (with fairly large number of pages from each script). It has also been noticed that, dataset development efforts are mainly for alphabetic texts. Few numeral datasets are available but they are restricted to digit level. There exist several applications for multi-script numeral identification, such as: automatic sorting of postal documents based on PIN codes, sorting of answer scripts based on student roll number, arranging application form based on numeric application id. In all these applications, script identification from handwritten numeral string is the key idea. To bridge the gap of unavailability of numeral string dataset, we have proposed *Numeral_db* [54]. To make it more clear, our key contribution can be highlighted as follows:

- We proposed *PHDIndic_11,* a new dataset which contains 1458 handwritten page-level images from 11 official scripts of India, namely, Bangla, Devanagari, Roman, Urdu, Oriya, Gurumukhi, Gujarati, Tamil, Telugu, Malayalam and Kannada. Except few, many of these scripts are also used in outside of India too [77] (as shown in Table 2.2).
- We proposed a printed word-level dataset of 39K words from 11 different scripts (from 13 different languages).

- *Numeral_db*, a handwritten numeral dataset from four popular Indic scripts is also proposed.
- We proposed benchmark results on these datasets for handwritten and printed script identification. Printed script identification and handwritten script identification results have been discussed in Chapter 4 and 5 respectively.

Table 2.2 Global demographic distribution of different official Indic scripts [77]

Script	**Countries/regions outside India**	**Population (M)**
Bangla	Bangladesh, Nepal, Singapore	156.70
Devanagari	Nepal, Singapore, South Africa, Bhutan	32.14
Telugu	Singapore	0.006
Tamil	Malaysia, Mauritius, Singapore, South Africa, Sri Lanka	7.90
Urdu/ Perso-Arabic	United Kingdom, Saudi Arabia, United States, Pakistan, Middle East Asia, Bangladesh, Mauritius, Nepal, South Africa	15.77
Gujarati	Bahrain, Kenya, Pakistan, Singapore, Tanzania, Zambia	0.36
Malayalam	Singapore	0.03
Gurumukhi	Kenya, Singapore	0.02
Roman	Throughout the world	256.90

2.4 OVERVIEW ON PROPOSED DATASET

2.4.1 PHDINDIC_11: A PAGE-LEVEL HANDWRITTEN DATASET

PHDIndic_11 is a collection of text pages of 11 official scripts of India. These 11 scripts are used by all the official languages of India which are included in the 8th schedule of the Indian constitution till date. The naming convention of the dataset is as follows: 'P' stands for page-level, 'H' stands for handwritten, 'D' stands for dataset, 'Indic' signifies Indian subcontinent and '11' is the number of scripts covered in the present dataset. The *PHDIndic_11* contains a fairly large amount of text pages with enormous diversity in terms of the number of scripts/languages, number of writers from different

geographical locations, shape and size of the characters, content type and different writing directions (i.e. from left to right or right to left). Overall, this dataset has a volume of 1458 handwritten text pages and 463 individuals have contributed to build them.

CHALLENGES

During collection of *PHDIndic_11,* we have faced the following key challenges:

- *Standardization:* To standardize our data, we needed to understand the collection mechanism and protocols. For this purpose, initially we studied many standard public datasets and their collection procedures and preparation. These are already listed in Table 2.1.
- *Time:* Data collection is a tedious and time consuming task. It took more than two and half years to collect the entire *PHDIndic_11.*
- *Demography:* India is a large country with 1.3 billion people living in 36 different states/union territories. To collect different scripts data we had to extensively travel throughout the country.
- *Writer psychology:* It was not easy to collect data from different writers especially from unknown ones. To incorporate variability and realness among the data we considered people of different age, sex, educational qualification. We had to approach many unknown people and need to explain the necessity of this data collection project.

DATA COLLECTION METHODOLOGY AND CONVENTIONS

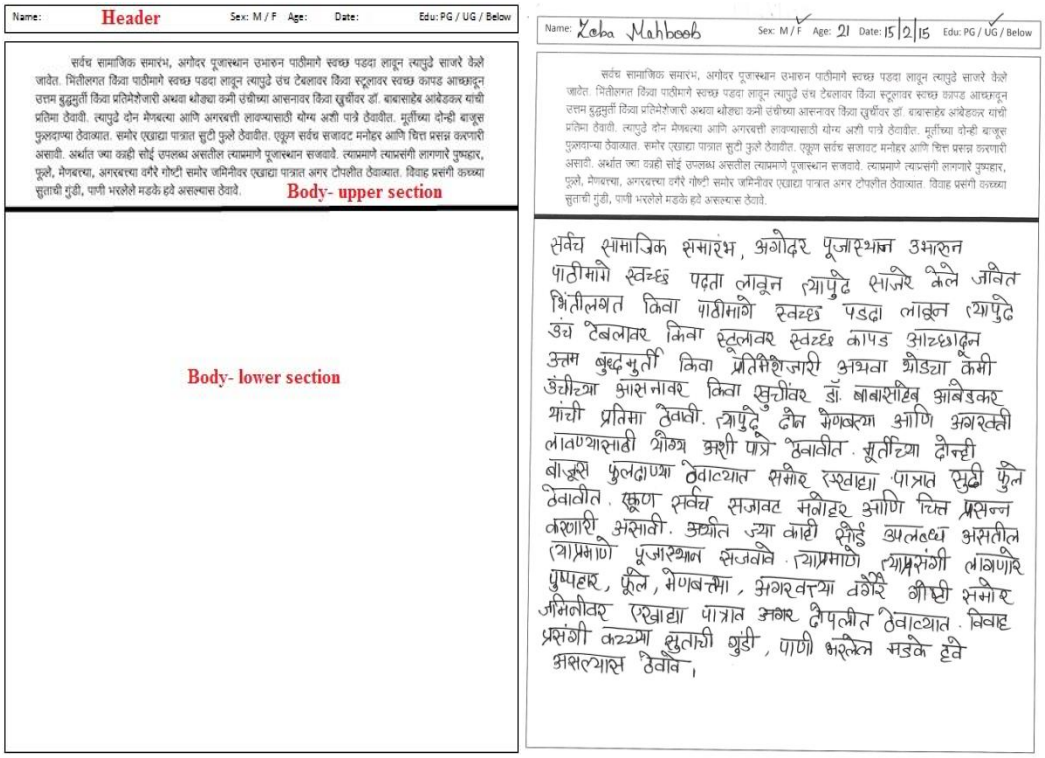
Data collection is one of the most time consuming and tedious task in any pattern recognition work. It becomes more challenging when the demographic variations of data to be collected are very much wide. Printed data are available from different easy sources like: newspaper, books, magazine articles etc. but collection of handwritten data from different writers of different places across the country is a real challenge. Handwritten texts can be written either in structured document (pre-formatted forms

with predefined text provided for writting) or in an unconstrained fashion. In our case both the modalities were adopted, i.e. in one type of sheets the volunteers were asked to write the text given in the specified area and the second type of forms were totally unconstrained, i.e. the writers were asked to write anything they want in their native script. Total six such forms were given to each writer and out of these six, five were pre-formatted (with predefind texts) and one was completely unconstrained (writer can write any text as per their choice). In the following paragraph we discuss about the data collection form preparation and conventions.

The first stage was preparing a standard form for collecting the dataset, which was prepared in our lab as shown in Figure 2.1. The form contains header and body, but no footer. Header field contains the name of the writer, sex, age and educational qualification at the top most position. For simplicity, we have provided the customized information like 'M' or 'F' in the sex tag, 'PG' or 'UG' or 'Below' in the education tag so that the writer can simply mark the appropriate choice. The body was divided into two sections – upper and lower. In the upper section, machine printed texts were provided. These texts were selected by consulting linguistics so that we do not miss characters (including compound characters) for all scripts. During text collection we have considered different news, novels, stories, state board and university syllabus contents, etc. to incorporate the maximum variability within the texts. The lower section of the body was left blank, where the writer has to write the given content in his/her own handwriting. They were asked to write the given content in the blank area of the form without any constraint (i.e. no restrictions were imposed regarding the type of pen used, ink color, or style of the writing). We paid special attention to collect data from people with different age and education qualification. Moreover we collected data from different places like office, home, college, school etc. to ensure maximum variability of writing. It has also ensured that most of the scripts were written by native writers (> 95% cases) except for few of the exceptions.

After collecting the forms, these handwritten text pages were scanned using HP flatbed scanner M1136 MFP at 300 dpi and were stored at 256 gray scale. Then the text

contents were extracted using an automatic text extraction technique. The final images were stored in a gray level format so that user can use them as per their need. Each image file has been given a name as: <document level>_<Script>_<4 digit serial number>. For example**,** a sample Bangla file is named as 'p_ben_0001', where 'p' stands for page-level, 'ben' stands for Bangla script and '0001' stands for image serial number and these three fields are separated by a '_' sign. As 'tif' file format is chosen, the first Bangla image file is stored as 'p_ben_0001.tif'.



(a) (b)

Figure 2.1 (a) Sample data collection form prepared in our lab for Devanagari script. The header and two body sub-sections are shown in red color (b) Filled up version of the same form as shown in (a)

PREPROCESSING

Preprocessing include text extraction from the scanned pages, which contains both predefined printed and handwritten texts. The handwritten texts were extracted using

an automated technique. Further these texts are converted into binary form applying the following thresholding technique.

- **THRESHOLDING**

PHDIndic_11 is available publicly in gray scale format. But for script identification purpose the data has to be converted into binary format. Initially the images are in gray tone and digitized at 300 dpi using a flat bed HP scanner. After digitization, pre-processing was carried out. A two stage based approach is used to convert the images into binary (0 and 1) or two tone images [43]. At first stage, pre-binarization is done using a local window based algorithm, in order to get an idea of different Region Of Interest or ROI. Then Run Length Smoothing Approach (RLSA) [5] is applied on the pre-binarized image. This will overcome the limitations of the local binarization method used. The stray/hollow regions created due to fixed window size are converted into a single component. Finally, using component labeling, each component is selected and mapped them in the original gray image to get respective zones of the original image. The final binary image is obtained by applying a histogram based global binarization algorithm on these regions/components of the original image.

EASTERN INDIC SCRIPTS: BANGLA, DEVANAGARI, URDU AND ORIYA

Bangla, Devanagari, Urdu and Oriya are the most popular eastern Indian scripts. For Bangla text pages, we have selected six different types of text contents. These texts contain both the Bangla basic characters and compound characters. These texts were given to 42 individuals with varying age, sex and educational background. Finally, we were able to collect total 161 handwritten text pages, which befitted the Bangla part of *PHDIndic_11*. The Bangla part of the dataset contains a total of 1820 text lines and 12447 words/subwords. On an average, each Bangla text page contains 11.30 text lines and 77.31 word/subwords. Two sample Bangla handwritten text images are shown in the following Figure 2.2.

For Devanagari script, initially we have prepared five different types of text from different areas. These texts were then given to 60 individuals of varying age, sex and

educational background. The collected texts were converted into gray scale using the same technique as we did for Bangla script. Altogether, 220 handwritten Devanagari text pages which include 2457 text lines and 23264 words/sub-words were gathered. On an average, each Devanagari text page contains 11.16 text lines and 105.74 words/sub-words. The first Devanagari text page was saved as 'p_dev_0001.tif'. Two sample Devanagari handwritten text images are shown in the Figure 2.3.

Urdu part of *PHDIndic_11* dataset contains 201 handwritten text pages written in Urdu and Oriya part of *PHDIndic_11* contains 172 text pages written in Oriya script. Total number of text lines is 1595 and 1422 for Urdu and Oriya respectively. So, on an average each Urdu text page contains 7.93 text lines and 91.65 words/sub-words. Whereas, each Oriya text page contains 8.26 text lines and 62.05 words/sub-words. The first sample of both Urdu and Oriya part of *PHDIndic_11* is named as 'p_urd_0001.tif' and 'p_ory_0001.tif' respectively. Few sample images of Urdu and Oriya handwritten text pages are shown in Figure 2.4 and Figure 2.5 respectively.

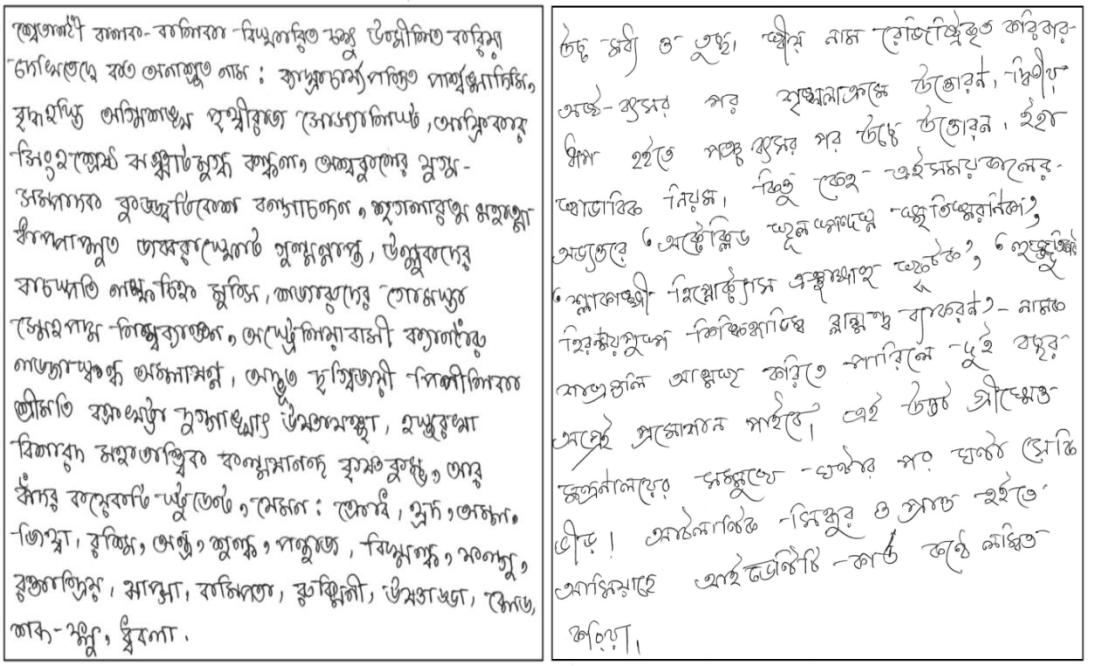


Figure 2.2 Two sample gray level scanned images of handwritten Bangla text

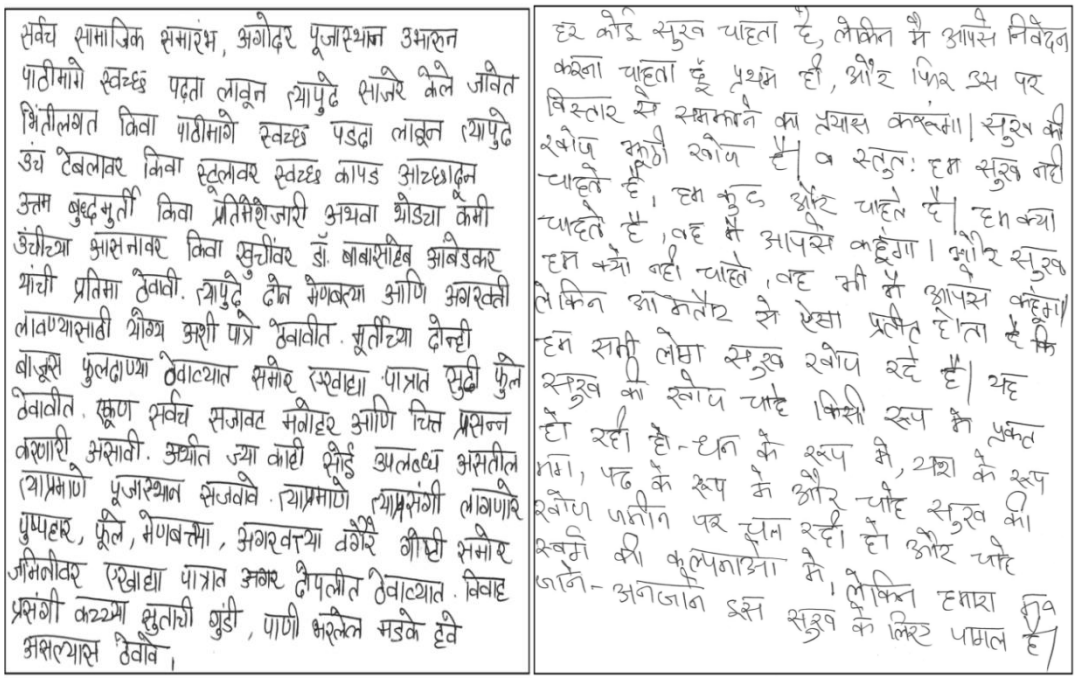


Figure 2.3 Two sample gray level scanned images of handwritten Devanagari text

Figure 2.4 Two sample gray level scanned images of handwritten Urdu text

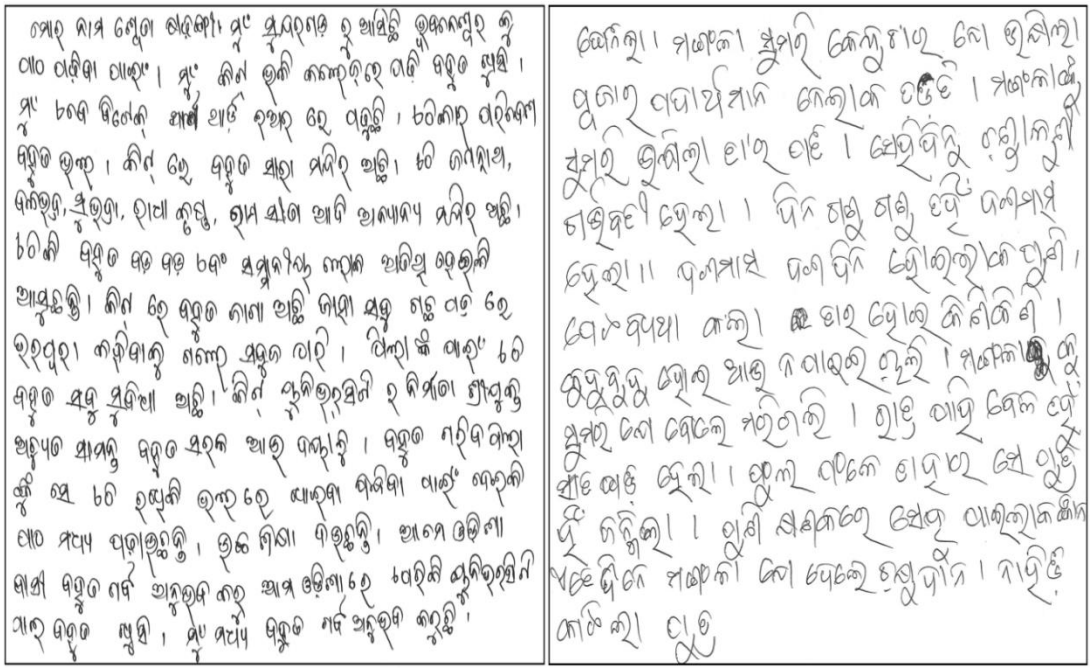


Figure 2.5 Two sample gray level scanned images of handwritten Oriya text

SOUTHERN INDIC SCRIPTS: TAMIL, TELUGU, MALAYALAM AND KANNADA

Now, we will discuss about the Southern Indian part of *PHDIndic_11* dataset, which has four scripts namely Tamil, Telugu, Malayalam and Kannada. We have collected a total of 358 handwritten text pages for the South Indian part of *PHDIndic_11,* which includes 120 handwritten text pages written in Tamil, 85 handwritten text-pages written in Telugu, 107 handwritten text pages written in Malayalam and 46 handwritten text pages written in Kannada. For Tamil part, number of text lines are 991 with an average of 8.25 lines and 46.11 words/sub-words per text page.

For Telugu part, number of text lines are 826 with an average of 9.71 lines and 53.94 words/sub-words per text page. For Malayalam part, number of text lines are 1028 with an average of 9.60 lines and 55.11 words/sub-words per text page. For Kannada part, number of text lines are 307 with an average of 6.67 lines and 33.95 words/sub-words per text page. As we named earlier, we named the first text page image as 'p_tam_0001.tif', 'p_tel_0001.tif', 'p_mal_0001.tif', 'p_kan_0001.tif' for Tamil, Telugu, Malayalam and Kannada respectively. The sample images of Tamil, Telugu, Malayalam

and Kannada handwritten text pages are shown in Figure 2.6, Figure 2.7, Figure 2.8 and Figure 2.9 respectively.

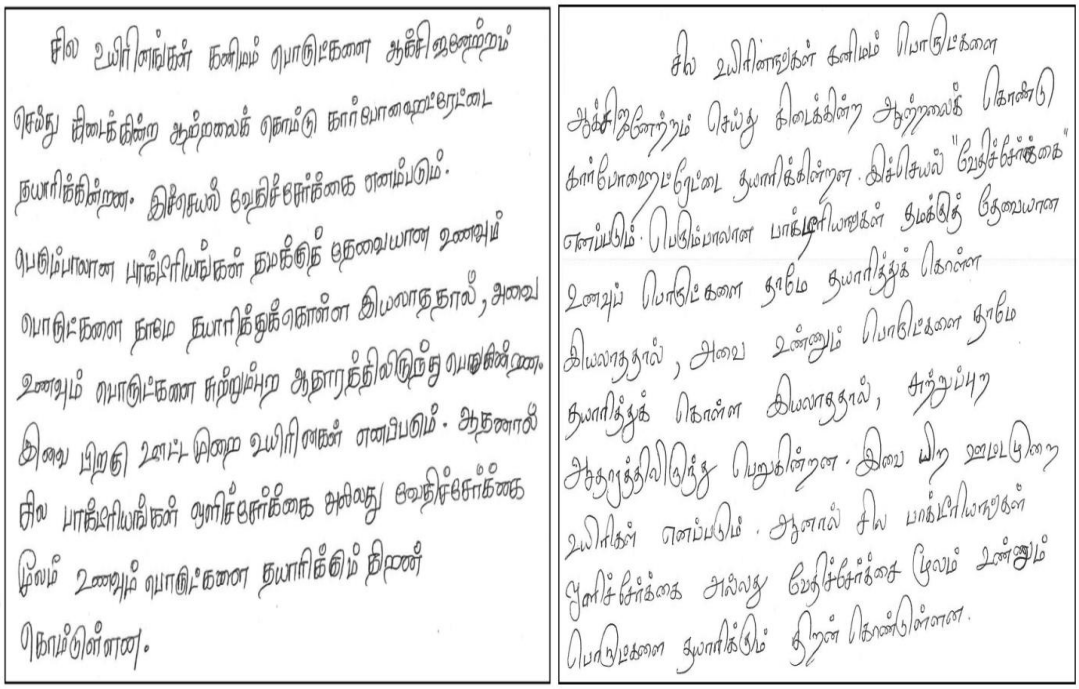


Figure 2.6 Two sample gray level scanned images of handwritten Tamil text

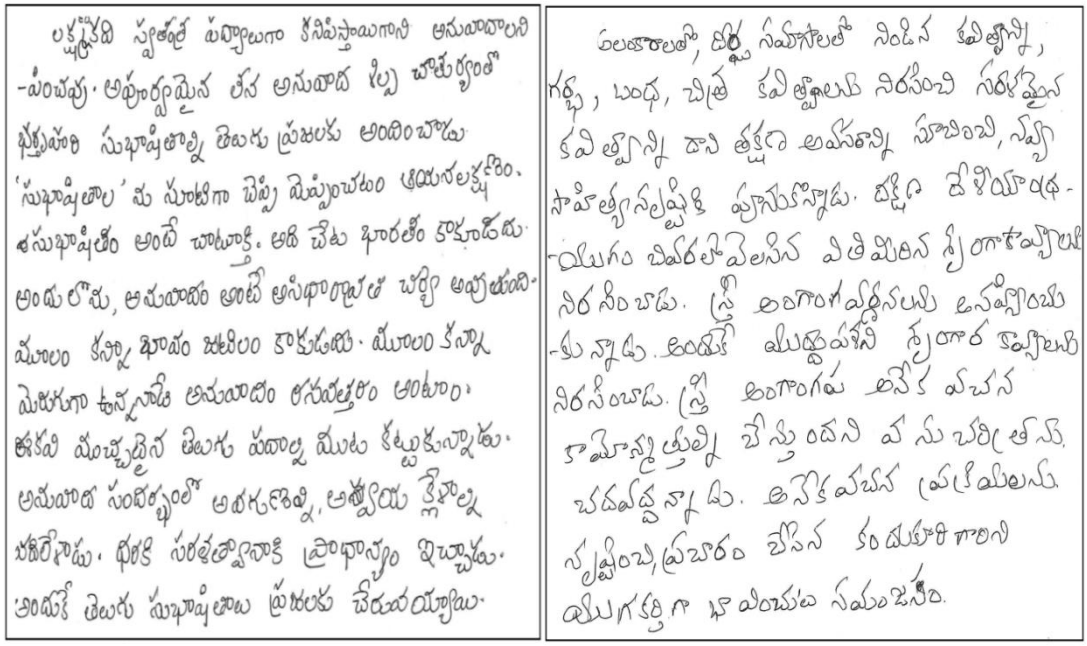


Figure 2.7 Two sample gray level scanned images of handwritten Telugu text

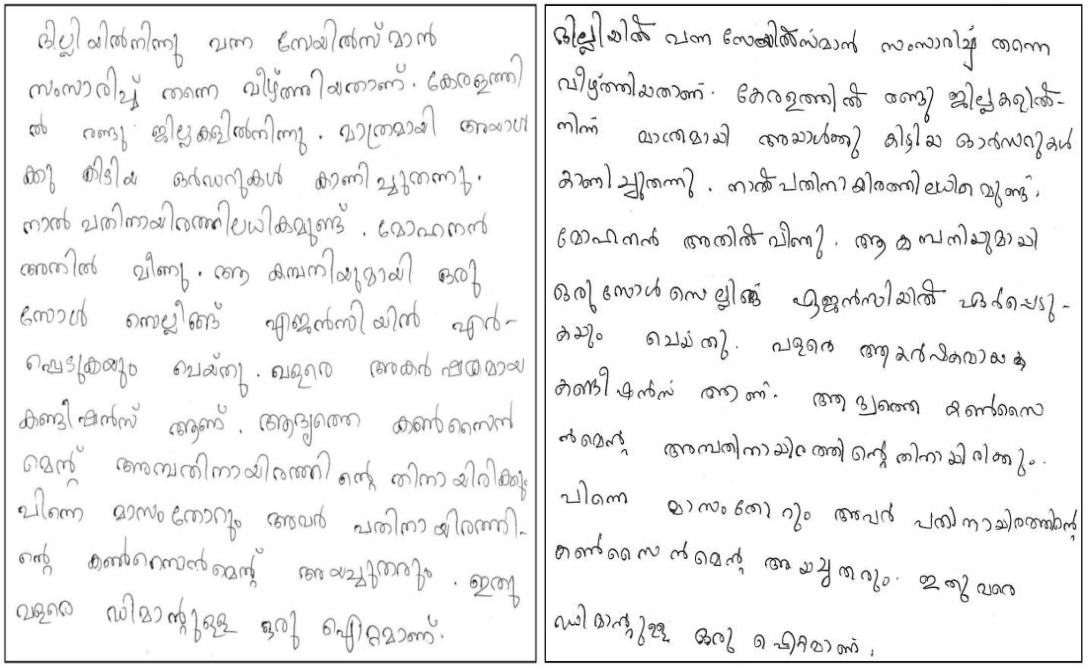


Figure 2.8 Two sample gray level scanned images of handwritten Malayalam text

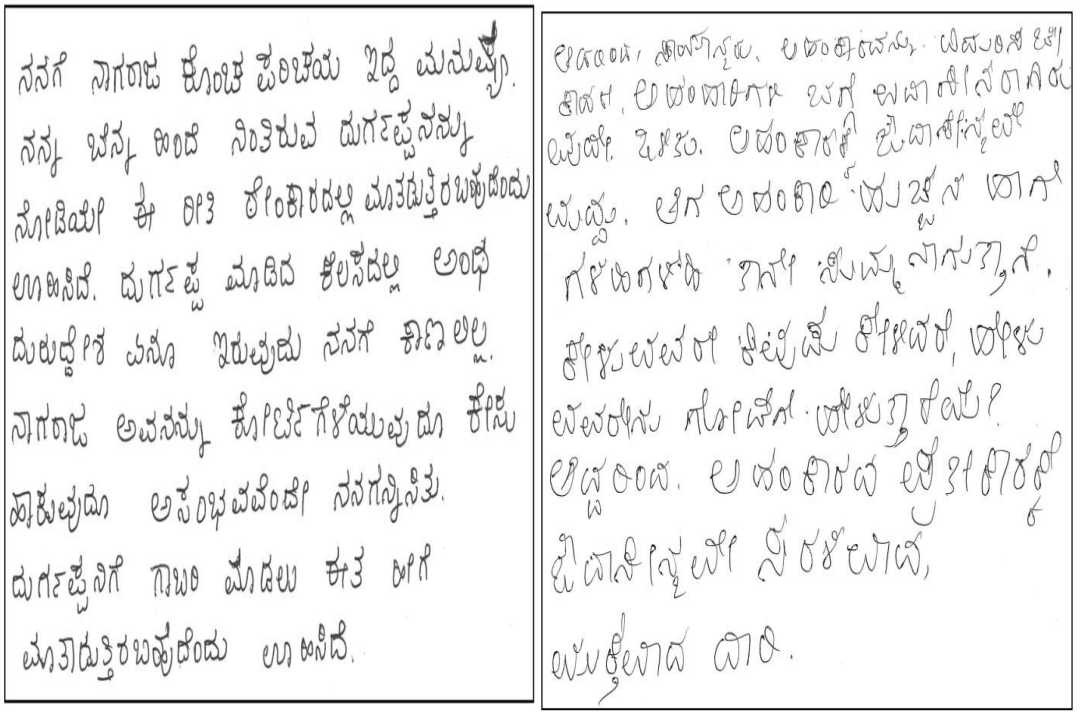


Figure 2.9 Two sample gray level scanned images of handwritten Kannada text

OTHER INDIC SCRIPTS: ROMAN, GURUMUKHI AND GUJARATI

Now, we are left with Roman and other two official scripts of India, which are Gurumukhi and Gujarati. Though Roman script handwritten dataset had been reported in many places but most of them are at character/digit level. Not only that, as a part of *PHDIndic_11,* to collect all the 11 scripts used in India, we have collected and prepared handwritten page level Roman script. Total 114 handwritten text pages are collected in the Roman part of *PHDIndic_11*. The number of text lines for Roman is 1521 with an average of 13.34 lines and 123.92 words/sub-words per text page. The first sample of the Roman text page is named as 'p_rom_0001.tif'. In Figure 2.10, two sample text images from the Roman dataset have been shown.

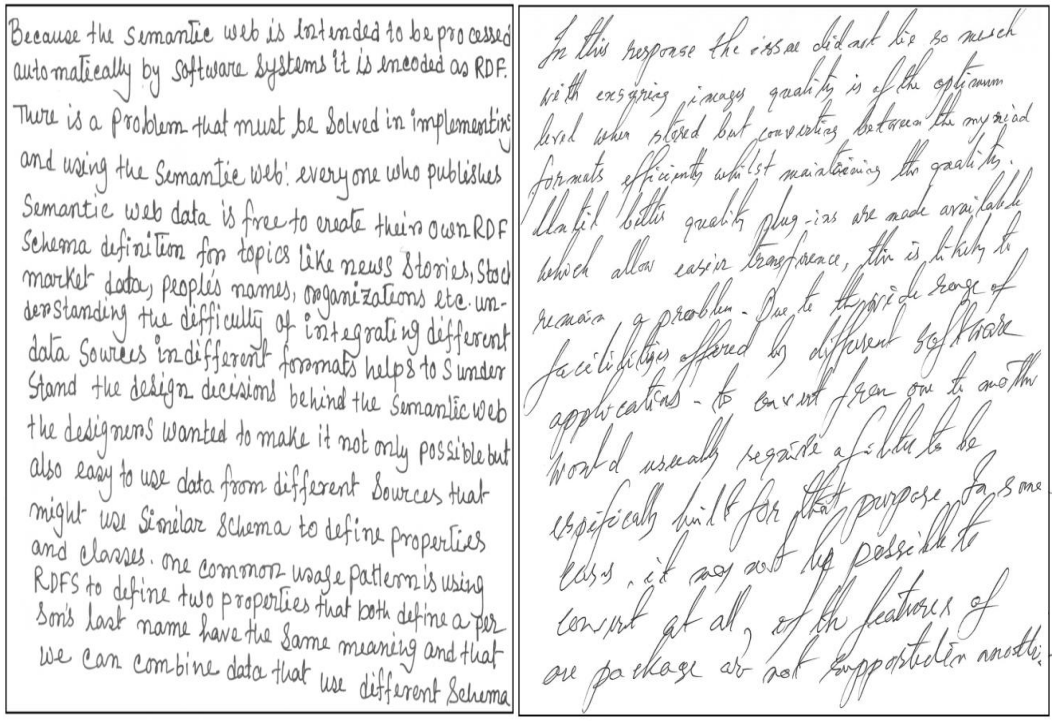


Figure 2.10 Two sample gray level scanned images of handwritten Roman text

The Gurumukhi and Gujarati part of *PHDIndic_11* contains 132 and 100 handwritten text pages respectively. The number of text lines for Gurumukhi is 1601 with an average of 12.12 lines and 98.07 words/sub-words per text page. Whereas, the number of text lines for Gujarati are 1442 with an average of 14.42 lines and 138.22 words/sub-words per text page. The first samples of Gurumukhi and Gujarati part of *PHDIndic_11*

are named as 'p_gur_0001.tif' and 'p_guj_0001.tif' respectively. In Figure 2.11 and Figure 2.12, sample text images of Gurumukhi and Gujarati have been shown respectively.

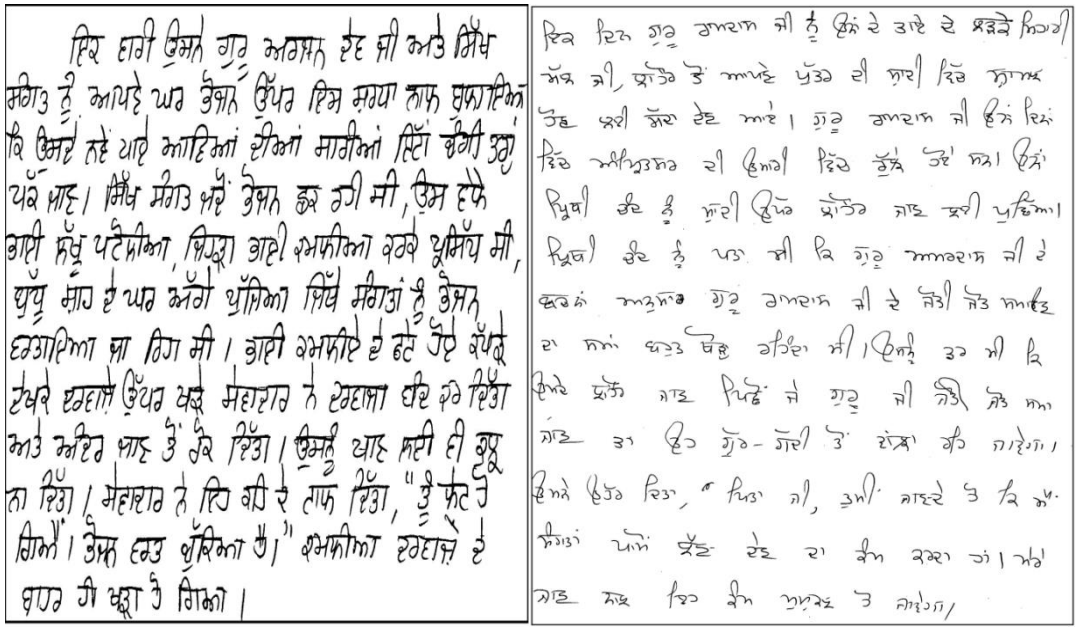


Figure 2.11 Two sample gray level scanned images of handwritten Gurumukhi text

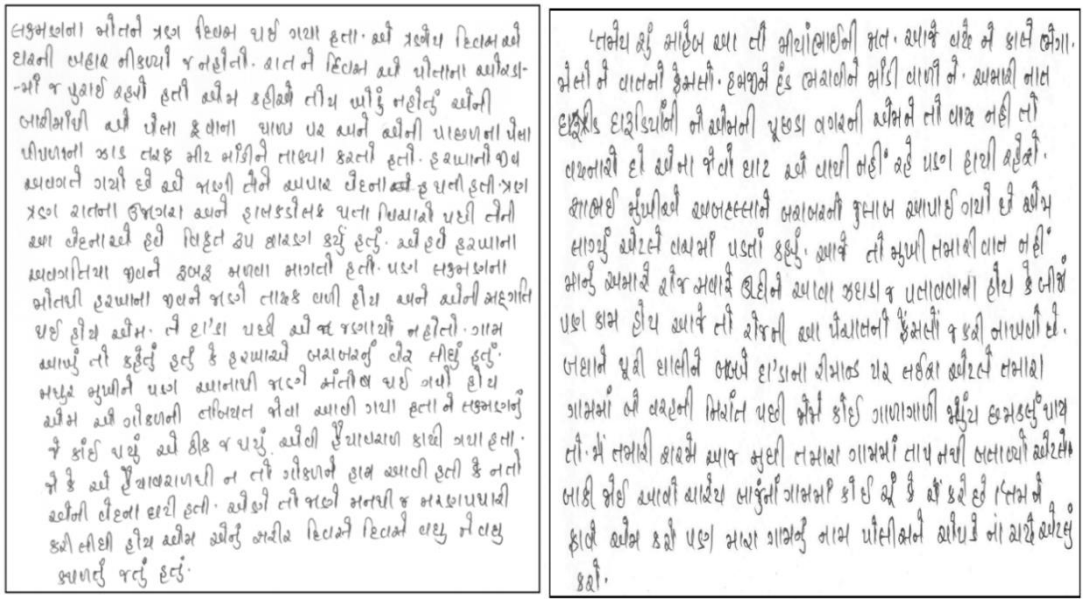


Figure 2.12 Two sample gray level scanned images of handwritten Gujarati text

SUMMARY OF *PHDINDIC_11*

PHDIndic_11 contains handwritten page level text images (comprising a fairly large amount of text pages, text lines, words/sub-words of all scripts) from 11 official scripts of India. This dataset is the first of its kind, with a collection of 1458 handwritten text pages of 11 official Indic scripts, collected from different parts of Indian subcontinent, and spread over North (Kashmir) to South (Kanyakumari) and West (Gujarat) to East (Tripura). The number of text lines in *PHDIndic_11* is 15010, with an average of 10.29 lines per text page. The number of words/sub-words in *PHDIndic_11* is 124279 with an average of 85.23 words/sub-words per page. In a nutshell, important information about *PHDIndic_11* is provided in Table 2.3.

Table 2.3 Few important statistics of the proposed *PHDIndic_11* dataset

Script	**Number of writers**	**Number of pages**	**Number of text lines**	**Number of words**	**Average number of lines per text page**	**Average number of words per text page**
Bangla	42	161	1820	12447	11.30	77.31
Devanagari	60	220	2457	23264	11.16	105.74
Urdu	45	201	1595	18422	7.93	91.65
Oriya	40	172	1422	10673	8.26	62.05
Tamil	71	120	991	5534	8.25	46.11
Telugu	46	85	826	4585	9.71	53.94
Malayalam	36	107	1028	6896	9.60	55.11
Kannada	17	46	307	1562	6.67	33.95
Roman	45	112	1521	14128	13.34	123.92
Gurumukhi	50	132	1601	12946	12.12	98.07
Gujarati	11	100	1442	13822	14.42	138.22
Total	***463***	***1458***	***15010***	***124279***	***10.29***	***85.23***

Finally, Table 2.4 shows a comparative study of *PHDIndic_11* and other popular page-level dataset such as: ICDAR [63], KHTD [68], CMATERdb1 [4] and PBOK [72]. *PHDIndic_11* contains 79.42% more pages than ICDAR, 86% more than KHTD, 89.71% more than CMATERdb1 and 51.50% more than PBOK. So, it is a fairly large

dataset proposed so far on handwritten Indic scripts. As per the number of scripts coverage, till date PBOK was the largest dataset (three Indic and one non Indic). On contrary, *PHDIndic_11* covers eleven official Indic scripts. The number of contributors of *PHDIndic_11* is also fairly large enough i.e. 463 different writers across India with varying age, sex and educational qualification. *PHDIndic_11* is benchmarked for handwritten script identification problem as it is the main focus of the thesis work. Beside script identification, the dataset can be effectively used in many other applications of DIA such as: script sentence identification/understanding, text-line segmentation, word segmentation/identification, word spotting, handwritten and machine printed texts separation and writer identification from a wide range of Indic scripts. So, *PHDIndic_11* is a unique database for document analysis in terms of scripts coverage, volume, number of contributors and variations.

Table 2.4 Comparison of *PHDIndic_11* with other popular page-level dataset

Dataset	**#Scripts**	**Scripts name**	**Statistics**	**Remarks**
ICDAR [63]	01	Roman	300 text pages	Ground truth for text line and word segmentation
KHTD [68]	01	Kannada	204 documents, 4298 lines, 26115 words	Benchmark results of line segmentation
CMATERdb1 [4]	02	Bangla, Roman	Total 150 pages	No benchmarking
PBOK [72]	04	Persian, Bangla, Oriya, Kannada	707 text pages	Benchmark results of line segmentation
PHDIndic_11 (proposed)	11	Bangla, Devanagari, Roman, Urdu, Oriya, Gurumukhi, Gujarati, Tamil, Telugu, Malayalam and Kannada	1458 pages, 15010 lines, 124279 words (contributed by 463 writers)	This dataset is benchmarked for script identification problem.

2.4.2 PRINTED WORD-LEVEL DATASET

We have prepared a word-level dataset of 13 different languages [78], which comprises of eleven different scripts [3]. Total 39K word images are considered with equal distribution of each language type, i.e. 3K words from each language. The sources of data collection were newspaper, articles and books. For example, Bangla words were collected from scanned copy of different Tagore's books, novels, poems and newspaper. As a consequence, the collected samples vary with respect to the writing style, thickness of the characters and resolution. Document image scanning was carried out using HP flatbed scanner, resolution 300 dpi and stored at 8-bit gray level jpeg format. The word dimension is found in the range of 150×50 pixels. Note that, the word images are extracted using an automated process, as explained in [24].

PREPROCESSING

Collected document images are preprocessed which includes segmentation from page/block level images to word-level images and further conversion from gray scale to binary version. Following section discuss about the preprocessing techniques.

- **SEGMENTATION INTO WORD-LEVEL IMAGES**

An automated word segmentation technique has been employed to extract word level images from the digitized images. Inter-word/line spacing is very much regular and prominent in case of printed documents in comparison to handwritten documents which helps in the segmentation process. Initially a *LSE* (Line Structuring Element) has been designed and the dimension of *LSE* was set experimentally. Then morphological dilation operation was applied using *LSE* on the complemented version of the threshold image. It will create single block for each of the word image. Then component labeling was done and word blocks were extracted applying bounding box technique on the original image file. Figure 2.13 shows a graphical illustration of thethe word segmentation process followed in the present work. Binarized images were obtained by applying the same thresholding technique that we have applied in case of *PHDIndic_11*. Table 2.5 shows sample gray-scale word images of each of the languages.

(a) (b)

Figure 2.13 (a) Original Bangla document image fragment, (b) Segmented word blocks

Table 2.5 Sample word images of different Indic languages

Language	Sample 1	Sample 2
Bangla	কলকাতা	সুৰাপানজনিত
Devanagari	जनश्रुतियाँ	आशतोष
Dogri	महत्वपूर्ण	प्रतिनिधिएं
Gujarati	વિસ્તારમાં	બદનસીબ
Gurumukhi	ਸਮੱਸਿਆਵਾਂ	ਸਹਿਯੋਗ
Kannada	ಅಧಿಕೃತ	ವರ್ತಿಸುತ್ತಿದ್ದಾರೆ
Kashmiri	استعمال	سیاستس
Malayalam	തീവ്രവാദി	ഇതിനെ
Oriya	ନିଛଘୁର	ଉପଯୁକ୍ତ
Roman	direction	SECTIONS
Tamil	கவாஅன்	குறைத்திட
Telugu	సమానమేనని	మదతు
Urdu	اخراجات	چوہدری

2.4.3 NUMERAL_DB DATASET

It is a handwritten numeral image dataset of four popular Indic scripts namely: Bangla, Devanagari, Roman and Urdu [54]. More than 5600 word level handwritten numeral images have been collected under this dataset. The whole dataset is distributed over four scripts with a distribution of 1602 words for Bangla, 1139 words for Devanagari, 1602 words for Roman and rest 1316 for Urdu. Total 43 different writers contributed to build the entire set of data. Out of these total writers, for Bangla, Devanagari, Roman and Urdu were 12, 9, 12 and 10 respectively. Efforts were taken to maintain the statistical distribution of the writers in terms of age group, sex group, and qualification group. Figure 2.14 shows some sample images of *Numeral_db*. Table 2.6 shows the statistical distribution of our present dataset mentioning script name, word count and number of writers involved.

Figure 2.14 Sample numeral words from our present dataset (a) Bangla, (b) Devanagari, (c) Roman, (d) Urdu (left to right)

Table 2.6 Statistical distribution of the *Numeral_db* dataset

Script	No. of pages	Total words	No. of writers
Bangla	12	1602	12
Devanagari	9	1139	9
Roman	12	1602	12
Urdu	10	1316	10
Total	**43**	**5659**	**43**

2.5 CONCLUSION

PHDIndic_11, a dataset of handwritten document images comprising 11 popular Indic scripts namely: Bangla, Devanagari, Gujarati, Gurumukhi, Kannada, Malayalam, Oriya, Roman, Tamil, Telugu and Urdu have been porposed. The dataset is composed of 1458 text pages written by 453 different writers. The uniqueness of *PHDIndic_11* lies in the presence of a fairly large volume of handwritten text pages from 11 official Indic scripts. It has enormous diversity in terms of the number of scripts/languages, number of writers from different geographical locations and writing styles. Further, we have reported the benchmark results on handwritten script identification. Bi-script, tri-script and multi-script identification results have been analyzed using state-of-the-art features and classifiers. In addition to *PHDIndic_11*, a printed word-level dataset of volume 39k from 13 different languages has been proposed which comprises 11 different scripts. We have also proposed *Numeral_db*, a handwritten numeral string dataset of size 5659 words from four popular Indic scripts namely: Bangla, Devanagari, Roman and Urdu.
Beside script identification, the dataset can be effectively used in many other applications of DIA, such as, script sentence identification/understanding, text-line segmentation, word segmentation/identification, word spotting, handwritten and machine printed texts separation and writer identification.

CHAPTER THREE

TECHNOLOGY AND METHODS

Any script identification system, either printed or handwritten follows the concept of pattern recognition. It relies on the fact that each script has unique visual and spatial properties which makes it possible to distinguish one script from another. So, the preliminary tasks in script identification involve finding those features from the supplied document images and then classify the documents according to the script written. Features are in general application dependent. This means, a particular technique/system is designed for a particular application/dataset. In general, texture based features are commonly used as reported in literature, but they are not capable to categorize all scripts efficiently [8]. Therefore, we combine features (script dependent/ independent) to develop a generic concept to be applied for all possible scripts. All the classifications or supervised learning systems follow three core steps: extraction of suitable features to classify the objects, classifying the features using suitable classifiers and finally evaluating the performance using important performance measuring parameters. In the following section, we discuss in detail about those features, classifiers and evaluation protocol used for Indic script identification.

3.1 FEATURE EXTRACTION TECHNIQUES

Selection of good features is the most important task in any classification problem. The selected features should be robust and easy to compute. Performance directly depends on the selection of good features. Here the term "good features" means features which will classify with more accuracy. Whereas, if the selected features are not good enough, that means there is a chance of misclassification. All the features can be classified into two broad categories: script dependent features (eg. structural feature, topological

feature, stroke based feature) and script independent features (eg. texture analysis, transform fusion). Script dependent features analyze the visual appearance of the scripts and then compute certain features which are specific to particular script. For example, features like: number of small component count, overall shape of the connected component, presence of different script specific strokes, topological property (i.e. presence or absence of 'shirorekha' or 'matra') are script dependent features. On the other hand, overall texture variation of different scripts can be identified using global texture feature. Here, without considering property of specific script, a global texture descriptor is applied on all the scripts and the intra-class difference is noted.

3.1.1 SCRIPT DEPENDENT FEATURE

To compute the script dependent features, first we visually inspect key properties of various scripts, followed by computing set of suitable features based on this observation. Script dependent featuers are categorized as: structural, topological and directional. Following section describe each of them (summarized in Table 3.1).

STRUCTURAL AND VISUAL APPEARANCE (SVA)

Based on the writing pattern associated with the script character set, stroke structure and connections, different script classes significantly differ from one another. So, structural analysis is a global measurement of an image component (connected component, i.e. continuous run of pixels) which can be used as an important shape descriptor. In our work, we have considered the following structural properties: (i) Presence of number of small components (ii) Directional chain code (iii) Circularity (iv) Rectangularity (v) Convexity and (vi) Topological distribution of the pixels or fractal dimension. Figure 1.4(a) (see Chapter 1) has shown 'matra' or 'shirorekha', which is a horizontal line over the words joining few graphemes. Two most popular Indic scripts namely Bangla and Devanagari contain this distinguishing property. Figure 1.6 (see Chapter 1) also has shown our observation regarding the presence of 'T' like structural shape within most of the Tamil script characters. Another observation of south Indian

scripts is shown in Figure 1.7 (see Chapter 1) where, difference/similarity among the direction of concavities of Tamil, Telugu, Kannada and Malayalam scripts is pointed out. Other structural features include presence of small component in scripts like Urdu, circular shaped characters of scripts like Oriya and Malayalam etc. Beside these, features like rectangularity, chain code, convexity etc. can also be used as global shape measurement of different script components. Following section discusses about the structural features which we have implemented on Indic scripts.

- **SMALL COMPONENT ANALYSIS**

Dimensionality is an important measure in component analysis [7]. We have classified all the script components into three major categories namely (i) *LC* (Large Component), (ii) *MC* (Medium Component) and (iii) *SC* (Small Component). Different component sizes are computed based on these categories and these values are stored in the feature table. An algorithm for computation of component dimensionality is shown. The threshold value considered for our experiment is 5. Figure 3.1 shows presence of "dot" like small components in Urdu script.

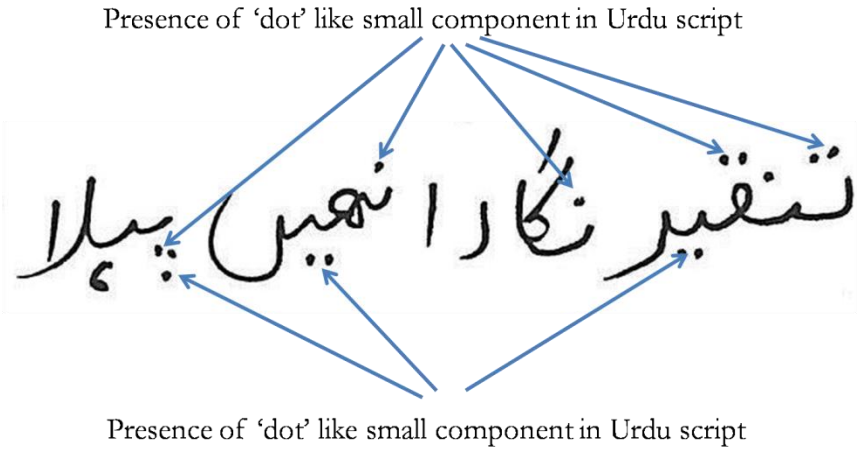


Figure 3.1 Presence of "dot" like small component in Urdu script characters

It is found that among all the eleven official Indic scripts, Urdu contains maximum number of small components which is a distinguishing property of Urdu script.
The Algorithm 3.1 for computation of small components is shown below:

Algorithm 3.1:

Algorithm for computation of small component:

Initially set SC=0;

Using component analysis each component is

considered and pixel count is done.

If Number of Pixel (NOP) <= Predefined threshold

SC++;

- **CHAIN CODE**

The presence of different directional strokes like horizontal, vertical, left and right diagonal or any stroke with arbitrary orientation can be captured using chain code. First, contours of image component are drawn, and then 8-directional chain code is drawn on the contours (both inner and outer contour). So, the code of the components of different scripts will differ from each other. Then we compute chain-code direction histogram values as feature. Figure 3.2 shows an example of popular 8-directional freeman chain code computed for Bangla character 'ব'. 'Matra' or 'Shirorekha' feature can also be identified from the chain code values computed on Bangla or Devanagari scripts.

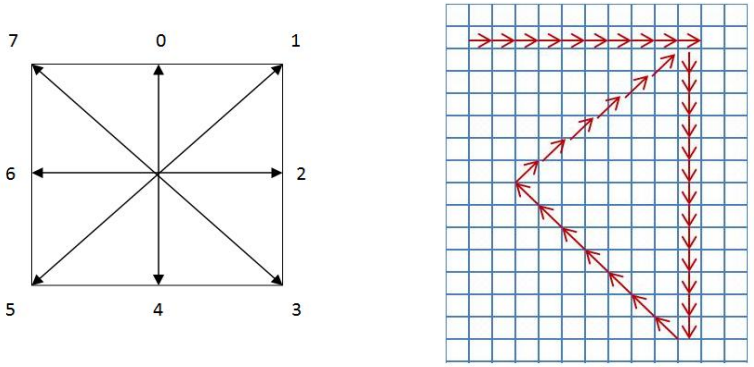


Figure 3.2 Example of 8-directional chain-code and the same computed on a sample Bangla character 'ব'

- **CIRCULARITY**

Computation of circularity of image component can be used as one of the key feature [79]. It is observed that scripts like Oriya, Malayalam etc. have more circular components compared to others. Following is the algorithm for calculation of circularity of a component.

Algorithm 3.2 computes circularity of an image component:

Algorithm 3.2:

- *At first, minimum enclosing circle is drawn. This enclosing circle will cover the component minimally. The radius of the enclosing circle is stored in a variable say* R_1
- *Then circle fitting is done. This operation will fit a circle in the component in as minimum manner as possible. Radius of the fitted circle is stored in a variable say* R_2.
- *The difference of the two radiuses* R_1 *and* R_2 *are stored in a variable say* R. *This value of* R *indicates the proximity of circularity of the component. In optimum case the value of* R *will be zero which stands absolute circular component.*

$$So,\ R = R_1 - R_2 \qquad (3.1)$$

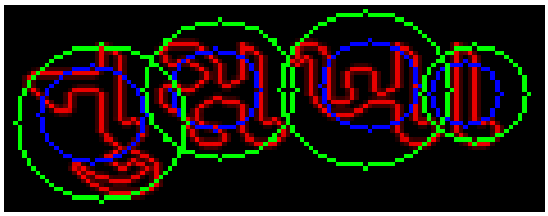


Figure 3.3 Illustration of Circularity property on Gujarati script components using fitted circles (blue: minimum encapsulating & green: best fitted)

In fact the complete or almost complete circular components will have their *R* value tending to zero. Figure 3.3 shown computation of circularity feature on Gujarati script.

- **RECTANGULARITY/BOUNDING BOX**

Bounding box/Rectangularity is used to measure the shape of the component whether perfectly square or not. Three measures are taken here: (i) perfect square (height/width

= 1) (ii) 'horizontal rectangle' (height/width < 1) and (iii) 'vertical rectangle' (height/width > 1). The script with 'matra' will have larger bounding box size compared to the scripts without-'matra' due to the presence of larger component size (as 'matra' joins different characters). So, it is a distinguishable shape descriptor. To compute the feature we measured the height, width, aspect ratio. Figure 3.4 shows sample output of bounding box computation.

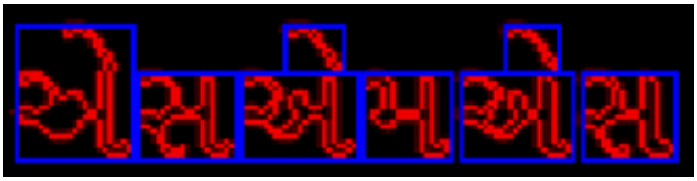


Figure 3.4 Illustration of Rectangularity property on Gujarati script components (blue: rectangular box)

- **CONVEX HULL**

Convex hull is computed to comprehend the shape of the components [79]. The hull is computed for every selected component's inner and outer contours in the proposed method. Minimum and maximum of surrounding of both the inner and outer contour of the component is computed. Their average values and variance are also calculated. An example of computation of convex hull is shown in

Figure 3.5. This feature is very much useful to comprehend the overall shape variation and convex shape of different Indic scripts.

Figure 3.5 Illustration of Convex hull property on Urdu script components

TOPOLOGICAL FEATURE: SEPARATION OF 'MATRA' BASED SCRIPT

The problem of script identification depends on the fact that different scripts have unique visual attributes and spatial pixel distribution which make it distinguishable from

others. So, the primary task associated with script identification is to devise a technique to identify these features from a document image and then classify document's script accordingly. As mentioned earlier, few of the demographically popular Indic scripts contain an important topological property which is known as 'matra' or 'shirorekha'. 'Matra' joins different characters of such scripts, resulting in a longer connected component. Example of such 'matra' based scripts is: Bangla and Devanagari (see Figure 3.6). Topological dimension can be one effective approach to separate 'matra' based scripts from their counter part. This is because, if we compute average topological dimension of top and bottom profile of connected components, then there will be a significant difference between the average topological dimension of 'matra' based scripts and the scripts without having 'matra'.

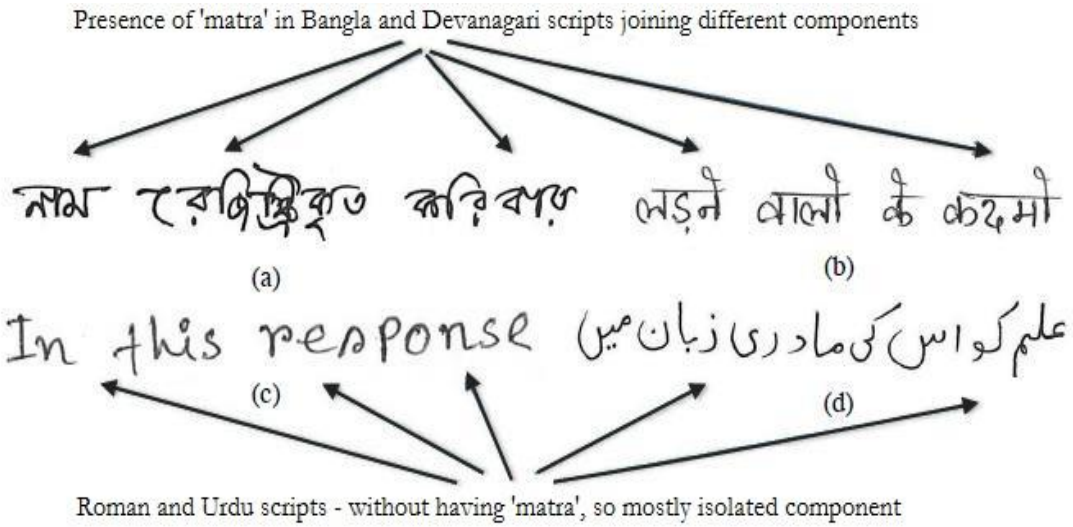


Figure 3.6 Presence of `matra' in (a) Bangla, (b) Devanagari scripts and the same is absent in (c) Roman, (d) Urdu scripts in Roman. `Matra' joins different charactes resulting a large connected component (in case of (a) and (b)), whereas, component size is relatively smaller for scripts without `matra' (in case of (c) and (d))

As our problem solely relies on 'matra' separation and then script classification [80], we choose such state-of-the-art techniques which are able to do so. We considered following three different features:

- Fractal geometry analysis (FGA),
- Canny edge detector (CED) and

- Morphological line transform (LT)

- **FRACTAL GEOMETRY ANALYSIS (FGA)**

The present work is motivated by the concept of Fractal Geometry Analysis or in short FGA of an object [7] [12] [80]. A fractal is an irregular geometric object with an infinite nesting of structure at all scales (self-similarity). Formally a fractal is defined as a set for which the Hausdorff-Besikovich [81] dimension is strictly larger than the topological dimension. The fractal dimension can play an important role towards object analysis in an image. The geometric characteristics of the objects or connected components on an image can be understood by fractal dimension. So by performing fractal analysis, researchers typically estimate the dimension of connected components in an image. The fractal dimension of continuous object is an entity specified in terms of well-defined mathematical limiting processes.

The fractal theory developed by Mandelbrot and Van Ness was derived from the work of mathematicians Hausdorff and Besikovich. The Hausdorff-Besikovich dimension (DH) is defined by the following equation:

$$D_H = \lim_{\varepsilon \to 0^+} \frac{\ln N_\varepsilon}{\ln 1/\varepsilon} \qquad (3.2)$$

where N_ε is the number of elements of ε diameter required to cover the object. Mandelbrot defines a fractal as a set for which the Hausdorff-Besikovich dimension strictly exceeds the topological dimension.

When working with discrete data, one is interested in a deterministic fractal and the associated fractal dimension (D_f) which can be defined as the ratio of the number of self-similar pieces (N) with magnification factor (1/r) into which an image may be broken. However, the surfaces of many objects cannot be described with an integer value. These objects are said to have a "fractional" dimension. D_f is defined as:

$$D_f = \frac{\ln N}{\ln 1/r} \qquad (3.3)$$

D_f may be a non-integer value, in contrast to objects lying strictly in Euclidean space, which have an integer value. However, D_f can only be directly calculated for a deterministic fractal. There are varieties of applicable algorithms for estimating D_f, and we have used Box-counting algorithm for the same.

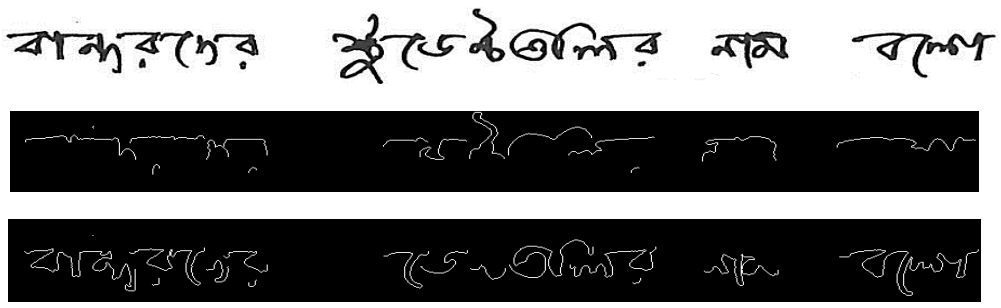
The upper part and the lower part play a significant role in feature extraction from the document image. This observation motivated us to solve the present problem by FGA. Indic scripts can be categorized as 'shirorekha' based and non-'shirorekha' based with respect to topological structure. A 'shirorekha' is a horizontal line present on upper part of few scripts which joins different characters in words or words in lines. Bangla and Devanagari are two popular 'shirorekha' based scripts. Whereas Roman and Urdu are two popular scripts that contains no 'matra' or 'shirorekha'. So if pixel density of the connected components is calculated, there will be difference in pixel density of upper part and lower part of the components of different scripts. As shown by Table 3.1, the size of fractal based features is only two and as it is computed directly on the pixels so it is very fast. So, to separate scripts with matra from without having matra this feature will take lesser time compared to others.

The following algorithm computes average fractal dimension of connected components.

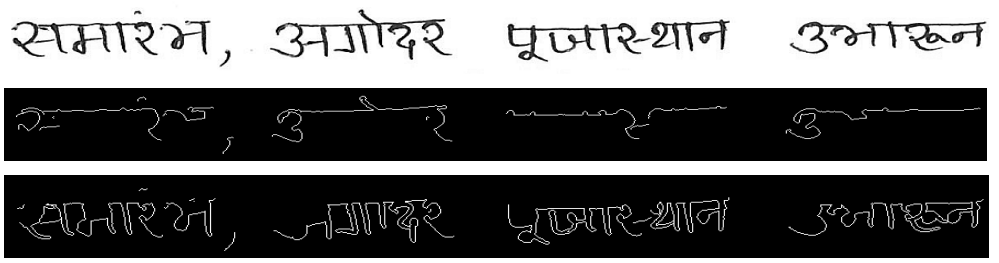
Algorithm 3.3:

- *Compute D_f from both upper (D^u_f) and lower (D^l_f) parts of each image component.*
- *Take the average of both upper and lower components: $D^u_{f,avg}$ and $D^l_{f,avg}$, respectively.*
- *Compute their ratio: $D^u_{f,avg}$ / $D^l_{f,avg}$*

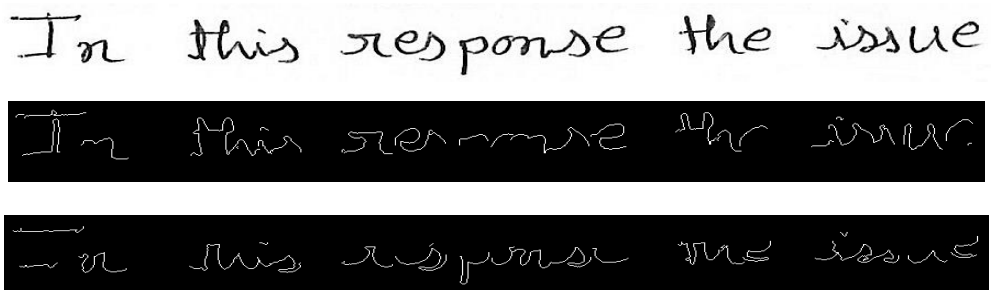
In Figure 3.7, sample results are shown for Bangla, Devanagari, Roman and Urdu scripts



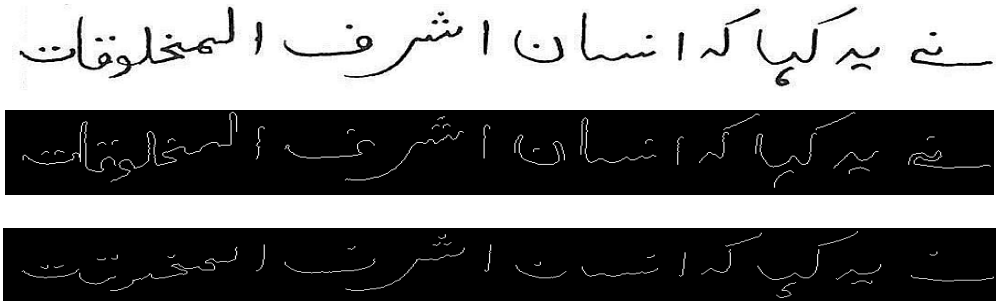
(a)



(b)



(c)



(d)

Figure 3.7 Illustrating fractal dimension of (a) Bangla, (b) Devanagari, (c) Roman and (d) Urdu scripts, where topmost part shows original line level document image, middle and lower part show fractal dimension D_f of upper profile and lower profile, respectively for each of the four scripts (a)-(d)

- **CANNY EDGE DETECTOR (CED)**

The process of Canny edge detection algorithm [35] can be broken down to 5 different steps.

- **Apply smoothing:** It refers to blurring, which attempts to remove noise. For this, a Gaussian filter is applied to convolve with the image,

$$G(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}} \quad (3.4)$$

- **Compute the intensity gradients:** An edge in an image may point in a variety of directions. In case of Canny algorithm, four filters are used to detect horizontal, vertical and diagonal edges in the blurred image. The edge gradient and directions (by using G_x and G_y) can be determined by:

$$G = \sqrt{{G_x}^2 + {G_y}^2} \quad (3.5) \text{ and}$$

$$\theta = atan2(G_y, G_x) \quad (3.6)$$

  Note that the edge direction angle is rounded to one of four angles representing vertical, horizontal and the two diagonals: 0, $\pi/4$, $\pi/2$, $3\pi/4$.

- **Apply non-maximum suppression:** It is an edge thinning technique; it helps to get rid of spurious response to edge detection.
- **Apply double threshold:** It determines potential edges, by using two different thresholds: high and low that are empirically set. High threshold yields strong edges, and in the same way, low threshold yields weak edges. Edges are suppressed if the pixel value is smaller than the low threshold value.
- **Track edge by hysteresis:** It finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

In our case, we apply CED on script image, as shown in Figure 3.8. Since we are interested in separating scripts with 'matra', we calculate pixel density from the upper block.

Figure 3.8 Sample output after applying Canny edge detector algorithm on (a) Bangla, (b) Devanagari, (c) Roman and (d) Urdu scripts

- **LINE TRANSFORM (LT)**

Considering 'matra' in our script, we aim to extract by using LT. For this, we convolve an original image with a kernel. The kernel is defined as a linear structuring element that decides the nature of morphological operations: erosion and dilation are considered. In this study, to duplicate 'matra'-like image component, our kernel (linear structuring element) of size 1×10 (i.e., $[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$) is considered. Consider an image $I\ (x, y)$ and a kernel $K\ (u, v)$, both operations: erosion and dilation can be generally expressed as, respectively:

$$I_{ero} = I \ominus K = \min\{I(x+u, y+v) - S(u,v)\} \quad (3.7) \text{ and}$$

$$I_{dil} = I \oplus K = \max\{I(x-u, y-v) + S(u,v)\} \quad (3.8)$$

In our study, we apply this technique on image component and calculate pixel density as in CED. Figure 3.9 provides outputs of LT from four different scripts.

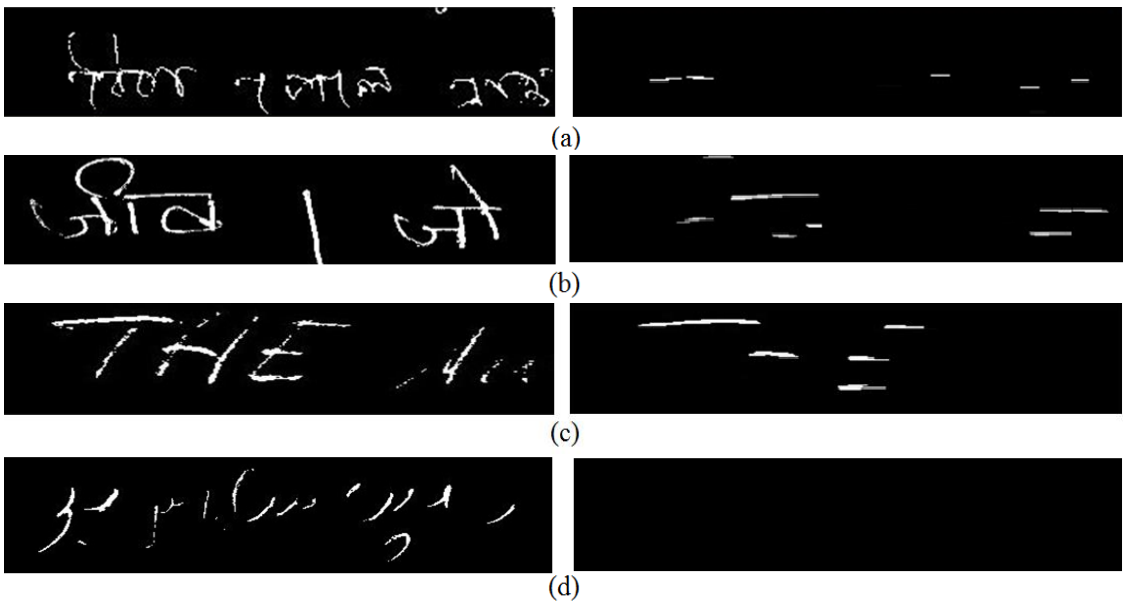


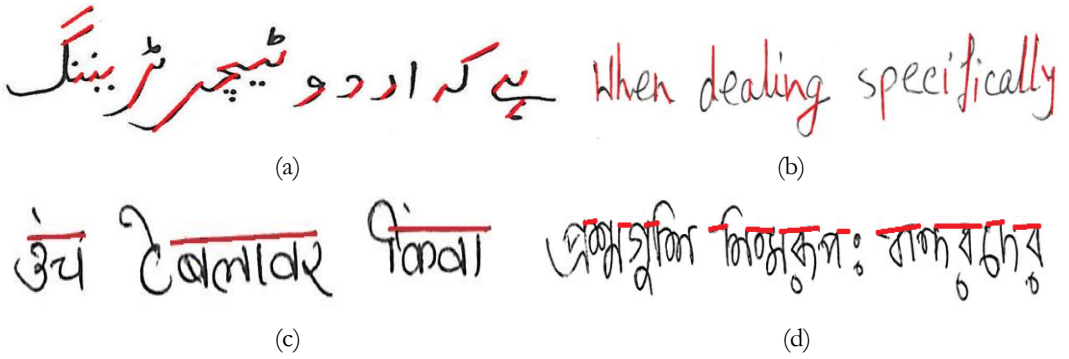
Figure 3.9 Illustration of line transforms output on (a) Bangla, (b) Devanagari, (c) Roman and (d) Urdu scripts. The first column shows original image and second one shows output image after applying line transform

DIRECTIONAL STROKE IDENTIFICATION (DSI)

Different directional strokes are present in different Indic scripts. Urdu script has many characters which have about 75^0 directional strokes (see Figure 3.10). Roman script has characters with about 45^0 strokes. Bangla and Devanagari scripts contain 'matra' which is an 180^0 stroke. Besides these, other scripts also have different directional strokes with arbitrary orientations. To capture stroke features, we have used directional morphological reconstruction with directional kernels [52] [82]. The morphological operations considered in this work are: image dilation, erosion, opening, closing, top-hat and black-hat transforms. Based on our visual observation of the different directional strokes presence in different Indic scripts, first we define four directional morphological kernels: *H-kernel* (horinzontal direction), *V-kernel* (vertical direction), *RD-kernel* (right diagonal direction) and *LD-kernel* (left diagonal direction). These kernels are 3x11, 11x3, 11x11 and 11x11 matrices correspondingly, where horizontal, vertical, right diagonal and left diagonal pixels are 1 and rests are 0. A sample *H-kernel* is as follows:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

To compute the feature values, at first the original image is dilated using a default kernel. Then each of the dilated images is eroded four times using four directional kernels (i.e. *H-kernel*, *V-kernel*, *RD-kernel* and *LD-kernel*). The ratio of those eroded images with the dilated one gives 4 features and computation of the average and standard deviation of the eroded images give other 8 features, resulting into a total of 12 features. In a similar way other morphological operations, namely opening, closing, gradient, top-hat and bottom-hat were performed, where each of them generates 12 features. Finally, under this category, 72 dimensional feature set is generated.



(a) (b)

(c) (d)

Figure 3.10 Different directional strokes in Indic scripts shown by overwriting on the original image (line fragment) using red color (a) slanting strokes (60^0 to 90^0 orientations) in Urdu, (b) vertical and diagonal strokes in Roman, (c) (d) horizontal strokes due to 'shirorekha' or 'matra' in Devanagari and Bangla

Table 3.1 Summary of the script dependent features

Enumeration	**Feature type**	**Feature description**	**Feature Dimension**
$\boldsymbol{FS_{SVA}}$	Structural and visual appearance	Chain code based feature on outer and inner contour	16
		Circular or roundness of an image component	10
		Bounding box fitting as a global measure	8
		Convexity of a component as a global measure	8
		Dimension @ $\boldsymbol{FS_{SVA}}$	***42***
$\boldsymbol{FS_{FGA}}$	Fractal dimension	Avg. fractal dimension of upper part of the contour	01
		Avg. fractal dimension of lower part of the contour	01
		Dimension @ $\boldsymbol{FS_{FGA}}$	***02***
$\boldsymbol{FS_{DSI}}$	Directional strokes	Ratio of the eroded and *dilated* image	04
		Average and standard deviation	08
		Previous two steps using morphological *opening*	12
		Previous two steps using morphological *closing*	12
		Previous two steps using morphological *gradient*	12
		Previous two steps using morphological *top-hat*	12
		Previous two steps using morphological *black-hat*	12
		Dimension @ $\boldsymbol{FS_{DSI}}$	***72***

3.1.2 SCRIPT INDEPENDENT FEATURE

These features are not script specific. So, they are applied globally to different scripts and output responses are measured. They are described in the following section (summerized in Table 3.2).

TEXTURE ANALYSIS

Gray Level Co-occurrence Matrix (GLCM)

The GLCM (Gray Level Co-occurrence Matrix) is a statistical calculation of how often different combination of gray level pixel values occur in an image. It has been the workhorse for textural analysis of images since the inception of the technique by Haralick et al. [83]. GLCM matrix describes the frequency of occurrence of one gray level with another gray level in a linear relationship within a defined area. Here, the co-occurrence matrix is computed based on two parameters, which are the relative distance between the pixel pair d measured in pixel number and their relative orientation φ. Normally, φ is quantized in four directions $(0^0, 45^0, 90^0, 135^0)$. The GLCM is a matrix where the number of rows and columns are equivalent to the number of gray levels of the image. The matrix element $P(i, j \mid \Delta x, \Delta y)$ is the relative frequency with which two pixels, separated by a pixel distance$(\Delta x, \Delta y)$, occur within a given neighbourhood, one with intensity i and the other with intensity j. One may also say that the matrix element $P(i, j \mid d, \theta)$ contains the second order statistical probability values for changes between gray levels i and j at a particular displacement distance d and at a particular angle (θ). Detail description of GLCM is available in [83]. In the current approach the GLCM is calculated with Contrast, Correlation, Energy and Homogeneity statistical measures in all four directions considering both type of pairs like $P[i, j]$ and $P[j, i]$. Figure 3.11 shows a sample GLCM calculation technique considering four directions and eight gray levels.

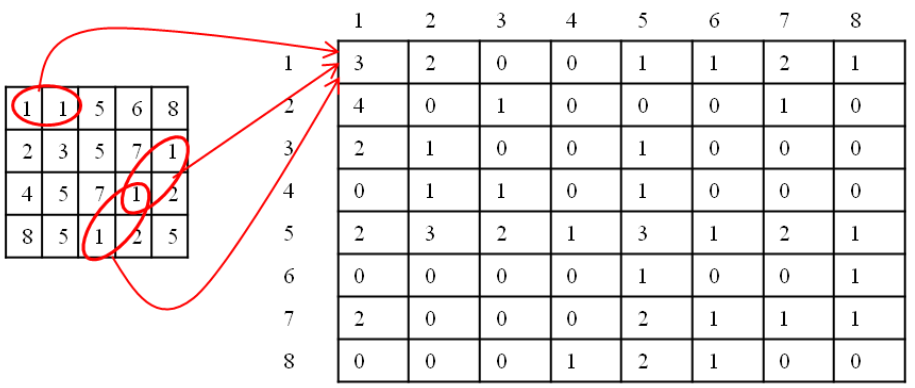


Figure 3.11 Schametic diagram of computation of GLCM (Gray Level Co-occurrence Matrix)

Gabor Filter Bank

It is a convolution based technique used widely for texture analysis [84] [85]. The response of Gabor filter to an image is determined by the 2-D convolution operation. In general the filter will convolve with the input image signal and a Gabor space is generated. If *I(x,y)* is an image and *G(x, y, f, ϕ)* is the response of a Gabor filter with frequency *f* and orientation ϕ to an image on the *(x,y)* spatial coordinate of the image plane (refer Eq. no. 3.9).

$$G(x,y,f,\emptyset) = \iint I(p,q)g(x-p,y-q,f,\emptyset)dpdq \qquad (3.9)$$

In the proposed approach, multiple feature values are computed forming a Gabor filter bank. Experimentally we set the filter with frequency 0.25 and orientation of 60º, 90º, 120º and 150º for computations of varying Gabor filter inspired features. Afterwards the standard deviation of the real part and imaginary part are considered as feature values [84].

Spatial energy (SE)

SE distribution varies in accordance with the change in textural information, and therefore, it is important in our study [53]. SE distribution is observed by computing entropy on the grayscale images. It can be represented by:

$$Entropy = -\sum p(i,j)\log\big(p(i,j)\big) \qquad (3.10)$$

In general, entropy is complement of energy. Therefore, for any non-uniform or aperiodic gray level distribution, there exists high entropy.

Another measure is the standard deviation of binary images of different scripts. Standard deviation is a measure of the variability of the image pixels. It can be represented by:

$$\sigma_x = \sqrt{\frac{1}{n}\left\{\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i^2\right)^2\right\}} \quad (3.11)$$

Where, $x_1, x_2, \ldots, x_n$ be n observations of a random variable X, which is representation of an arbitrary image pixel.

Wavelet Energy (WE)

Wavelet is used for multi-resolution image analysis. In the work of handwritten numeral script identification, we have used wavelet as the sole pertinent feature [86]. For this work, wavelet packets are generated using DWT or discrete wavelet transform which uses sub-band coding on images with respect to spatial and frequency components and allows analysis the images from coarse to fine level [87]. Here Daubechies wavelets *db*N where N = 1, 2, 3 are chosen to generate sub-band images with approximation coefficients *cA*, *cH*, *cV* and *cD*. Their advantage includes computational ease with minimum resource and time requirements. These orthogonal wavelets are characterized by maximum number of vanishing moments for some given support. Here, a signal (for present work it is a word image) is decomposed into different frequencies with different resolutions for further analysis. In general the family of Daubechies wavelet is denoted as *db*N, where the family is denoted by the term *db* and the number of vanishing moments is represented by N.

It is observed that, an image can be represented by the combinations of different coefficients i.e constant, linear, quadratic etc. Daubechies *db1* represents the constant coefficient of the image component, *db2* represents the linear and *db3* can represent quadratic coefficients. So, wavelet decomposition at level 1 is done using *db1*, *db2* and *db3* which capture the constant, linear and quadratic coefficients of an image

component. Four coefficients namely approximation coefficients (*cA*), horizontal coefficients (*cH*), vertical coefficients (*cV*), and diagonal coefficients (*cD*) are computed. To measure the WE or wavelet energy feature we have computed wavelet entropy on these approxcimation coefficients for each of the sub-band images. Suppose *ws* is the word level image signal and $(ws_i)_i$ the coefficients of *ws* in an orthonormal basis, then the normalized shanon entropy is defined by Eq. no. 3.12 and 3.13.

$$SE(ws_i) = (ws_i^2)\log\,(ws_i^2) \qquad (3.12)$$

$$\text{So, } SE(ws_i) = -\sum(ws_i^2)\log\,(ws_i^2) \qquad (3.13)$$

Figure 3.12 shows Computation of different Daubechies wavelet coefficients at level 1 on Bangla numeral Word-level image.

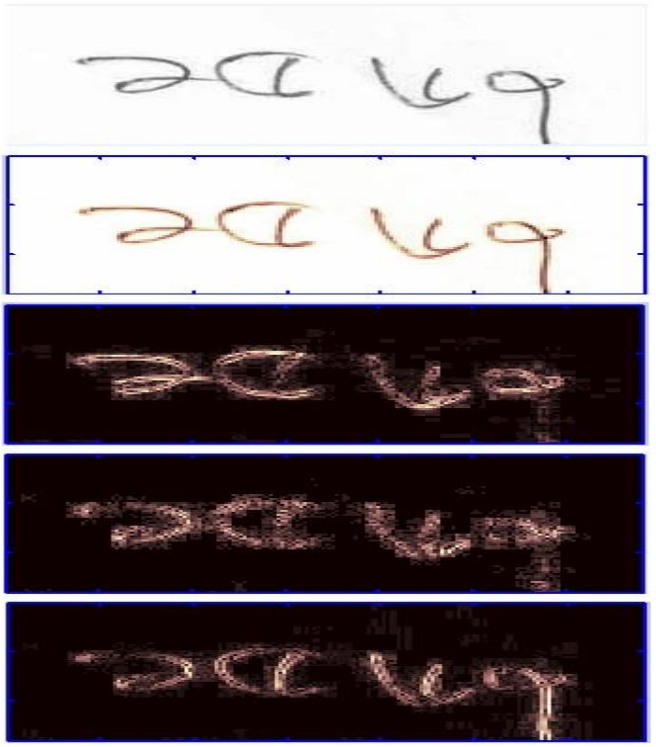


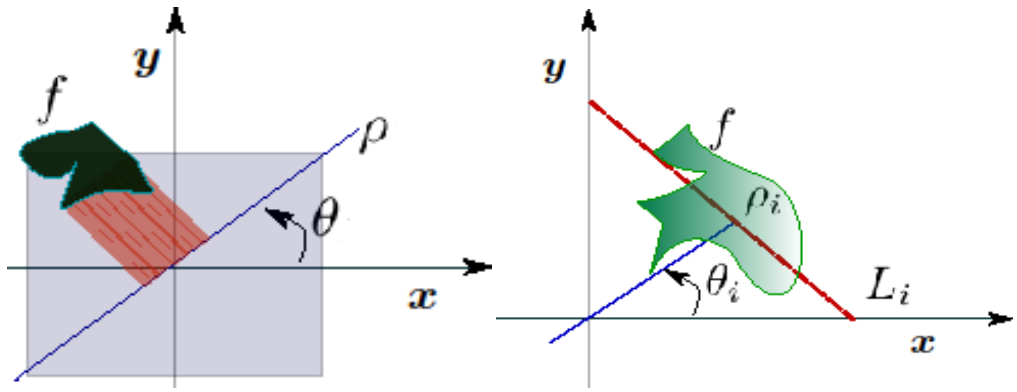
Figure 3.12 Computation of different Daubechies wavelet coefficients at level 1 on Bangla numeral Word-level image (top to bottom: original Bangla word image, approximation coefficient *cA1*, horizontal coefficient *cH1*, diagonal coefficient *cD1*, vertical coefficient *cV1*

THE RADON TRANSFORM

Motivated by the presence of the strokes at different orientations in the word images, we propose to use of the Radon Transform (RT) [88] [53]. The RT consists of a collection of projections of a pattern at different angles [89], as illustrated in Figure 3.13. In other words, the radon transform of a pattern $f(x, y)$ and for a given set of angles can be thought of as the projection of all non-zero points. This resulting projection is the sum of the non-zero points for the pattern in each direction, thus forming a matrix. The matrix elements are related to the integral of f over a line $L(\rho, \theta)$ defined by $\rho = \mathrm{x}\cos\theta + y\sin\theta$ and can formally be expressed as, in Eq. no. 3.14

$$R(\rho, \theta) = \iint_{-\infty}^{\infty} f(x, y)\delta(x\cos\theta + y\sin\theta)dxdy \qquad (3.14)$$

Where, $\delta(.)$ is the Dirac delta function, $\delta(x) = 1$, if x=0 and 0 otherwise. Also, $\theta \in [\,0, \pi\,)$ and $\rho \in]-\infty, \infty[$. For the RT, L_i be in normal form (ρ_i, θ_i)
Figure 3.13 (a) and Figure 3.13 (b) shows the working principles of RT.



(a) Projection at angle θ (b) Definition of RT

Figure 3.13 The Radon transform

Such a description is useful for scripts such as Bangla and Devanagari, where there exists horizontal line, known by the name 'matra' or 'shirorekha'. These clear lines can be exploited by computing 0^0 projection. Similarly, scripts like Tamil and Roman have

many vertical lines which can be represented by 90^0. However, to exploit meaningful information, we do not require all possible orientations, and therefore, we study the RT at an interval of 15^0. The RT spectrum computed on different Indic scripts is shown in Figure 3.14.

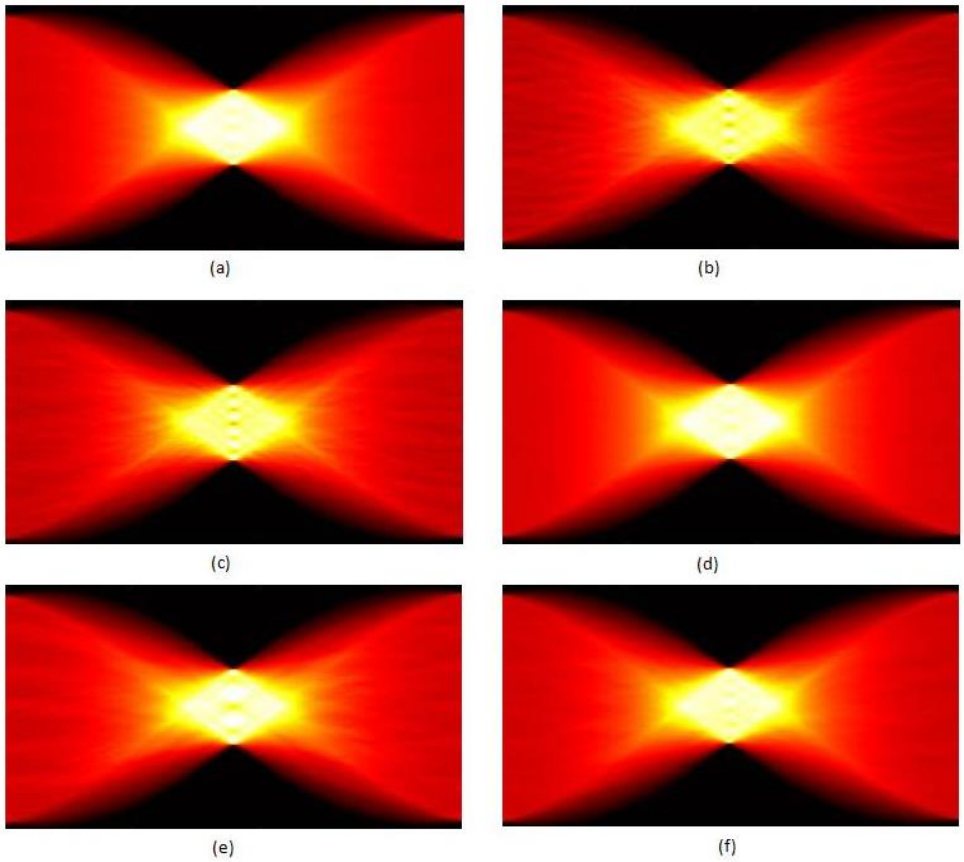


Figure 3.14 RT spectrum computed on different script images, (a) Bangla (b) Devanagari, (c) Malayalam (d) Oriya (e) Roman (f) Urdu. (RT spectrums are shown on 32x32 images)

3.1.3 IMAGE TRANSFORM FUSION

WAVELET-RADON TRANSFORM (WRT)

Experimentally we have found that, the performance of wavelet can be further optimized if it is combined with radon transform with proper tuning. So, in our work

[54], for feature extraction, these two frequency domain techniques (Discrete Wavelet Transform and Radon Transform) are combined to form a new hybrid technique named as *WRT* (Wavelet Radon Transform). The *WRT* features are computed as follows: Firstly, *DWT* decomposition of the input binary images is done using Daubechies *db4* decomposition and four sub-band images are produced at the first level. Then *RT* is computed on each of them (*cA, cH, cV, cD*) at seven different rotation angles starting from 0^0 and ending at 180^0 varying with a distance of 30^0. We considered $\theta = (0^0, 30^0, 60^0, 90^0, 120^0, 150^0, 180^0)$ for present experiment. Finally some local features namely entropy, mean, standard deviation are computed on each of the *WRT* spectrum to generate the final feature set. The block diagram of the proposed fusion technique is shown in Figure 3.15.

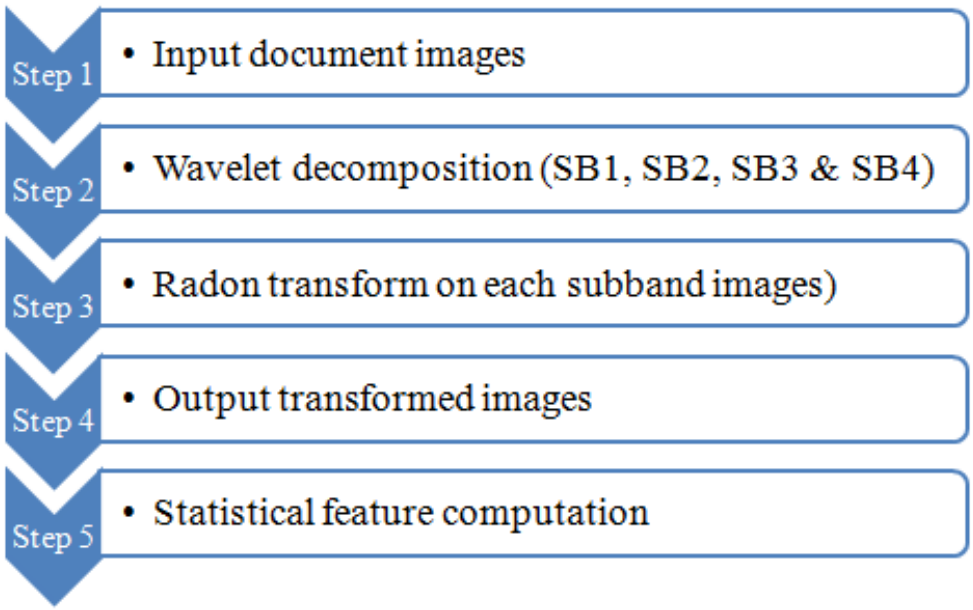


Figure 3.15 Steps for computation of Wavelet Radon Transform based features

INTERPOLATED MORPHOLOGICAL TRANSFORM

Image upsize and downsize operation is performed using interpolation. This property is combined with directional morphological operation to form a new feature vector known as Interpolated Morphological Transform or *IMT* [54]. The flow diagram of IMT operation is shown in Figure 3.16. Initially, image dilation is performed using

default 3x3 kernel [90], then the images are interpolated using different mechanism namely nearest neighbor, bilinear, pixel area re-sampling method, bicubic interpolation. Normally nearest neighbor interpolation takes the closest pixel value for resizing calculation. The 2x2 surroundings are taken for bilinear operation. The virtual overlapping between the resized image and original image is performed and then the average of the covered pixel values is computed in case of pixel area re-sampling method. For bicubic operation, a cubic spline between the 4-by-4 surrounding pixels in the source image is fitted, and then reading off the corresponding destination value from the fitted spline is performed. Finally, the ratio of the interpolated image and morphological image obtained by applying directional kernel is computed as feature values.

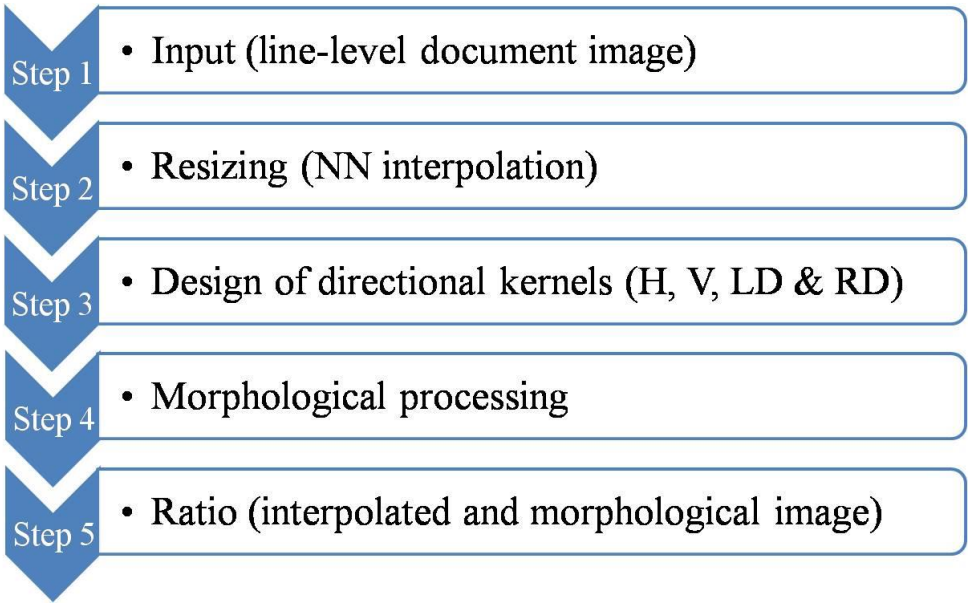


Figure 3.16 Steps for computation of interpolation based feature

Table 3.2 Summary of the script independent features

Enumeration	Feature type	Feature description	Feature Dimension
$\boldsymbol{FS_{GLCM}}$	Gray level co-occurance matrix	Co-occurance matrix at four offsets [0 1] [-1 1] [-1 0] [-1 -1] and then compute local features	40
$\boldsymbol{FS_{GABOR}}$	Gabor filter bank	Gabor filter based feature with varying orientations (60º, 90º, 120º and 150º) and frequencies (0.25)	08
$\boldsymbol{FS_{SE}}$	Saptial energy	Energy of the gray-scale image	04
$\boldsymbol{FS_{WE}\#1}$	Wavelet energy	Decomposing input image at *db1* (constant), *db2* (linear) and *db3* (quadretic) level and then computing the energy at each level	15
$\boldsymbol{FS_{WE}\#2}$	Wavelet energy	Computation of approxcimation, horizontal, vertical and diagonal approxcimation coefficients at *db1*, *db2* and *db3* level of an image and then measure the energy at each level	51
$\boldsymbol{FS_{WRT}\#1}$	Fusion of Wavelet and Radon transform	Four approxcimation coefficients from the image at *db1*, *db2* and *db3*, generating 12 wevelets. RT spectrum at $\theta = (0^0, 30^0, 60^0, 90^0, 120^0, 150^0, 180^0)$ on the original and each of these sub-bands and then computing the 04 energy features from each of them.	52
$\boldsymbol{FS_{WRT}\#2}$	Fusion of Wavelet and Radon transform	Same as $FS_{WRT}\#1$ in addition with 13 local features (i.e. min of value of the RT spectrum)	65
$\boldsymbol{FS_{IMT}}$	Interpolated morphological transform	Interpolated (NN) images are passes trough four directional kernels and then applying morphological dilation and erotion on them in a similar way of FS_{DSI}	24

Different feature set mentioned in Table 3.1 and Table 3.2 is used individually or in combination to solve different printed and handwritten script identification problems. The out come are discussed in Chapter 4 and Chapter 5.

3.2 CLASSIFICATION

After feature extraction and labeling of the features with the particular script the immediate task is classification. Classification includes a decision-theoritic approach to the identification of scripts from the document images. So, in our problem, classification analyzes the numerical properties (extracted as feature values) of various image features and organizing images into different script catagories. All the classification algorithm typically considers two phases of processing: training and testing/validation. Training phase is typically assumed as "gold standard" data, where we train our model by pairing our input with the expected output. During the test phase we estimate how well our model is trained (sometimes it depends on the size of the data, input etc. factors). There are several performances measuring parametes we consider in our work like: overall accuracy rate, classification errors, model building time etc.

CROSS-VALIDATION

During experimentation, sometimes we follow k-fold cross validation approach. This approach is also called rotation estimation. In this case, all the sample images are initially divided into k different subsets. Out of the k subsets, one subset is kept for the validation data for testing the model and the remaining subsets (k-1 number) are used as training data. This process is repeated k times or k folds, where each of the k-1 subsets is used as the validation test data [79] [52].

In the present work, we have used state-of-the-art classifiers. These are mainly catagorized into four groups: Baysian, Functional, Rule based and Tree based. BayesNet is one of the popular Baysian classifier which is used under Baysian category. Under Functional classifier we have used five popular classifiers namely: LibLINEAR, MLP, SVM, RBFNetwork and Simple Logistic. FURIA and PART are two Rule based classifier that we have used. Finally under Tree based classifiers we have used NBTree and Random Forest. Figure 3.17 shows a sample tree diagram of different classifiers used for the present work. Among the classifiers mentioned above, MLP is widely used in our experiment.

In the following section we will discuss briefly about the above mentioned classifiers.

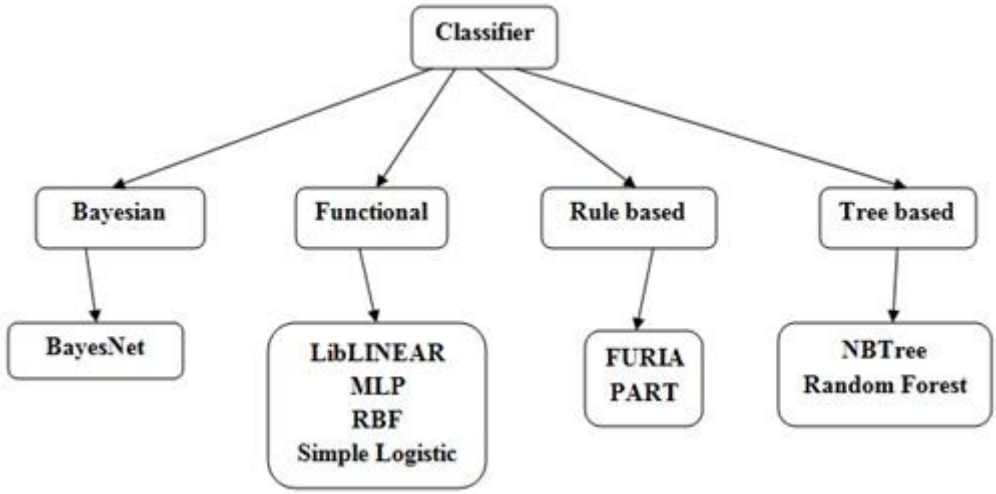


Figure 3.17 Classifier hierarchy considered for present work

3.2.1 BAYESIAN CLASSIFIER

BAYESNET

Popular Bayesian classifier uses Bayes Network learning using different search algorithms and quality parameters [91]. The Base class of this classifier provides data structures (conditional probability distributions, network structure etc.) and facilities common to Bayes Network learning algorithms like K2 and B.

3.2.2 FUNCTIONAL CLASSIFIER

LIBLINEAR

LibLINEAR is a good linear classifier based on functional model for data with large number of instances or features. It has converged faster for our dataset than other classifiers we have considered. We have used the L2-Loss Support Vector Machine (dual) as the SVM Type parameter of the LIBLINEAR both the Bias and Cost parameters are 1.0. The EPS (the tolerance of the termination criterion) is 0.01. More details are given in [92].

MULTILAYER PERCEPTRON (MLP)

Among the classifiers considered multi layer perceptron (MLP) is used most widely in our work. MLP is a type of feed forward *ANN* (Artificial Neural Network) which is nothing but a mapping from an input set to output set. It can be represented by a direct acyclic graph where direction of the signal flow is specified. Each node of a *MLP* is mimicking of an artificial neuron. In the connection between two neurons a weight or label is associated which represents the capacity or strength of the connection. The number of neurons in input layer is same as the number of feature selected for the particular pattern recognition problem. Whereas the number of output layer is same as the number of target classes. This feed forward neural network has been widely used since decades [21] [79] for pattern recognition applications. Each node (except for the input nodes) can be viewed as a neuron with a nonlinear activation function. In our work, we use the sigmoid function as the activation function:

$$\sigma_x = \frac{1}{1+\exp(-(\omega * x + v))} \qquad (3.15)$$

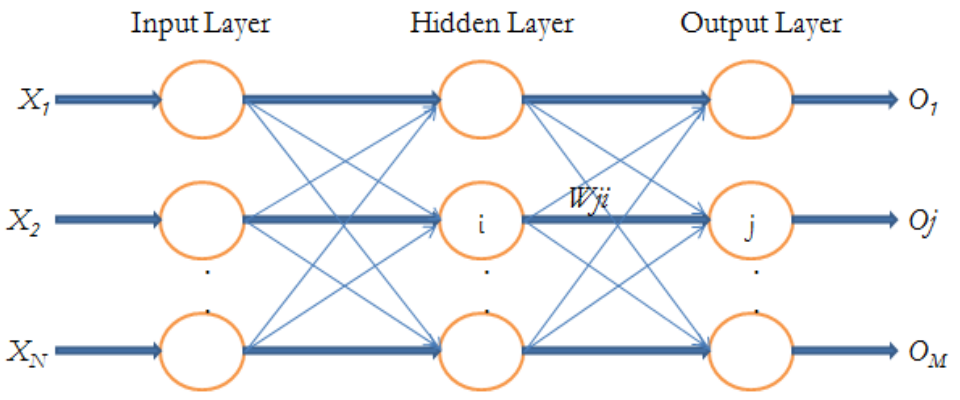


Figure 3.18 Graphical representation of multi layer MLP

Where, the weight vector w and bias vector b in each layer pair are trained by the back propagation algorithm. Figure 3.18 shows a schematic diagram of MLP with input, hidden and output layers.

BACK PROPAGATION ALGORITHM (BP)

Training of the MLP is most crucial part of learning. We use BP algorithm, which is a supervised learning technique. We find a set of weights for the perceptron which minimizes the SSEs (sum squered errors) produced with all the training patterns. To do so, a gradient descent search is done by BP algorithm on the error surface of the perceptron in the weight space. The amount of weight change ΔW_{ji}, needed to minimize the sum of squared errors is given as follows:

$$\Delta W_{ji} = \eta\, \delta_j O_j \qquad (3.16)$$

Where, η is the learning rate parameter, and $0 < \eta < 1$,

δ_j is the error gradient of the j^{th} neuron.

$$\delta_j = \begin{cases} O_j(1-O_j)(d_j - O_j) & if\ j^{th}\ neuron\ belongs\ to\ output\ layer \\ O_j(1-O_j)\sum_k \delta_k W_{kj} & if\ j^{th}\ neuron\ belongs\ to\ a\ hidden\ layer \end{cases} \qquad (3.17)$$

Considering the momentum α, *where,* $0 < \alpha < 1$, *the weight updation of BP algorithm become as follows:*

$$W_{ji}(t+1) = W_{ji}(t) + \eta\, \delta_j O_j(t) + \alpha\left(W_{ji}(t) - W_{ji}(t-1)\right) \qquad (3.18)$$

Where, t is time.

The weight updation is an iterative process. This process continues with the training patterns until certain stopping criteria are met. For present work, we optimized the parameters for MLP using learning rate of 0.3, momentum 0.2, epoch size 500, and empirically chose number of nurons in the hidden layers. We considered the stopping criteria as, when sum squared error of all the training patterns fall below 0.1.

RADIAL BASIS FUNCTION (RBF) NETWORK

In Radial basis function (RBF) networks for hidden layer processing elements the static Gaussian function has been used as the nonlinearity. The function works in a small centered region of the input space. The implementation of the network depends on the

centers of the Gaussian functions [93] [94]. The main functionality depends on how the Gaussian centers are derived and they act as weights of input to hidden layer. The widths of the Gaussians are calculated depending on the centers of their neighbors. The faster convergence criterion is one of the advantages of this network. This is because it only updates weights from hidden to output layer. We optimize the performance of RBFNetwork considering parameters like: number of cluster for K-means as 2, the random seed to pass on K-means as 1, minimum standard deviation as 0.1 and the ridge value for logistic regression as $1.0E^{-8}$.

SIMPLE LOGISTIC

It is a classifier for building linear logistic regression model [95]. Here LogitBoost is used with simple regression functions as base learner for fitting the logistic model. The optimal number of LogitBoost iterations to perform is cross-validated here, which helps for the selection of automatic attribute.

3.2.3 RULE BASED CLASSIFIER

FURIA

Fuzzy Unordered Rule Induction Algorithm (FURIA) is a fuzzy-rule-based classifier, used to obtain fuzzy rules. FURIA has recently been developed as an extension of the well-known RIPPER algorithm. Instead of conventional rules and rule lists it learns fuzzy rules and unordered rule sets. Furthermore it uses an efficient rule stretching scheme to deal with uncovered examples [96]. All the parameters for FURIA classifier of Weka tool are set to its default values for this work like the MINNO (minimum total weight of the instances in a rule) has been set to 2.0.

PART

It is a class for generating a PART decision list. It uses separate-and-conquer method. Then builds a partial C4.5 decision tree in each iteration and makes the best leaf into a rule [79].

3.2.4 TREE CLASSIFIER

NBTREE

This is a tree based classifier [79]. It contains class for generating a decision tree with naive Bayes classifiers at the leaves.

RANDOM FOREST

Random forest (RF) is an ensemble classifier. It operates by constructing a group of decision trees at training time and outputting the class that is the mode of the classes output by individual trees [97]. Considering a training set $X = \{x_1, x_2, \dots\dots, x_n\}$ with corresponding responses $Y = \{y_1, y_2, \dots\dots, y_n\}$, we continuously select samples from the training set and fit trees to the samples, using bagging approach. In general, for $b = 1, \dots\dots\dots, B$, we sample (with replacement) n training samples from X, Y, we call these X_b, Y_b. We then train a decision or regression tree f_b on X_b, Y_b. After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x' or by taking the majority Voting in the case of decision trees:

$$f^{\wedge} = \frac{1}{B}\sum_{b=1}^{B} f_b^{\wedge}(x') \qquad (3.19)$$

3.3 EVALUATION PROTOCOL

During experimentation k-fold cross validation is followed. All the sample images are initially divided into k different subsets. In our experimentation the value of k was chosen empirically as 5. The identification accuracy is calculated using the following formula (Eq. 3.20):

$$\mathrm{Re}cognition\_accuracy = \frac{\#correctly\_classified\_pages}{\#total\_pages} x100\% \qquad (3.20)$$

For evaluation, we have investigated total three types of script identification scenarios: (i) bi-script (ii) tri-script and (iii) multi-script (in our case it is 11). Most multi-script documents in India are in general bi-script in nature, so this case is handled first. Presence of tri-script documents in real life encourages us to test this case also. Finally, being encouraged from the bi-script and tri-script results we have tested 11-script scenario. Here we explore the possibility of recognizing a script of a page without any prior knowledge. So, this is a kind of blind script recognizer, where the training set contains sample pages from all the classes. In the following section we explore each of these cases separately.

To evaluate the performance of our method, we have used different performance metrics in our work. These are: average accuracy rate (AAR), model building time (MBT), true positive rate (TP rate), false positive rate (FP rate), false negative rate (FN rate) precision, recall, f-measure and ROC area. A brief description of these metrics is discussed in following section.

AAR: Average accuracy rate (%) is the actual identification rate. It is measured by the equation (2).

MBT: Measured in Sec. It is the total time to train the system.

TP Rate: True positive rate is the proportion of test samples among all which were classified correctly to a target class at which they should belong.

FP Rate: False positive rate is the proportion of test samples which belongs to a particular class but misclassified to a different class.

FN Rate: False negative rate provides the total misclassification rate, i.e. the proportion of samples among all which were misclassified to other classes.

Precision: It is defined as the proportion of test samples which truly have classified to a particular class among all those which were classified to that class. So, *Precision = TP Number / (TP Number + FP Number).*

Recall: Recall is defined as follows: *Recall = TP Number / (TP Number + FN Number).* Here *FN Number* is the false negative number.

F-Measure: It is a combined measure of precision and recall. It is defined as: *F-Measure = 2 * Precision * Recall / (Precision + Recall).*

3.4 CONCLUSION

Script identification is a well studied problem in literature since the last decade but still it is far from the complete solution. To propose a solution for the said problem, in this chapter, we studied different features and classifiers. These features are broadly catagorized into two types: (i) script dependent feature and (ii) script independent feature (summerized in Table 3.1 and Table 3.2). Script dependent features are mainly: structure based feature, topological feature and stroke based feature. These features first analyze the shape and visual appearance of different Indic scripts and then compute certain values which are script specific. In our work, different script dependent features are considered. They are: count of number of small components, circularness, rectangularness of an component, shape of convex hull, chain code histogram, presence/absence of topological property like 'matra' or 'shirorekha', presence of directional strokes of different orientations. Under script independent feature, we considered texture analysis and image transform fusion. Texture is an important tool to differentiate different scripts. So, we computed different texture features like: gray level co-occurance matrix, Gabor filter bank, spatial energy, wavelet energy, the radon transform. Further, we propose the fusion of two texture feature, i.e. fusion of wavelet and radon transform. The effectiveness of the fusion technique is supported by the experimental result as the performance of wavelet has been optimized while we were combining it with radon transform.

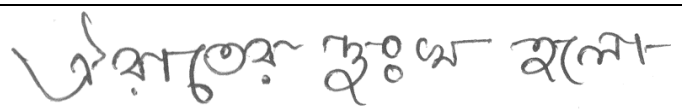
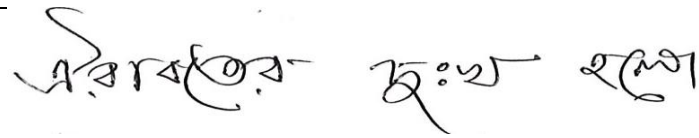
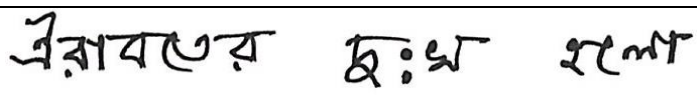
Different state-of-the-art classifiers are considered in our work. Among the classifiers MLP was found to be most efficient. We have compared the performance of MLP with other classifiers too. Among them, random forest and simple logistic performs comparatively well enough. Finally, the performance of our methods is measured using different well known evaluation metrices. These are: average accuracy rate (AAR), model building time (MBT), true positive rate (TP rate), false positive rate (FP rate), false negative rate (FN rate) precision, and recall, f-measure and ROC area. In Chapter 4 and Chapter 5, the outcome of printed and handwritten script identification is discussed.

CHAPTER FOUR

PRINTED SCRIPT IDENTIFICATION

In our study, we have categorized all script identification techniques into two major divisions based on the nature of input document image, i.e. whether they are machine printed or handwritten. These two techniques are: printed script identification (PSI) and handwritten script identification (HSI). PSI is the technique which is applied only to printed documents to know about document's script type. On the other hand, HSI is a technique concerning only about handwritten text images. In general printed texts are more uniform compared to handwritten one. Here uniformity means: regular shape of the characters/components, uniform spacing between characters in a word, words in a line and lines in a paragraph. This uniformity occurs in printed text due to writer independence of printed documents as they are machine generated.

Table 4.1 A sample Bangla printed text and the same text written by three different writers

Printed Bangla Text	ঐরাবতের দুঃখ হলো
Writer 1	
Writer 2	
Writer 3	

In Table 4.1, we have shown a sample Bangla printed text and the same text written by three different writers. It is observable that, the characters are words are 100% uniformly distributed in printed text. The same text while written by three different writers, i.e. Write 1, Writer 2 and Writer 3, significantly varies in terms of few important parameters: character spacing, word spacing, height and width of the characters, overall height and width of the words, length of the lines and connected component length. This uniformity of texts in printed documents makes the document processing task much easier. That is why, so far most of attempts on Indic or non-Indic scripts were made mainly on printed documents. In following section we discuss about the state-of-the-art on PSI techniques.

4.1 PRINTED SCRIPT IDENTIFICATION - LITERATURE REVIEW

Ghosh *et al.* [8] presented a review on different script identification techniques. Few works are reported in literature on printed Indic script identification. Sometimes non Indic scripts are also considered in the database along with Indic scripts. Among those, Spitz [46] in his work identified Latin, Han, Chinese, Japanese, and Korean scripts by using features like upward concavity distribution, optical character density etc. He carried out his work at document level. Lam *et al.* [48] identified some non-Indic scripts using horizontal projection profile, height distribution, presence of circles, ellipse, and presence of vertical stroke features. Hochberg *et al.* [47] identified six scripts namely Arabic, Armenian, Devanagari, Chinese, Cyrillic, Burmese using some textual symbol based features. Zhou *et al.* [30] identified Bangla and English scripts using connected component based features from both printed and handwritten document. Patil and Subbareddy [98] proposed a tri script identification technique on English, Kannada and Hindi using neural network based classification technique. They performed their work at word level. Elgammal and Ismail [99] proposed a block level and line level script identification technique from Arabic and English scripts using Horizontal projection profiles and run-length histograms analysis. Dhandra *et al.* [100] proposed a word level script identification technique from Kannada, Hindi, English

and Urdu using morphological analysis. Chaudhuri and Pal [101] proposed a line based script identification techniques from Roman, Bangle and Devanagari scripts. Tan *et al.* [49] proposed a mixed script identification techniques considering Chinese, Latin and Tamil using upward concavity based features. In another work Padma and Vijaya [102] proposed a work using wavelet transform based feature considering seven Indic and non-Indic scripts namely English, Chinese, Greek, Cyrillic, Hebrew, Hindi, and Japanese. Using Multi Channel Log Gabor filter based features Joshi *et al.* [103] proposed a block level script identification technique from English, Hindi, Telugu, Malayalam, Gujarati, Kannada, Gurumukhi, Oriya, Tamil and Urdu scripts. Dhanya *et al.* [104] proposed a word level script identification technique from Roman and Tamil scripts using Multi Channel Gabor Filters and Discrete Cosine Transform (DCT) based feature. Pati *et al.* [105] proposed a word level script identification technique from eleven Indic scripts using two pertinent features: DCT and Gabor filter.

In this chapter, we discuss about the experimentation carried out on printed dataset. Two separate printed dataset have been used: page-level and word-level printed dataset from eleven official Indic scripts. The outcome is discussed in the following section.

4.2 PROPOSED WORK ON PSI

In Chapter 3, we have discussed about different techniques and methods. We performed the experimentation on printed datasets to analyse the performance of printed script identification. Two different datasets, i.e. page-level and word-level from eleven scripts are developed. In the following section, we discuss about the proposed work on PSI.

4.2.1 PAGE-LEVEL SCRIPT IDENTIFICATION FROM ELEVEN OFFICIAL SCRIPTS

It is indeed clear from the above survey at Section 4.1, that very few works are attempted so far considering page-level documents, especially on Indic scripts. To bridge this gap, the present work is an attempt to identify any one of the eleven official

Indic scripts, and also the performance of different well known classifiers are analysed for the same [79]. The eleven scripts considered are: Bangla, Devanagari, Gujarati, Gurumukhi, Kannada, Malayalam, Oriya, Roman, Tamil, Telugu and Urdu. Figure 4.1 shows a block diagram of the proposed model. Page-level document images are supplied as input; they are pre-processed, i.e. converted into binary level. Features are extracted from those binary level images and then the performance of different state-of-the-art classifiers are compared to find the best performer.

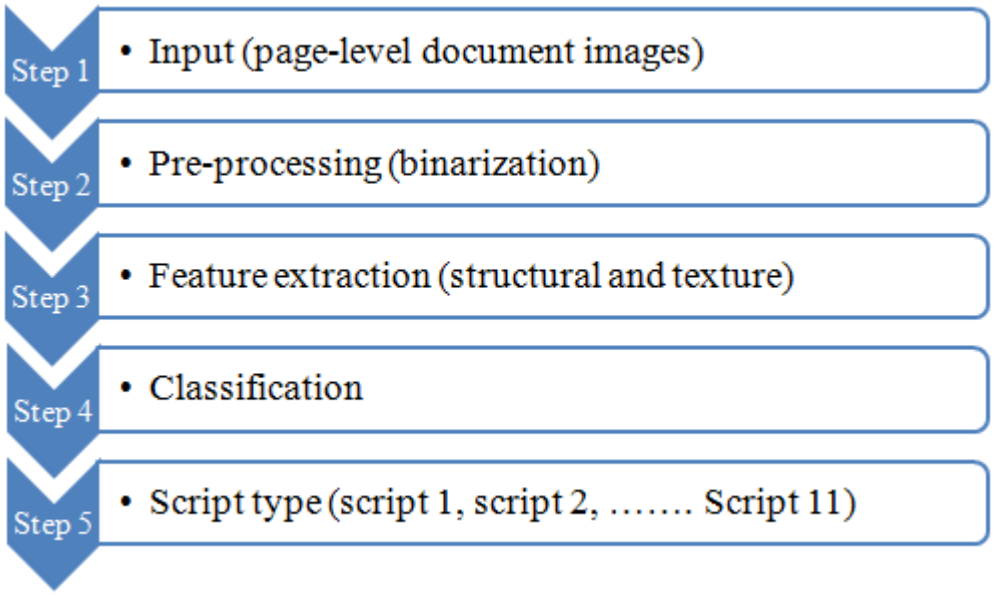


Figure 4.1 The general block diagram of the proposed page-level PSI system

DATA COLLECTION AND PRE-PROCESSING

Availability of standard database is one of the most important issues for any pattern recognition research work. No printed page-level dataset is available till date for all official Indic scripts. Real life printed script data are collected from different sources like book pages, articles etc. A total of 498 printed document pages are collected for experimentation. The script wise dataset distribution is provided in the following Table 4.2. Figure 4.2 shows sample script images from our database. Initially the images are in gray tone and digitized at 300 dpi using a flatbed HP scanner M1136 MFP. A two stage

based binarization technique is used to convert the images into binary images [21]. At first stage pre-binarization is done using a local window based algorithm in order to get an idea of different Region of Interest or ROI. Then Run Length Smoothing Approach (RLSA) is applied on the pre-binarized image. This will overcome the limitations of the local binarization method used. The stray/hollow regions created due to fixed window size are converted into a single component. Finally, using component labelling, each component is selected and mapped in the original gray image to get respective zones of the original image. The final binary image is obtained by applying a histogram based global binarization algorithm on these regions/components of the original image. After pre-processing feature extraction process is carried out to construct the feature vector. Major features considered for the present work are discussed in the following section.

Table 4.2 Script wise distribution of page-level printed dataset

Script Name	**# of Samples**
Bangla	60
Devanagari	60
Gujarati	58
Gurumukhi	31
Kannada	60
Malayalam	29
Oriya	20
Roman	60
Tamil	30
Telugu	60
Urdu	30
Total	498

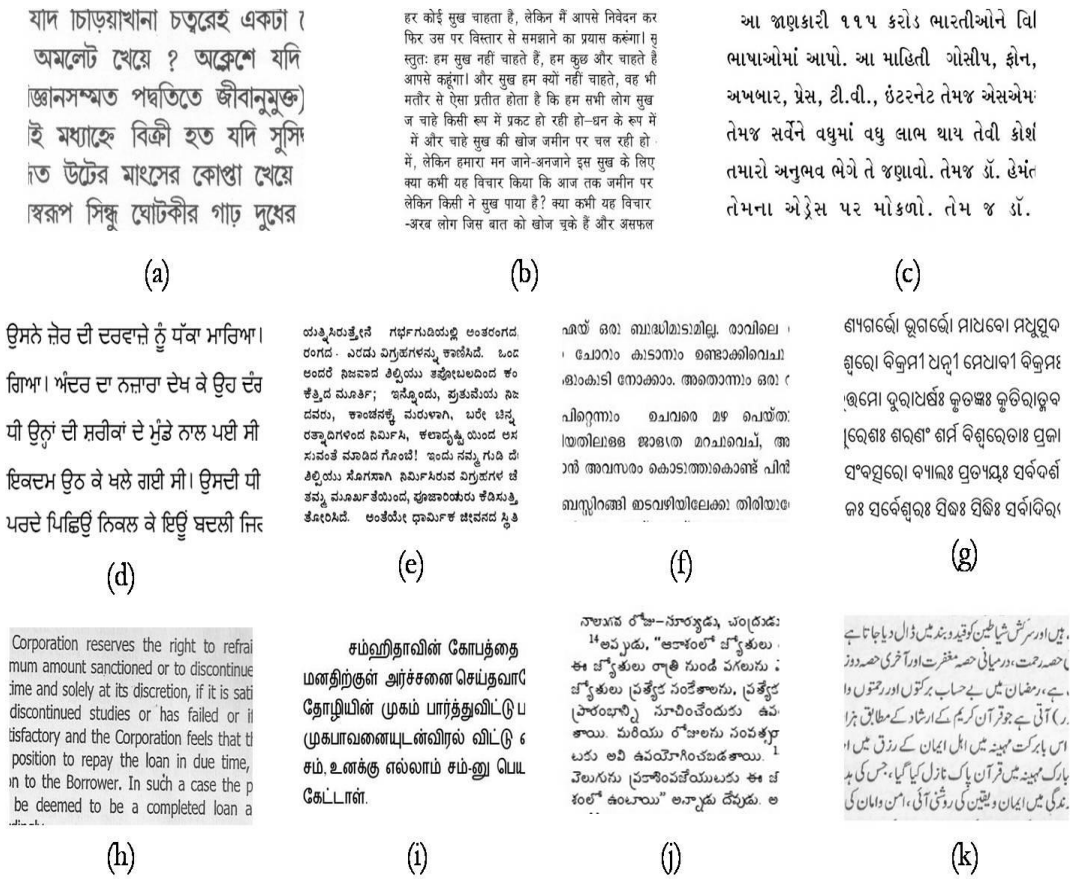


Figure 4.2 Sample from our dataset of (a) Bangla, (b) Devanagari, (c) Gujarati (d) Gurumukhi (e) Kannada (f) Malayalam (g) Oriya (h) Roman (i) Tamil (j) Telugu and (k) Urdu script documents

FEATURE EXTRACTION AND DESIGN OF FEATURE SET

During feature extraction, first, visual observations are made on Indic scripts to study the nature of different graphemes of different scripts. The main features considered are structural along with few texture based feature. The features considered for this work are as follows:

- $\boldsymbol{FS_{SVA}}$ **–** Structural and Visual Appearance based feature set. Overall feature dimension is 42 [See Chapter 3, Table 3.1]
- $\boldsymbol{FS_{DSI}}$ **–** Directional Stroke Identification based feature. Here we only consider two basic morphological operations: erosion and dilation. Overall feature dimension is 12 [See Chapter 3, Table 3.1].

- FS_{GABOR}- Computation of texture based feature using Gabor filter with varying frequency and orientation. Overall feature dimension is 08 [See Chapter 3, Table 3.2].

Final feature set for present page-level script identification problem:

$$FS_{SVA \cup DSI \cup GABOR} = FS_{SVA} \cup FS_{DSI} \cup FS_{GABOR}$$
$$= \textbf{62 dimensions}$$

Details about computation of these features are discussed in Chapter 3. Following figures show few snapshot of the sample outcome. In Figure 3.3, computation of circularity feature on Oriya script component has been shown. The blue circle is minimum encapsulating circle on the outer contour and the green circle is the best fitted one. In Figure 3.4, the computation of rectangularity or bounding box feature on the same script, i.e. Oriya has been shown. In Figure 3.5 the computation of convex hull on Urdu script components has been shown.

CLASSIFIER AND EVALUATION PROTOCOL

To evaluate the features, we have considered state-of-the-art classifiers and analyze their performances to find the best classifier with respect to average accuracy rate (AAR) and model building time (MBT) on the present dataset. The classifiers considered are: BayesNet, LibLINEAR, MLP, RBFNetwork, Simple Logistic, PART, and Random Forest. The detail about these classifiers is discussed in Chapter 3. During experimentation, k-fold cross validation is followed. In our experimentation the value of k was chosen empirically as 5.

EXPERIMENTAL RESULTS

Table 4.3 provides comparison of different classifiers based on two parameters AAR and MBT (defined in Section 3.3). It has been found that, Random Forest classifier which is a tree based classifier performs best with 98.99% average accuracy followed by LibLINEAR and MLP with a nominal difference of 0.80% and 0.99% respectively. The fastest model building time is reported by BayesNet classifier. Table 4.4 shows the confusion matrix of the Random Forest classifier.

Table 4.3 Comparison of result for different classifiers using feature set $\boldsymbol{FS_{SVA \cup DSI \cup GABOR}}$

Type	Classifier	AAR (%)	MBT (s)
Bayesian	BayesNet	96.38	**0.28**
Functional	LibLINEAR	98.19	1.81
	MLP	98.00	120.67
	RBFNetwork	94.57	15.49
	Simple Logistic	97.38	15.77
Rule Based	PART	93.37	0.66
Tree Based	Random Forest	**98.99**	1.57

Table 4.4 Confusion matrix for Random Forest classifier (top performer in Table 4.3), Abbreviation: BEN: Bangla, DEV: Devanagari, GUJ: Gujarati, GUR: Gurumukhi, KAN: Kannada, MAL: Malayalam, ORY: Oriya, ROM: Roman, TAM: Tamil, TEL: Telugu and URD: Urdu

Classified As	BEN	DEV	GUJ	GUR	KAN	MAL	ORY	ROM	TAM	TEL	URD
BEN	60	0	0	0	0	0	0	0	0	0	0
DEV	0	60	0	0	0	0	0	0	0	0	0
GUJ	0	0	57	0	0	0	0	1	0	0	0
GUR	0	0	0	29	0	0	0	0	2	0	0
KAN	0	0	0	0	60	0	0	0	1	0	0
MAL	0	0	0	0	0	29	0	0	0	0	0
ORY	1	0	0	0	0	0	19	0	0	0	0
ROM	0	0	0	0	0	0	0	60	0	0	0
TAM	0	0	1	0	0	0	0	0	29	0	0
TEL	0	0	0	0	0	0	0	0	0	60	0
URD	0	0	0	0	0	0	0	0	0	0	30

The present work discusses the issue of page-level printed script identification from eleven official Indic scripts. Mainly structural features are used as sole pertinent feature to distinguish different Indic scripts. Page-level script identification problem can be treated as a blind script identification problem where any type of documents written by any script is supplied to the system and the output script type is produced. Impressive

average identification accuracy of 98.99% is obtained by Random Forest classifier. We got 100% individual script identification accuracy for Bangla, Devanagari, Kannada, Malayalam, Roman, Telugu and Urdu scripts. The highest misclassification occurs for Gurumukhi script. About 6.45% Gurumukhi pages are misclassified as Tamil script. We also receive comparable performance from other two popular classifiers namely LibLINEAR and MLP. For model building time, BayesNet works fastest among all.

4.2.2 WORD-LEVEL SCRIPT IDENTIFICATION FROM ELEVEN OFFICIAL INDIC SCRIPTS

In many occasions multi-script documents occur at word-level, i.e. the document contains words written by different scripts. Figure 4.3 shows a sample word-level multi-script printed document image. To handle such type of document in OCR we need to identify the script type at word-level. So, word-level script identification is a real problem in our country.

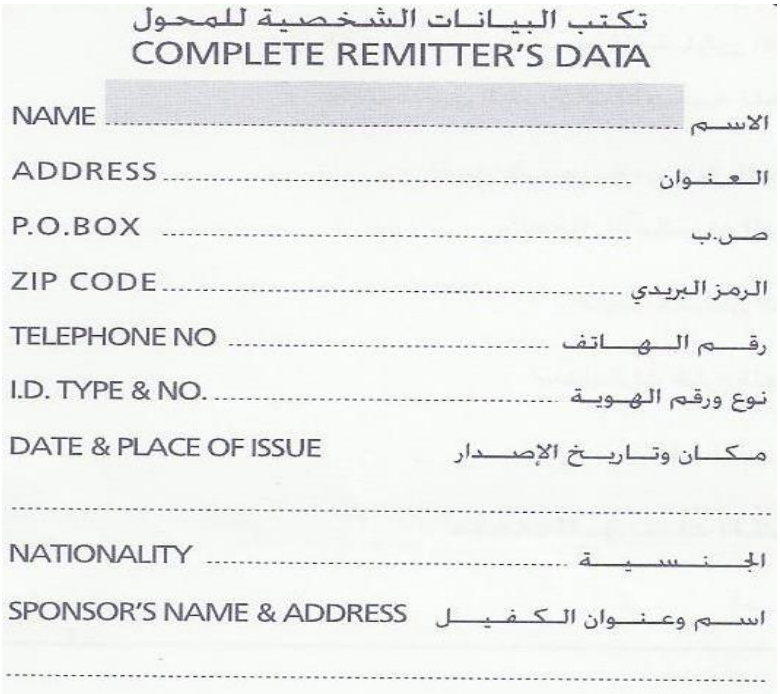


Figure 4.3 Word-level multi-script printed document

In this work, we consider word-level images from thirteen different languages, which belong to eleven official scripts. Our study is not an exception; we start with pre-processing, and then extract features for script identification purpose. In this experiment, we study three different features:

- $\boldsymbol{FS_{SE}}$ **–** Spatial Energy based feature set. The overall feature dimension is 04 in our experiment [See Chapter 3, Table 3.2].
- $\boldsymbol{FS_{WE}\#1}$ **–** Feature based on Wavelet Energy, overall feature dimension under this category is 15 [See Chapter 3, Table 3.2].
- $\boldsymbol{FS_{WRT}\ \#2}$**-** Consider a fusion based feature, Wavelet Radon Transform. The feature dimension in this category is 65 [See Chapter 3, Table 3.2].

Final feature set *present word-level script identification problem:*

$\boldsymbol{FS_{SE \cup WE\#1 \cup WRT\#2}} = \boldsymbol{FS_{SE}} \cup \boldsymbol{FS_{WE}\#1} \cup \boldsymbol{FS_{WRT}\#2}$ **= 84 dimensions**

During experimental evaluation we have also tested the performance of their possible combination for suitable feature selection.

Three popularly used classifiers are considered:

- Multilayer Perceptron (MLP),
- Fuzzy Unordered Rule Induction Algorithm (FURIA) and
- Random Forest (RF).

We have discussed about these features and classifiers in Chapter 3. In our study, from 13 different languages i.e. 11 different scripts, we have considered two different test categories:

- Bi-script and
- Tri-script

In general, there are C_2^{13} and C_3^{13} possible combinations of bi-script and tri-script categories. But, considering the nature of the multi-script documents, these straightforward combinations may not hold true in the real-world (e.g. postal

documents and application forms). We have also observed that, Devanagari and Roman exist in most of the documents. This means that any bi-script or tri-script document in general contains either or both Devanagari and/or Roman in addition to their local script. Considering such a context, we have formed two different script sub-categories for bi-script: case 1 and case 2. Bi-script case 1 contains twelve script combinations with Devanagari common. Bi-script case 2 contains Roman as common script, for all remaining 12 scripts. For tri-script category, we have a total number of 11 combinations where both Devanagari and Roman are kept as common with other local scripts. Also, note that, we have divided the database into training and test sets as 2:1 ratio.

Again, our experimental test framework can be summarized as follows. As said before, in this work, our idea is not only to check what features but also to check what classifiers can consistently provide optimal performance. Therefore, we have seven different tests in accordance with the use of individual features and their possible combinations: FS_{SE}, $FS_{WE}\#1$, $FS_{WRT}\#2$, $FS_{SE \cup WE\#1}$, $FS_{SE \cup WRT\#2}$, $FS_{WE\#1 \cup WRT\#2}$ and $FS_{SE \cup WE\#1 \cup WRT\#2}$. These are tested by using three different classifiers: MLP, FURIA and RF.

Table 4.5 Bi-Script case 1 (Devanagari common): average performance scores (in %) for different feature combinations

Feature type (dimension)	**Classifier**		
	MLP	**FURIA**	**RF**
FS_{SE}(4)	81.93	80.30	86.28
$FS_{WE}\#1$ (15)	91.10	89.30	92.35
$FS_{WRT}\#2$ (65)	96.93	95.31	95.84
$FS_{SE \cup WE\#1}$ (19)	94.83	93	94.98
$FS_{SE \cup WRT\#2}$ (69)	97.86	97.09	97.03
$FS_{WE\#1 \cup WRT\#2}$ (80)	97.80	96.36	96.48
$FS_{SE \cup WE\#1 \cup WRT\#2}$ (84)	98.38	97.42	97.35

Table 4.6 Bi-Script case 1 (Devanagari common): average performance (in %) scores for 12 different combinations for $FS_{SE \cup WE\#1 \cup WRT\#2}$

Bi-script combination case 1	Classifier		
	MLP	**FURIA**	**RF**
DEV-BEN	94.70	95.00	94.20
DEV-DOG	99.70	99.00	98.30
DEV-GUJ	99.40	98.70	98.30
DEV-GUR	90.90	89.50	91.60
DEV-KAN	99.20	97.90	97.90
DEV-KAS	99.90	99.30	99.00
DEV-MAL	99.70	98.80	98.30
DEV-ORY	99.90	99.50	99.60
DEV-ROM	99.30	97.60	97.50
DEV-TAM	98.40	95.90	96.10
DEV-TEL	99.90	98.80	98.60
DEV-URD	99.60	99.00	98.80
Average	**98.38**	97.42	97.35

In Table 4.5, average performance scores for different feature combinations are provided. The results are provided for bi-script case 1 (Devanagari common). One of the scores in this table is computed by making 12 numbers of runs as shown in Table 4.6. Altogether, we have $C_2^{13} \times 3 = 36$ runs, for just a single feature type. In Table 4.6, MLP provides the best performance (i.e., 98.38%) when all features are combined, which, however, does not provide a significant difference other classifiers. In a similar fashion, bi-script case 2 has been tested, where Roman is common. Results are provided in Table 4.7 and Table 4.8 for bi-script case 2 (Roman common). In the latter case (i.e., Table 4.8), the observed highest accuracy is 99.24%. Like before, MLP provides better results when all features are combined -- even for tri-script combinations. It is worthy to mention here that we have used WEKA [78] a popular open source machine learning library to do our experemnt. All the parameters of the classifiers are remaining default during the above experiment. In Table 4.9, average

performance scores are provided for tri-script combinations, where the highest identification rate is 98.19%. In this test, we have submitted $11\ (tri-script\ combinations) \times 3(classifiers) = 36$ runs, for just a single feature type. Again, for a comparison (between the classifiers) purpose, their average scores are provided in Table 4.10, where we found MLP > RF > FURIA, even though there exists no significant difference between them. In this comparison table, one can also note that higher the script combination, lower the performance of classifiers -- which is obvious because it increases number of classes to be classified.

Table 4.7 Bi-Script case 2 (Roman common): average performance scores (in %) for different feature combinations

Feature type (dimension)	**Classifier**		
	MLP	**FURIA**	**RF**
FS_{SE}(4)	80.94	79.93	81.90
$FS_{WE}\#1$ (15)	92.56	91.20	93.50
$FS_{WRT}\ \#2$ (65)	98.01	96.67	96.66
$FS_{SE \cup WE\#1}$ (19)	95.06	94.07	95.60
$FS_{SE \cup\ WRT\#2}$ (69)	98.96	97.68	97.68
$FS_{WE\#1 \cup WRT\#2}$ (80)	99.07	97.58	97.62
$FS_{SE \cup WE\#1 \cup WRT\#2}$ (84)	**99.24**	97.91	98.11

Table 4.8 Bi-Script case 2 (Roman common): average performance (in %) scores for 12 different combinations for $FS_{SE \cup WE\#1 \cup WRT\#2}$

Bi-script combination case 2	**Classifier**		
	MLP	**FURIA**	**RF**
ROM-BEN	99.00	96.60	97.50
ROM-DEV	99.30	97.60	97.50
ROM-DOG	99.30	97.80	98.00
ROM-GUJ	98.20	94.90	95.40
ROM-GUR	99.30	99.20	98.80
ROM-KAN	99.30	99.00	98.40
ROM-KAS	99.50	99.20	99.40

ROM-MAL	99.10	97.40	98.60
ROM-ORY	99.70	98.90	99.20
ROM-TAM	99.30	97.40	97.10
ROM-TEL	99.50	98.60	99.00
ROM-URD	99.40	98.30	98.40
Average	**99.24**	97.91	98.11

Table 4.9 Tri-Script case (Devanagari & Roman common): average performance (in %) scores for 12 different combinations for $FS_{SE \cup WE\#1 \cup WRT\#2}$

Tri-script combination	**Classifier**		
	MLP	**FURIA**	**RF**
DEV-ROM-BEN	96.20	93.40	90.70
DEV-ROM-DOG	99.20	96.90	98.00
DEV-ROM--GUJ	97.80	95.60	94.00
DEV-ROM-GUR	94.00	89.00	84.70
DEV-ROM-KAN	98.90	96.70	96.50
DEV-ROM-KAS	99.60	97.00	96.70
DEV-ROM-MAL	98.70	96.00	95.50
DEV-ROM-ORY	99.50	97.60	98.00
DEV-ROM-TAM	97.90	94.40	94.00
DEV-ROM-TEL	99.30	97.70	97.90
DEV-ROM-URD	99.00	96.70	96.00
Average	**98.19**	95.55	94.73

Table 4.10 Comparison of classifiers for features $FS_{SE \cup WE\#1 \cup WRT\#2}$, Average scores are reported

Tri-script combination	**Classifier**		
	MLP	**FURIA**	**RF**
Bi-script case 1 (12)	98.38	97.42	97.35
Bi-script case 2 (12)	99.24	97.91	98.11
Tri-script (11)	98.19	95.55	94.73

Prior to this study, Pati et al. [105] proposed word-level script identification by using 11 Indic languages, where Gabor and DCT based features are taken. They have compared their performances using three different classifiers namely neural network (NN), linear discriminant analysis (LDA) and support vector machine (SVM). Their performance scores are approximately 98% from both bi-script and tri-script combinations. In contrast, our work is composed of all 13 official languages under 11 different scripts, with 39k dataset. Three types of features are used: spatial energy, wavelet energy and radon transform. Performances of three different classifiers namely MLP, FURIA, and RF have been compared, and MLP is found to be better performer. In our comprehensive tests, we have script identification rate of 98.38% (keeping Devanagari common) and 99.24% (keeping Roman common) for bi-script combination, and identification rate of 98.19% for tri-script combination. For better understanding a comparative chart is shown by Table 4.11.

The graphical representation of the performance comparison of different classifiers is illustrated in Figure 4.4.

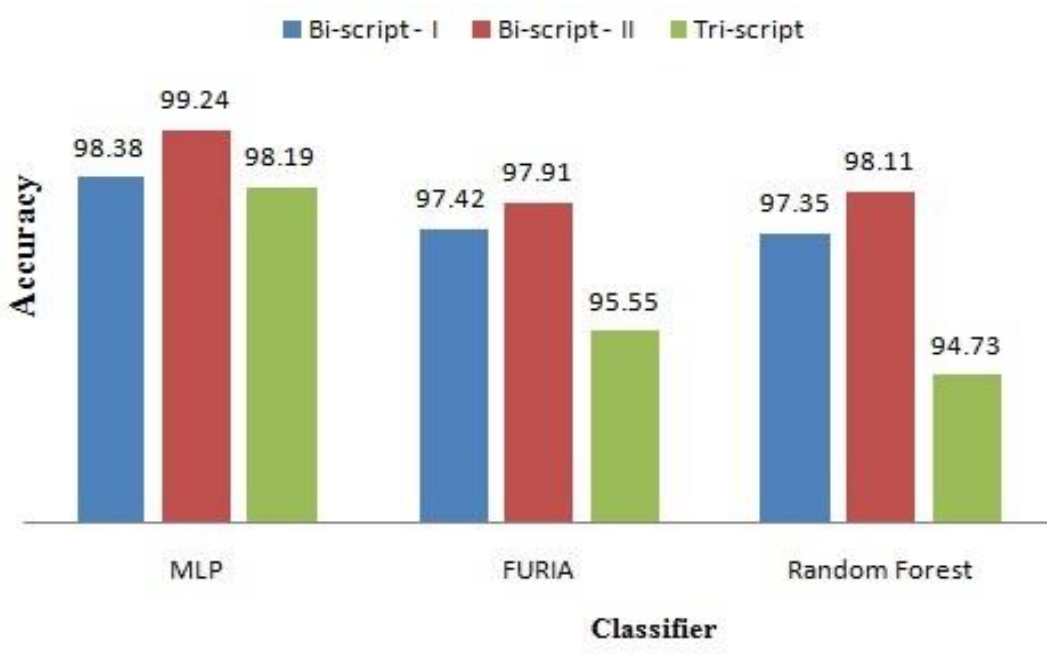


Figure 4.4 Performance comparison of different classifiers

Table 4.11 Analogy with the previous work

Method	Dataset	Identification rate (%)
Pati et al. [105]	11 languages	98.00 (bi-script) 98.00 (tri-script)
Proposed	13 languages	99.24 (bi-script case 1) 98.38 (bi-script case 2) 98.19 (tri-script)

Word-level printed script identification from thirteen official languages which belongs to eleven different scripts is reported in the work. Performances are evaluated using state-of-the-art features and classifiers. An exhaustive feature selection is also done where $FS_{SE} \cup FS_{WE}\#1 \cup FS_{WRT}\#2$ performs best. Bi-script and tri-script performance have been studied and MLP is found to be the best performer among all. The reported results can be considered as benchmark one on the present framework on present dataset.

Error Analysis:

By error analysis we try to understand the error pattern and possible cause of misclassification. Though printed script are very much uniform in nature as compared to handwritten one, still there are many misclassification instances we have found in our experiments. The most common cause of such misclassification is the visual and structural similarities that are common among many Indic scripts. As we know, 'matra' is common for Bangla and Devanagari, the characters of south Indian scripts looks similar to Oriya script, there is clear visual similariry between Devanagari and Gurumukhi scripts. The Table 4.4 shows there are misclassification among Gujarati and Roman, Gurumukhi and Tamil, Kannada and Tamil, Oriya and Bangla. In Table 4.6, which shows the word level script identification for bi-script category 1 (keeping Devanagari common with others), we found that most misclassification happened between Devanagari-Gurumukhi (9.10%) and Devanagari-Bangla (5.30%). The result pattern is similar among all the three classifiers considered for this experiment

i.e. for MLP, FURIA and RF classifiers. It is already said that, there are very much similarity btween Devanagari and Gurumukhi characters which is the sole reason behind the misclassification. While considering Bangla, we found that, although Bangla and Devanagari characters are not quite similar but there is a common feature btween these two scripts, which is the presence of "matra', that is the possible reason for misclassification. On the other hand, in Table 4.8, which shows the word level script identification for bi-script category 2 (keeping Roman common with others), we found that there are very less misclassification as compared to bi-script category 1. We observe that, this is because Roman script is visually and structurally very much different compared to other Indic scripts. But still there are situations where many Indic scripts are having some horizontal or vertical strokes similar to Roman. That's why we found few misclassifications between Roman and Gujarati (1.80%). In other cases the misclassification is very less. Besides the above observation of structural similariry, sometimes the reason for misclassification is also because of the presence of noise in the data, unwanted skewness due to improper scanning, low resolution etc. So, special care needs to be taken to handle all such issues. Figure 4.5 shows some sample images depicting the possible cause of misclassification among various Indiac scripts.

Figure 4.5 Sample images which show the possible cause of misclassification (a-c) Devanagari, Gurumukhi and Bangla scripts bearing similar 'matra' like component, (d-e) Gujarati and Roman words contain similar vertical strokes in many characters. (f-h) sample noisy images, (f) blur Devanagari word, (g) noisy dot like components in Malayalam word, (h) shows sample Tamil word where few characters are broken

4.3 CONCLUSION

No doubt, script identification has been taken as the well-studied problem since several years but, we do not have fairly large database for research, and therefore, one cannot make fair comparison. In this chapter we have proposed two approaches for script identification i.e. page-level script identification from ten official Indic scripts and word-level script identification from thirteen official languages. At page-level work, we have proposed different structural features like: small component, circularity, rectangularity, convexity and chain code. Not only features, we have also compared different state-of-the-art classifiers to compare the performance. The outcome is impressive, as we have achieved 98.9% overall accuracy using simple logistic classifiers.

In another work, we have considered thirteen official languages which consist of eleven different scripts. Different textual features namely: spatial energy, wavelet energy and the radon transform are applied to compute the feature vector. To evaluate the performance three different frameworks namely: bi-script case 1 (Devanagari common), bi-script case 2 (Roman common) and tri-script (both Devanagari and Roman common) are considered. The overall identification rate we have received as: 99.24% for bi-script case 1, 98.38% for bi-script case 2 and 98.19% for tri-script, using MLP classifier. We are in the process to investigate those few misclassification samples (i.e., from Kashmiri-Urdu, Devanagari-Gurumukhi combinations) so that we can come up with new features to achieve the expected performance. Integrating classifiers is also in our plan to further improve accuracy of our system. In both the work mentioned, due to unavailability of standard dataset, we have prepared our own dataset of 498 images at page-level and 39K images at word-level. These datasets will be available publicly for research purpose.

CHAPTER FIVE

HANDWRITTEN SCRIPT IDENTIFICATION

Identification of scripts from handwritten documents is more challanging compared to printed one. The main factors behind this are: different writing styles from people of diversified cultures across the globe, asymmetric nature of handwritten characters compared to symmetric printed characters, presence of skew at word, line or document-level, presence of dissimilar characters within a single word from a single writer, different spacing between different words, lines and characters. In Chapter 4, Figure 4.1 shows such differences between printed and handwritten texts. Script identification from handwritten documents is still an open challenge due to these stated reasons. The feature/combination of features which produce promising results on printed document may drop significantly when applied to handwritten documents. In this chapter, we proposed solutions for handwritten script identification (HSI) problems on different Indic scripts.

5.1 PROPOSED WORK ON HSI

Script identification can be done at page/block/line/word level. In literature, there exists such diversified distribution of script identification work. Initially we started with page-level script identification where segmentation is not necessary, i.e. the whole document is supplied as input. Few works are reported in literature on page-level but they were mainly restricted to non-Indic scripts. Till date, no work has been reported on page-level script identification from all the eleven official Indic scripts. So, this problem has been addressed first.

5.1.1 PAGE-LEVEL SCRIPT IDENTIFICATION FROM ELEVEN OFFICIAL INDIC SCRIPTS

In one of our earlier works [7], we proposed a page-level script identification scheme considering six Indic scripts namely Bangla, Devanagari, Roman, Oriya, Urdu and Malayalam. The fractal dimension is effective for distinguishing between 'matra' based scripts from their counterparts. If the average fractal dimension of top and bottom profile is computed (from 'matra' and without 'matra' scripts), then there will be a significant difference in average pixel density. Circularity feature was well suited to distinguish scripts like Oriya, Malayalam from others. Similarly, using small component analysis, scripts like Urdu can be easily distinguished from others. Using MLP classifier, an average accuracy rate of 92.8% was reported, although in Bangla and Devanagari script the rate had dropped to 8.4% and 9.4% respectively from the average rate. It can be observed that, for both the scripts, the misclassification rate were 9.4% (Bangla as Devanagari) and 8.3% (Devanagari as Bangla). The same may be due to the presence of 'matra' or 'shirorekha' in both the scripts. In 8.3% of the cases, the Devanagari script was misclassified as Malayalam also, as both the scripts have maximum structural dissimilarity. This issue needs to be addressed in future. This work uses very less test document images. So for better evaluation of the system, test document sets should considerably increase.

In another work [84], we proposed a page-level technique to identify Bangla, Devanagari, Roman and Urdu scripts using convolution based features namely Gabor filter bank and directional morphological filter. Gabor filter is a very popular texture computation tool which had been used with varying frequencies and orientations to compute feature values. These values and parameters to generate the filter bank had been chosen experimentally. The other feature used in their work was based on directional morphological filter. Observing the presence of different directional strokes in Indic scripts, four different morphological filters, namely horizontal, vertical, left

diagonal and right diagonal filters had been built. Important morphological operations, namely dilation and erosion were carried out using them to extract the prominent directional strokes from these four scripts. Then, feature values were computed measuring the ratios of original images with the dilated and eroded images. The reported average, bi-script and tri-script accuracies of this work were 94.4%, 97.5% and 98.2% of their own data set.

Different frequency domain techniques, namely Discrete Cosine Transform (DCT), Distance Transform (DT), Radon Transform (RT) and Fast Fourier Transform (FFT) had been applied to convert the page-level images at frequency domain and then some statistical feature values were computed from each of them to identify four eastern Indian scripts namely Bangla, Roman, Devanagari [106]. As per reported accuracies of these techniques, the average bi-script and tri-script accuracies were found to be 88.1%, 94.3% and 89.7% respectively. In this work, the combined performances of all the frequency domain techniques were measured. Looking back at the previous two works of the same author in terms of feature, we found component level features perform pretty well in distinguishing different Indic scripts in comparison to the frequency domain approach. Another advantage of component level features is their computational speedity in comparison to frequency domain techniques which are relatively slower when applied to the whole images.

All the three of our earlier works mentioned above had been carried out only on a subset of the official Indic script set. This was due to unavailability of a complete dataset of all official Indic scripts. Once we prepared the *PHDIndic_11* dataset, which has been discussed in Chapter 2, we proposed a page-level script identification technique considering all official Indic scripts, i.e. 11 Indic scripts. In the following section, we have elaborated the same.

FEATURE EXTRACTION AND DESIGN OF FEATURE SET

During feature extraction, visual observations have been made on Indic scripts to study the nature of different graphemes of different scripts. The main features considered are

structural along with few texture based features. The features considered for this work are as follows:

- $\boldsymbol{FS_{SVA}}$ **–** Structural and Visual Appearance based feature set. Overall feature dimension is 42 [see Chapter 3, Table 3.1]
- $\boldsymbol{FS_{FGA}}$ **–** A topological feature of dimension 2 [see Chapter 3, Table 3.1]
- $\boldsymbol{FS_{DSI}}$ **–** Directional Stroke Identification based feature. Morphological operations: erosion, dilation, opening, closing, gradient, top-hat and black-hat have been used to generate the feature vector. The FS_{DSI} feature dimension is 72 in our experiment [see Chapter 3, Table 3.1].

Final feature set for present page-level script identification problem:

$\boldsymbol{FS_{SVA \cup FGA \cup DSI}}$ $\boldsymbol{= FS_{SVA} \cup FS_{FGA} \cup FS_{DSI}}$

= 116 dimensions

In rest of the experiment, we have considered $FS_{SVA} = FS_{SVA} \cup FS_{FGA}$, while compared with FS_{DSI}. For simplicity in discussion, many times in text SVA represents FS_{SVA} and DSI represents FS_{DSI}.

To classify the features we have used three state-of-the-art classifiers: MLP, Simple Logistic and their combination through Voting. In voting different combinations of probability estimations can be done. Different combination rules for voting are: average of probabilities, product of probabilities, majority voting, minimum probability, maximum probability and median. In our case, we have computed average of probabilities of two default classifiers MLP and SL. For evaluation, we have investigated total three types of script identification scenarios: (i) bi- script (ii) tri-script and (iii) multi-script (in our case it is 11). Most multi-script documents in India are generally bi-script in nature, so this case has been handled first. Presence of tri-script documents in real life encourages us to test this case also. Finally, being encouraged from the bi-script and tri-script results we have tested the 11-script scenario. Here, we have explored the possibility of recognizing a script of a new page. So, this is a kind of

blind script recognizer, where the training set contains sample pages from all the classes. In the following section, we have explored each of these cases separately.

BI-SCRIPT IDENTIFICATION

We come across innumerable bi-script documents in our day to day life. In many cases a local script along with Roman or Devanagari can make a bi-script document. But in general scenario, with 11 scripts we have a total of $^{11}C_2$ or 55 bi-script classes. The bi-script identification accuracies have been reported in Table 5.1 to Table 5.5 respectively. Here, we have investigated the performances of both the features, individually and collectively. Table 5.1 shows the bi-script identification accuracy using $\boldsymbol{FS_{SVA}}$ feature only. Here, the upper and lower triangles provide the results of MLP and SL classifiers respectively. As mentioned earlier, total $^{11}C_2$ or 55 bi-script combinations are there. Here average bi-script identification accuracy using MLP and SL has been found to be 99.25% and 99.06% respectively. In both the cases, the standard deviations have been found to be 1.24 and 1.12 for MLP and SL respectively. So, it is evident from the reported accuracies that, SVA feature alone is strong enough to distinguish 11 Indic scripts in the bi-script scenario. In many cases, 100% accuracy has been reported which is really encouraging. So, structural and visual appearances can be used as a sole pertinent feature to distinguish different Indic scripts. Table 5.2 shows the performance of $\boldsymbol{FS_{DSI}}$ feature using MLP and SL classifiers. Similar to the previous Table 5.1, in Table 5.2 also, the upper and lower triangles provide the results of MLP and SL classifiers respectively. Here, the average bi-script identification accuracy using MLP and SL has been found to be 98.57% and 98.53% respectively. Though the reported accuracies using only $\boldsymbol{FS_{DSI}}$ feature is little bit less than $\boldsymbol{FS_{SVA}}$ features, but still they are comparable enough due to the inherent complexities of different handwritten Indic scripts. Comparing both the Table 5.1 and Table 5.2, it can be concluded that the feature-classifier combination ranking are $\mu_{(SVA\text{-}MLP)} > \mu_{(SVA\text{-}SL)} > \mu_{(DSI\text{-}MLP)} > \mu_{(DSI\text{-}SL)}$ in terms of average bi-script identification accuracies. While studying the consistency of the feature-classifier combination it has been observed that, $\sigma_{(SVA\text{-}SL)} < \sigma_{(SVA\text{-}MLP)} < \sigma_{(DSI\text{-}SL)} < \sigma_{(DSI\text{-}MLP)}$. So, in individual category the combination of $\boldsymbol{FS_{SVA}}$ feature with either

MLP or SL classifier handles the bi-script identification issue very well. $\boldsymbol{FS_{DSI}}$ feature alone is also comparable like $\boldsymbol{FS_{SVA}}$ but it lacks in some cases producing lower identification accuracy. For example, the bi-script identification accuracies for Roman-Tamil group are only 94.1% and 94.5% using $\boldsymbol{FS_{DSI}}-$MLP and $\boldsymbol{FS_{DSI}}-$SL respectively. But, the Roman-Tamil bi-script classification is handled very effectively by $\boldsymbol{FS_{SVA}}$ feature with average accuracies of 100% and 99.2% using $\boldsymbol{FS_{SVA}}-$MLP and $\boldsymbol{FS_{SVA}-SL}$ respectively. Table 5.3 shows the bi-script identification accuracies when $\boldsymbol{FS_{SVA}}$ and $\boldsymbol{FS_{DSI}}$ features are combinedly considered. Here we have combined $\boldsymbol{FS_{SVA}}$ and $\boldsymbol{FS_{DSI}}$ feature and tested their performance using both MLP and SL classifiers. The upper triangular matrix provides the result of MLP and lower triangular matrix provides the result of the SL classifier. Experimental results show notable improvements compared to previous results when $\boldsymbol{FS_{SVA}}$ and $\boldsymbol{FS_{DSI}}$ features were applied as individual category. The average bi-script identification accuracies using $\boldsymbol{FS_{SVA}}$ feature has been found to be 99.66% and 99.53% for MLP and SL classifier respectively. The consistency of the results by this feature combination can be found from the reported standard deviations, which are only 0.47 and 0.56 for MLP and SL respectively. Close inspection in Table 5.1, Table 5.2 and Table 5.3 reveals that $\boldsymbol{FS_{SVA}}$ features performs better compared to $\boldsymbol{FS_{DSI}}$ in most of bi-script classes, although few instances are there where $\boldsymbol{FS_{DSI}}$ performs better than $\boldsymbol{FS_{SVA}}$. Actually, it depends on the particular scripts considered. But, in general, we can say that structural variability is the most important feature to distinguish different Indic scripts. But still there are few misclassification instances of $\boldsymbol{FS_{SVA}}$, which can be overcome to certain extent through the use of DSI feature. As an example, the identification accuracy of Bangla-Malayalam is 95.6% and 95.9% by $\boldsymbol{FS_{SVA}}-$MLP and $\boldsymbol{FS_{DSI}}-$MLP respectively. But the success rate improves to 99.7% when $\boldsymbol{FS_{SVA} \cup FS_{DSI}}-$MLP is used. For $\boldsymbol{FS_{SVA} \cup FS_{DSI}}$ feature combination, we have found $\mu_{((SVA+DSI)\text{-}MLP)} > \mu_{((SVA+DSI)\text{-}SL)}$, and $\sigma_{((SVA+DSI)\text{-}MLP)} < \sigma_{((SVA+DSI)\text{-}SL)}$. So, a combined use of both $\boldsymbol{FS_{SVA}}$ and $\boldsymbol{FS_{DSI}}$ is suggestive along with MLP classifier for optimum results. The lower value of standard deviation proves that $\boldsymbol{FS_{SVA} \cup FS_{DSI}}$ are quite strong enough to successfully recognize different variations

of bi-script combinations. Finally, in present experiment, the benchmark bi-script results are found to be 99.66% average identification accuracies and 0.47 standard deviation using the feature set $\boldsymbol{FS_{SVA} \cup FS_{DSI}}$ and MLP classifier. Various feature-classifier combinations have been studied in Table 5.4 and Table 5.5. These tables show two special bi-script cases, where in the first case, Roman is kept common along with any regional script and in later case, Devanagari is kept common. These two combinations are realistic bi-script documents in India, which is why they have specially been studied. Furthermore, here we have performed integration of MLP and ML classifier through Voting and its performance is also analyzed. Considering $\boldsymbol{FS_{SVA}}$ and $\boldsymbol{FS_{DSI}}$ individually, as per identification accuracies of various feature-classifier combinations, we have found $\mu_{(SVA\text{-}Voting)} > \mu_{(SVA\text{-}SL)} > \mu_{(SVA\text{-}MLP)} > \mu_{(DSI\text{-}Voting)} > \mu_{(DSI\text{-}MLP)} > \mu_{(DSI\text{-}SL)}$. Individually, $\boldsymbol{FS_{SVA}}$ performs better compared to $\boldsymbol{FS_{DSI}}$ irrespective of particular classifier.

Table 5.1 Page-level bi-script identification accuracies (%) using FS_{SVA} feature. The upper triangular part of the matrix provides the results with MLP classifier and lower triangular part provides results with SL classifier. The script names are abbreviated as follows: Ben- Bangla, Dev- Devanagari, Guj- Gujarati, Gur- Gurumukhi, Kan- Kannada, Mal- Malayalam, Ory- Oriya, Rom- Roman, Tam- Tamil, Tel- Telugu, Urd- Urdu.

	Ben	Dev	Guj	Gur	Kan	Mal	Ory	Rom	Tam	Tel	Urd	Avg. Bi-script identification accuracy and standard deviation using $\boldsymbol{FS_{SVA}}$ feature and **MLP** classifier is **99.25% & 1.24**
Ben		99.3	100	99	100	95.6	98.5	97.9	94.7	100	100	
Dev	98.5		100	96.4	100	99.7	99.8	99.8	98.9	100	100	
Guj	100	99.7		100	100	100	99.7	100	99.6	100	100	
Gur	98	96.4	100		100	100	100	99.6	99.7	100	100	
Kan	99.1	98.5	99.4	100		100	99.6	98.2	98.2	98.5	100	
Mal	97.4	99.7	100	100	97.4		98.6	99.1	95.2	100	100	
Ory	98.5	99.5	99.7	99.7	99.6	98.6		99	99.4	100	100	
Rom	99.3	99.8	99.6	100	97.5	100	98.7		100	98	99.7	
Tam	96.1	98	98.7	99.7	97	97	98.3	99.2		98.1	99.4	
Tel	100	99.7	100	100	97	100	99.7	98.5	99.1		100	
Urd	100	99.8	100	100	99.6	100	99.8	99.7	99.7	99.7		
Avg. Bi-script identification accuracy and standard deviation using $\boldsymbol{FS_{SVA}}$ feature and **SL** classifier is **99.06% & 1.12**												

Table 5.2 Page-level bi-script identification accuracies (%) using FS_{DSI} feature. The upper triangular part of the matrix provides the results with MLP classifier and lower triangular part provides results with SL classifier. Abbreviations have usual meaning as mentioned earlier. The script names are abbreviated as follows: Ben- Bangla, Dev- Devanagari, Guj- Gujarati, Gur- Gurumukhi, Kan- Kannada, Mal-Malayalam, Ory- Oriya, Rom- Roman, Tam- Tamil, Tel- Telugu, Urd- Urdu.

	Ben	Dev	Guj	Gur	Kan	Mal	Ory	Rom	Tam	Tel	Urd	Avg. Bi-script identification accuracy and standard deviation using FS_{DSI} feature and **MLP** classifier is **98.57% & 1.99**
Ben		98	99.7	100	97.2	95.9	98.5	93.9	96.1	97.2	100	
Dev	97.2		99.4	98.3	97.8	98.5	99.5	98.3	98.9	98.7	100	
Guj	100	99.4		100	100	100	100	98.2	100	100	100	
Gur	99.7	98.3	100		100	100	100	100	100	100	100	
Kan	97.6	97.8	98.7	100		99.4	95.5	97.5	99.4	96.2	99.6	
Mal	97.4	97.3	99.6	99.6	98.1		99.7	90.5	95.2	97.4	100	
Ory	98.5	98.8	99.7	99.7	94.5	99.7		99.7	100	97.7	100	
Rom	95.3	98.3	98.6	100	96.3	93.3	99.4		94.1	96.5	100	
Tam	96.5	99.2	99.6	99.7	98.8	97	99.4	94.5		99.1	100	
Tel	97.2	99.4	99	100	98.5	95.9	98.1	96.5	99.6		100	
Urd	99.8	100	100	99.1	99.6	99.7	100	99.7	100	99.7		
Avg. Bi-script identification accuracy and standard deviation using FS_{DSI} feature and **SL** classifier is **98.53% & 1.62**												

Table 5.3 Page-level bi-script identification accuracies (%) when $FS_{SVA} \cup FS_{DSI}$ features are considered combinedly. The upper triangular part of the matrix provides the results with MLP and lower triangular part provides results with SL classifiers. Abbreviations have usual meaning as mentioned earlier. The script names are abbreviated as follows: Ben- Bangla, Dev- Devanagari, Guj- Gujarati, Gur-Gurumukhi, Kan- Kannada, Mal-Malayalam, Ory-Oriya, Rom- Roman, Tam- Tamil, Tel- Telugu, Urd-Urdu.

	Ben	Dev	Guj	Gur	Kan	Mal	Ory	Rom	Tam	Tel	Urd	Avg. Bi-script identification accuracy and standard deviation using $FS_{SVA} \cup FS_{DSI}$ feature and **MLP** classifier is **99.66%** & **0.47**
Ben		99.3	100	99.7	100	99.7	99.7	98.6	99.3	100	100	
Dev	99.5		99.7	98.9	99.3	99.7	99.8	99.5	99.8	99.7	100	
Guj	100	99.7		100	100	100	100	100	100	100	100	
Gur	100	97.2	100		100	100	100	100	100	100	100	
Kan	100	98.2	99.4	100		99.4	99.1	98.8	99.4	99.3	100	
Mal	99.3	99.4	100	99.6	99.4		99.7	99.6	98.7	100	100	
Ory	99.7	99.8	99.7	100	99.6	99.3		99.7	100	99.7	100	
Rom	99.7	99.5	99.6	100	98.2	100	99.7		97.9	98.5	100	
Tam	98.3	99.5	99.6	99.3	98.8	100	99	99.2		99.6	99.4	
Tel	100	99.1	100	100	99.3	100	99.7	99	99.1		100	
Urd	100	99.8	100	100	99.6	100	99.8	100	100	99.4		
Avg. Bi-script identification accuracy and standard deviation using $FS_{SVA} \cup FS_{DSI}$ feature and **SL** classifier is **99.53% & 0.56**												

Table 5.4 Page-level identification accuracies of various feature-classifier combination for the bi-scripts groups where, Roman is kept common with any one of the ten Indic scripts which is a realistic scenario in India. Abbreviations have usual meaning as mentioned earlier. The script names are abbreviated as follows: Ben- Bangla, Dev- Devanagari, Guj- Gujarati, Gur- Gurumukhi, Kan- Kannada, Mal- Malayalam, Ory- Oriya, Rom- Roman, Tam- Tamil, Tel- Telugu, Urd- Urdu.

		Ben	Dev	Guj	Gur	Kan	Mal	Ory	Tam	Tel	Urd	**μ**	**σ**
FS_{SVA}	MLP	97.9	99.8	100	99.6	98.2	99.1	99	100	98	99.7	99.13	0.82
	SL	99.3	99.8	99.6	100	97.5	100	98.7	99.2	98.5	99.7	99.23	0.79
	Voting	99	100	100	100	98.2	99.6	99.4	100	98	99.7	**99.39**	**0.75**
FS_{DSI}	MLP	93.9	98.3	98.2	100	97.5	90.5	99.7	94.1	96.5	100	96.87	3.14
	SL	95.3	98.3	98.6	100	96.3	93.3	99.4	94.5	96.5	99.7	97.19	**2.34**
	Voting	94.2	98.6	98.6	100	97.5	92.8	99.7	94.9	97	100	**97.33**	2.57
$FS_{SVA} \cup FS_{DSI}$	MLP	98.6	99.8	100	100	98.8	99.6	99.7	97.9	98.5	100	99.29	0.76
	SL	99.7	99.5	99.6	100	98.2	100	99.7	99.2	99	100	99.49	0.56
	Voting	99.7	99.5	99.6	100	98.2	100	99.7	99.2	99	100	**99.49**	**0.56**

Table 5.5 Page-level identification accuracies of various feature-classifier combination for the bi-scripts groups where, Devanagari is kept common with any one of the ten Indic scripts which is a realistic scenario in India. Abbreviations have usual meaning as mentioned earlier. The script names are abbreviated as follows: Ben- Bangla, Dev- Devanagari, Guj- Gujarati, Gur- Gurumukhi, Kan- Kannada, Mal- Malayalam, Ory- Oriya, Rom- Roman, Tam- Tamil, Tel- Telugu, Urd- Urdu.

		Ben	Guj	Gur	Kan	Mal	Ory	Rom	Tam	Tel	Urd	**μ**	**σ**
FS_{SVA}	MLP	99.3	100	96.4	100	99.7	99.8	99.8	98.9	100	100	99.39	1.11
	SL	98.5	99.7	96.4	98.5	99.7	99.5	99.8	98	99.7	99.8	98.96	1.11
	Voting	99	100	96.9	99.3	100	99.8	100	99.2	100	100	**99.42**	**0.96**
FS_{DSI}	MLP	98	99.4	98.3	97.8	98.5	99.5	98.3	98.9	98.7	100	98.74	**0.70**
	SL	97.2	99.4	98.3	97.8	97.3	98.8	98.3	99.2	99.4	100	98.57	0.94
	Voting	97.7	99.4	98.3	97.8	98.8	99.5	98.6	99.5	99.4	100	**98.90**	0.78
$FS_{SVA} \cup FS_{DSI}$	MLP	99.3	99.7	98.9	99.3	99.7	99.8	99.5	99.8	99.7	100	99.57	**0.32**
	SL	99.5	99.7	97.2	98.2	99.4	99.8	99.5	99.5	99.1	99.8	99.17	0.83
	Voting	99.5	100	98.6	99.3	99.4	100	99.5	99.8	99.7	100	**99.58**	0.43

TRI-SCRIPT IDENTIFICATION

Many official documents in India contain three scripts, namely, the states's official script, Roman and Devanagari. Such combination of three scripts has been referred as triplets of that state. Table 5.6, shows the experimental results of discriminating such triplets. Similar to our previous approach, first we have investigated the individual

performance of $\boldsymbol{FS_{SVA}}$ and $\boldsymbol{FS_{DSI}}$ features using MLP and SL classifier. Then the performance of classifier integration has also been tested. Finally, we have experimented $\boldsymbol{FS_{SVA}} \cup \boldsymbol{FS_{DSI}}$ features for MLP, SL and Voting classifiers. . It has been observed from Table 5.6, that in individual feature categories $\mu_{(\text{SVA-Voting})} > \mu_{(\text{DSI-Voting})}$. But, $\sigma_{(\text{DSI-SL})} < \sigma_{(\text{SVA-SL})}$, so, $\boldsymbol{FS_{DSI}}$ feature with SL classifier has better consistency compared to others. Now, we consider the feature combination where, $\boldsymbol{FS_{SVA}}$ and $\boldsymbol{FS_{DSI}}$ are used combinedly. Comparing individual $\boldsymbol{FS_{SVA}}$ and $\boldsymbol{FS_{DSI}}$ with $\boldsymbol{FS_{SVA}} \cup \boldsymbol{FS_{DSI}}$ it has been found that, $\mu_{((\text{SVA+DSI})\text{-Voting})} > \mu_{(\text{SVA-Voting})} > \mu_{(\text{DSI-Voting})}$ and $\sigma_{((\text{SVA+DSI})\text{-Voting})} < \sigma_{(\text{DSI-SL})} < \sigma_{(\text{SVA-SL})}$. For, $\boldsymbol{FS_{SVA}} \cup \boldsymbol{FS_{DSI}}$ feature $\mu_{((\text{SVA+DSI})\text{-MLP})}$, $\mu_{((\text{SVA+DSI})\text{-SL})}$ and $\mu_{((\text{SVA+DSI})\text{-Voting})}$ are 99.32%, 98.92% and 99.37%. On the other hand, $\sigma_{((\text{SVA+DSI})\text{-MLP})}$, $\sigma_{((\text{SVA+DSI})\text{-SL})}$ and $\sigma_{((\text{SVA+DSI})\text{-Voting})}$ are 0.63, 0.61 and 0.57 respectively. So, for tri-script identification the benchmark results have been reported as μ_{Voting} of 99.37% and σ_{Voting} of 0.57.

Table 5.6 Page-level identification accuracies (%) of various feature-classifier combination for the tri-scripts groups where, Roman and Devanagari is kept common with any one of the nine Indic scripts which is a realistic scenario in India. Abbreviations have usual meaning as mentioned earlier. The script names are abbreviated as follows: Ben- Bangla, Dev- Devanagari, Guj- Gujarati, Gur- Gurumukhi, Kan- Kannada, Mal- Malayalam, Ory- Oriya, Rom- Roman, Tam- Tamil, Tel- Telugu, Urd- Urdu.

		Ben	Guj	Gur	Kan	Mal	Ory	Tam	Tel	Urd	**μ**	**σ**
$\boldsymbol{FS_{SVA}}$	MLP	98.6	99.8	96.6	99.3	98.5	98.9	98.7	98.4	100	98.75	0.98
	SL	98.4	99.1	96.2	98.2	99.1	99.1	98.3	98.6	99.5	98.50	**0.96**
	Voting	98.8	99.8	96.6	98.7	99.4	99.1	99.4	98.6	100	**98.93**	0.99
$\boldsymbol{FS_{DSI}}$	MLP	94.4	98.7	97.7	97.4	95.3	98.1	96.3	97.7	99.1	**97.18**	1.56
	SL	96.4	96.6	97.5	96.4	95.7	97.1	96.1	97.2	98.7	96.85	**0.89**
	Voting	95	97.7	97.7	97.7	96	97.7	95.9	97.9	98.9	97.16	1.24
$\boldsymbol{FS_{SVA}} \cup \boldsymbol{FS_{DSI}}$	MLP	99.2	99.8	97.7	99.5	99.6	99.5	99.8	99.3	99.5	99.32	0.63
	SL	98.6	98.9	97.9	98.2	99.4	99.5	98.9	99.1	99.8	98.92	0.61
	Voting	99.2	99.6	97.9	99.5	99.8	99.5	99.6	99.6	99.7	**99.37**	**0.57**

MULTI-SCRIPT IDENTIFICATION

During multi-script identification we consider any number of scripts together and identify them. As mentioned earlier, this is a kind of blind script recognizer, where the

training set contains sample pages from all the classes. Observing the encouraging performance of our feature-classifier combination in bi-script and tri-script scenarios, we tried to identify the scripts in a multi-script scenario, involving all the 11 scripts considered earlier. Each test sample page is compared with the reference samples from all the other classes. The result of successful classification of each of the 11 scripts for different feature-classifier combinations is reported in Table 5.7. In this table, the last two rows show the average identification accuracy (denoted by μ) and standard deviation (denoted by σ) respectively for corresponding feature-classifier combination. Here, using only $\boldsymbol{FS_{SVA}}$ feature MLP and SL classifiers show average identification accuracy of 95.32% and 94.25% respectively. Using only $\boldsymbol{FS_{DSI}}$ feature these two classifiers produce an average identification accuracy of 91.95% and 91.55% respectively. In a different test, we have integrated MLP and SL classifier using Voting, and encouraging improvements have been found for both of the $\boldsymbol{FS_{SVA}}$ and $\boldsymbol{FS_{DSI}}$ features. It can be noticed from Table 5.7, that performance wise ($\boldsymbol{FS_{SVA}}$ −Voting) > ($\boldsymbol{FS_{SVA}}$ −MLP) > ($\boldsymbol{FS_{SVA}}$ −SL) > ($\boldsymbol{FS_{DSI}}$ −Voting) > ($\boldsymbol{FS_{DSI}}$ −SL) > ($\boldsymbol{FS_{DSI}}$ −MLP) for various feature-classifier combination. In both the individual cases of $\boldsymbol{FS_{SVA}}$ and $\boldsymbol{FS_{DSI}}$ features, $\sigma_{Voting} < \sigma_{MLP} < \sigma_{SL}$ so, proves the consistency of Voting for individual script discrimination. Now, we have experimented $\boldsymbol{FS_{SVA} \cup FS_{DSI}}$ feature and notable improvement in terms of μ and σ has been found compared to earlier ones when individually $\boldsymbol{FS_{SVA}}$ and $\boldsymbol{FS_{DSI}}$ were considered. Here μ_{MLP} shows highest average identification accuracy of 98.60%, which is slightly better (0.07%) than μ_{Voting}. But in terms of consistency Voting performs slightly better (0.7) than MLP. Again the differences are very much nominal. So, in conclusion, we can say that, $\boldsymbol{FS_{SVA} \cup FS_{DSI}}$ handle various multi-script scenarios pretty well using MLP. Classifier integration has also notable impact on the performance and consistency. Finally, benchmark average 11-script identification accuracy of 98.60% and standard deviation of 1.56 has been reported.

Table 5.7 Page-level identification accuracies (%) of FS_{SVA} and FS_{DSI} features individually and combinedly using MLP, SL and Voting classifier for multi-script scenario (11-script combination in our case). Abbreviations have usual meaning as mentioned earlier. The script names are abbreviated as follows: Ben- Bangla, Dev- Devanagari, Guj- Gujarati, Gur- Gurumukhi, Kan- Kannada, Mal- Malayalam, Ory- Oriya, Rom- Roman, Tam- Tamil, Tel- Telugu, Urd- Urdu.

	FS_{SVA}			FS_{DSI}			$FS_{SVA} \cup FS_{DSI}$		
	MLP	SL	Voting	MLP	SL	Voting	MLP	SL	Voting
Ben	93.78	95.03	95.03	89.44	90.06	91.92	98.75	98.13	98.13
Dev	95.45	94.09	96.36	91.81	95.45	95	97.27	97.27	96.81
Guj	100	100	100	97	97	99	100	100	100
Gur	98.48	95.45	96.96	96.96	98.48	98.48	99.24	99.24	99.24
Kan	89.13	82.60	86.95	84.78	84.78	89.13	95.65	91.30	95.65
Mal	91.58	92.52	94.39	90.65	85.98	91.58	100	99.06	100
Ory	95.34	96.51	96.51	96.51	94.18	95.93	97.09	98.25	98.25
Rom	97.36	93.85	96.49	84.21	80.70	85.08	100	98.24	99.12
Tam	87.50	89.16	88.33	88.33	87.50	92.5	96.66	97.5	96.66
Tel	100	97.64	100	91.76	92.94	92.94	100	100	100
Urd	100	100	100	100	100	100	100	100	100
μ	95.32	94.25	**95.54**	91.95	91.55	**93.77**	**98.60**	98.09	98.53
σ	4.42	4.99	**4.37**	5.18	6.21	**4.49**	1.63	2.45	**1.56**

STATISTICAL SIGNIFICANCE TEST

We have carried out statistical significance test over multiple dataset to compare different classifiers [107]. To do so, we have carried out a safe and robust non-parametric Friedman test [108]. We know that, during repeated measures analysis of variances same parameter has been measured under different conditions on the same subjects. The Friedman test is actually an alternative for repeated measures analysis of variances. In this experiment, the number of classifiers (k) and dataset (N) are taken as 3 and 5 respectively. The datasets are dataset #1, dataset #2, dataset #3, dataset #4 and dataset #5 chosen randomly from the original dataset *PHDIndic_11*. The performance of different classifiers over different datasets is shown in Table 5.8. Based on the performance of identification accuracies, the classifiers are ranked separately for each of

the dataset, i.e. the best performing algorithm is assigned rank 1, the second best as rank 2 and so on (as shown in Table 5.8). Whenever there is a tie between more than one algorithm, average ranks are assigned.

Table 5.8 Statistical significance test: identification accuracies of three different classifiers MLP, SL and Voting, their corresponding rank on five different dataset (subset of the original dataset). In parenthesis, classifiers ranks are given for each dataset #1 to #5.

	Classifiers in % (rank)		
Dataset	**MLP**	**SL**	**Voting**
#1	95.86 (3)	97.24 (2)	97.59 (1)
#2	98.28 (1)	96.90 (3)	97.93 (2)
#3	98.28 (1)	95.55 (3)	97.24 (2)
#4	96.55 (1)	93.45 (3)	94.83 (2)
#5	97.99 (1.5)	97.32 (3)	97.99 (1.5)
Mean rank	$\mathbf{R_{MLP} = 1.5}$	$\mathbf{R_{SL} = 2.8}$	$\mathbf{R_{Voting} = 1.7}$

Let us consider r_j^i be the rank of j^{th} classifier on i^{th} dataset. The mean of the ranks of all the j^{th} classifiers over all the N datasets are computed as follows (Eq. 5.1):

$$R_j = \frac{1}{N}\sum_{i=1}^{N} r_j^i \qquad (5.1)$$

As per the statement of null hypothesis, all the classifiers are equivalent, i.e. their rank R_j should be equal. To justify this, we compute the Friedman statistics [108] using the following equation:

$$X_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] \qquad (5.2)$$

For the present experiment, this statistics is distributed according to X_F^2 with k-1 (=2) degree of freedom. Using the above Eq. 5.2, the value of X_F^2 is calculated as 4.9. The upper-tail critical values of chi-square distribution with k-1 degrees of freedom (here it is 2) is 5.991 for $\alpha = 0.05$. So, we conclude that for HSI results on *PHDIndic_11,* there is no significant difference in the performance of accuracies of three different classifiers

MLP, SL and Voting on five random datasets: dataset #1, dataset #2, dataset #3, dataset #4 and dataset #5. This, hereby, proves the effectiveness of the proposed dataset.

RESULTS AND COMPARATIVE STUDY

Following, Table 5.9 summarizes the results reported in Table 5.1 to Table 5.7 for quick view. Identification type, feature, classifier and benchmark results obtained for particular feature-classifier combination in terms of highest average accuracy rate (μ) and lowest standard deviation (σ) have been reported. In bi-script identification, if only $\boldsymbol{FS_{SVA}}$ feature is used, then MLP shows highest identification accuracy of 99.25%. On the other hand, if only $\boldsymbol{FS_{DSI}}$ feature is used then also MLP shows highest identification accuracy of 98.57%. But, in both of the cases, SL is more consistent than MLP with lowest standard deviation of 1.12 for $\boldsymbol{FS_{SVA}}$ and 1.62 for $\boldsymbol{FS_{DSI}}$. Finally, if both the features, $\boldsymbol{FS_{SVA}}$ and $\boldsymbol{FS_{DSI}}$ are combined then MLP performs best with highest average identification accuracy of 99.66% and lowest average standard deviation of 0.47. So, in bi-script scenario we can conclude that, $\boldsymbol{FS_{SVA} \cup FS_{DSI}}$ perform best along with MLP classifier. In other two special bi-script identification (once keeping Roman common and once Devanagari common), we have analyzed the performance of classifier integration along with individual classifiers. While Roman is kept common with other ten scripts, then $\boldsymbol{FS_{SVA} \cup FS_{DSI}}$ along with Voting perform best. While keeping Devanagari common, then $\boldsymbol{FS_{SVA} \cup FS_{DSI}}$ performs best along with Voting in terms of average identification accuracy but MLP shows more consistent with lowest average standard deviation. In tri-script scenario (where both Roman and Devanagari are kept common with other nine scripts), $\boldsymbol{FS_{SVA} \cup FS_{DSI}}$ performs best along with Voting, with highest average identification accuracy of 99.37% and lowest standard deviation of 0.57. Finally, in multi-script identification, $\boldsymbol{FS_{SVA} \cup FS_{DSI}}$ and MLP show highest average identification accuracy of 98.60%. But, Voting shows more consistent result for multi-script identification with lowest standard deviation of 1.56. As a matter of fact, we have come to a conclusion that, suitable feature combination always

has better impact on the identification rate and consistency. Among the classifiers, MLP outperforms SL in most of the cases, so it is preferred, but integrating classifiers show promising outcome.

The time complexity of the proposed techniques (i.e. $\boldsymbol{FS_{SVA}}$, $\boldsymbol{FS_{DSI}}$ and their combination) on different classifiers (i.e. MLP, SL and Voting) is reported in Figure 5.1. This time complexity evaluation was done under 5-fold cross validation approach. The experimentation was carried out in a machine with Intel core i3 2.13GHz processor and 4 GB memory. It can be found that for $\boldsymbol{FS_{SVA} \cup FS_{DSI}}$ feature SL performs faster compared to others followed by MLP and Voting.

Table 5.9 Summarization of the benchmark results (topmost values) from Table 5.1-5.7 for different identification types

Script identification type	**Feature**	**Classifier**	**Benchmark results**	
			μ	**σ**
Bi-script (average of all 55 combinations)	$\boldsymbol{FS_{SVA}}$	MLP	99.25	1.24
		SL	99.06	1.12
	$\boldsymbol{FS_{DSI}}$	MLP	98.57	1.99
		SL	98.53	1.62
	$\boldsymbol{FS_{SVA} \cup FS_{DSI}}$	MLP	99.66	0.47
		SL	99.53	0.56
Bi-script (keeping Roman common)	$\boldsymbol{FS_{SVA} \cup FS_{DSI}}$	Voting	99.49	0.56
Bi-script (keeping Devanagari common)	$\boldsymbol{FS_{SVA} \cup FS_{DSI}}$	Voting	98.58	0.43
		MLP	99.57	0.32
Tri-script (keeping Roman and Devanagari common)	$\boldsymbol{FS_{SVA} \cup FS_{DSI}}$	Voting	99.37	0.57
Multi-script (11-script scenario)	$\boldsymbol{FS_{SVA} \cup FS_{DSI}}$	Voting	98.53	1.56
		MLP	98.60	1.63

Finally, different methods described in [54] [109] [110] have been evaluated on the present dataset at 11-script scenario. All these techniques are experimented in the same setup, i.e. using Matlab 7.6.0 software, a machine with Intel core i3 2.13GHz processor and 4 GB memory. From the experiment (results are shown in Table 13), we have found that, as per identification accuracy of the features:

$FS_{SVA} \cup FS_{DSI}$ > FS_{SVA} > FS_{DSI} >> DWT+RT > WE > DCT > GLCM. So, the proposed features (in individual category or in combination) perform significantly better compared to state-of-the-arts (refer Table 5.10).

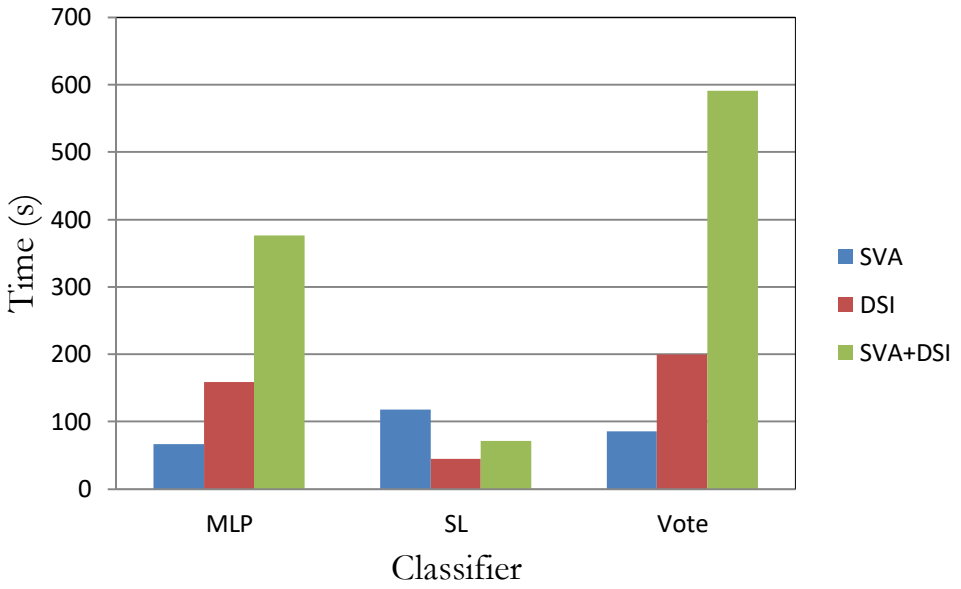


Figure 5.1 Time complexity of different feature-classifier combination using 5-fold cross validation approach, evaluation was done in a machine with Intel core i3 2.13GHz processor and 4 GB memory

Table 5.10 Comparative overview of different methods on the proposed *PHDIndic_11* dataset at 11-script scenario

Method	**Feature dimension**	**Scripts considered**	**Identification accuracy (%)**
GLCM [109]	44	Bangla, Devanagari, Roman, Urdu, Oriya, Gurumukhi, Gujarati, Tamil, Telugu, Malayalam and Kannada	77.29
Wavelet energy [110]	55		78.53
Texture analysis using DWT and Radon transform [54]	84		85.11
Directional stroke identification (FS_{DSI}) *(proposed)*	72		93.77
Structural and visual appearance (FS_{SVA}) *(proposed)*	44		95.54
$FS_{SVA} \cup FS_{DSI}$ *(proposed)*	116		98.60

5.1.2 HANDWRITTEN SCRIPT IDENTIFICATION – PAGE, BLOCK, LINE AND WORD-LEVEL APPROACH

In general handwritten script identification works were carried out at page, block, line or word level. Section 5.1.1 has already shown the outcome of handwritten script identification at page-level. Further, depending on the availability of dataset we have carried out block, line and word level script identification too. In [88], we have proposed a work on block-level script identification from six Indic scripts namely: Bangla, Devanagari, Malayalam, Oriya, Roman and Urdu. A dataset of volume 600 blocks with equal distribution of each script type was prepared for experimentation. We found the average bi-script, tri-script, tetra-script and all-script average accuracies of 95.33%, 88.89%, 87.18% and 81.9% respectively.

A line-level tri-script identification framework for eastern Indian documents was reported in one of our earlier work [111]. In another work [82], an automatic approach for line-level handwritten script identification (HSI), considering eight official Indic scripts namely: Bangla, Devanagari, Kannada, Malayalam, Oriya, Roman, Telugu and Urdu was proposed. For classification, we divided the whole script dataset based on different regions of India, to study a region-wise classification performance. A total of 2034 line-level document images are collected. For Kannada script, KHTD [29] dataset was used. A standard data collection form was prepared in our lab. Some of the forms contain pre-specified texts that were asked to be copied and some forms were left blank where users were asked to write anything as per their choice. During feature extraction we have used some local features (script dependent) directional stroke features and texture feature. We also carried out exhaustive feature selection with all the three type of features and found the optimal performance for all features combined, i.e. combination of local, directional and texture features. Finally, experimentation was carried out using three different state-of-the-art classifiers: multilayer perceptron (MLP), random forest (RF) and fuzzy unordered rule induction algorithm (FURIA). Among all, we have observed MLP as the best performer in terms of average accuracy of 98.2%, 99.5%, 99.1%, 99.5%, 99.9%, 98%, 98.9% for eight-script, bi-script, eastern,

north, south Indian script groups, scripts with 'matra' versus without 'matra' and dravidian versus non-dravidian groups respectively

In another work [80], we present a novel framework, which can be used as a precursor to ease subsequent Indic scripts identification problem. Our proposed method starts with feature extraction for line-level documents, and three different classifiers to make a decision for separating scripts with and without 'matra'. For feature extraction, fractal geometry analysis (FGA), Canny edge detector (CED) and morphological line transform (LT) are used. Similarly, for script separation task, three different classifiers such as multi-layer perceptron (MLP), Bayesnet (BN) and random forest (RF) are used. We carefully check which combination (i.e., feature-classifier) performs the best. For this work, we have prepared a dataset of 1204 line-level handwritten document images. Out of which, 525 lines belong to scripts with 'matra' and remaining 679 lines are from scripts without 'matra'. Among the scripts with 'matra', there are 325 lines from Bangla and 200 lines from Devanagari script. On the other hand, for scripts without 'matra', Roman and Urdu contribute 370 and 309 lines, respectively. The dataset was collected from different persons with varying age, sex, educational background and demographic location. The main task of the precursor is to separate scripts with 'matra' from their counterpart. For simplicity, we call it as a two class problem. In our dataset, the scripts with 'matra' (i.e., Bangla and Devanagari) are labelled as class 1, and the scripts without 'matra' (i.e., Roman and Urdu) are labelled as class 2. Different features (i.e., FGA, CED and LT) and classifiers (i.e., MLP, BN, RF) as explained in Chapter 3 are evaluated. In Table 5.11, we achieved the highest possible accuracies of 95.68%, 85.30% and 76.49% for FGA-RF, CED-MLP and LT-BN feature-classifier combination respectively. FGA outperforms other features when combining with RF classifier, and overall we have FGA-RF > CED-MLP > LT-BN. This can be supported by computing standard deviation of accuracies by these three classifiers for FGA, CED and LT as 0.6614, 1.7276 and 12.0517, respectively. It implies that the FGA is more robust as compared to others. The primary reason behind using FGA is due to the fact that it works on the principle of non-Euclidean geometry, and handwritten scripts tend

to have more crooked lines than straight ones resemble non-Euclidean geometry. Between CED and LT, CED performs better, since line transform cannot classify script like Roman.

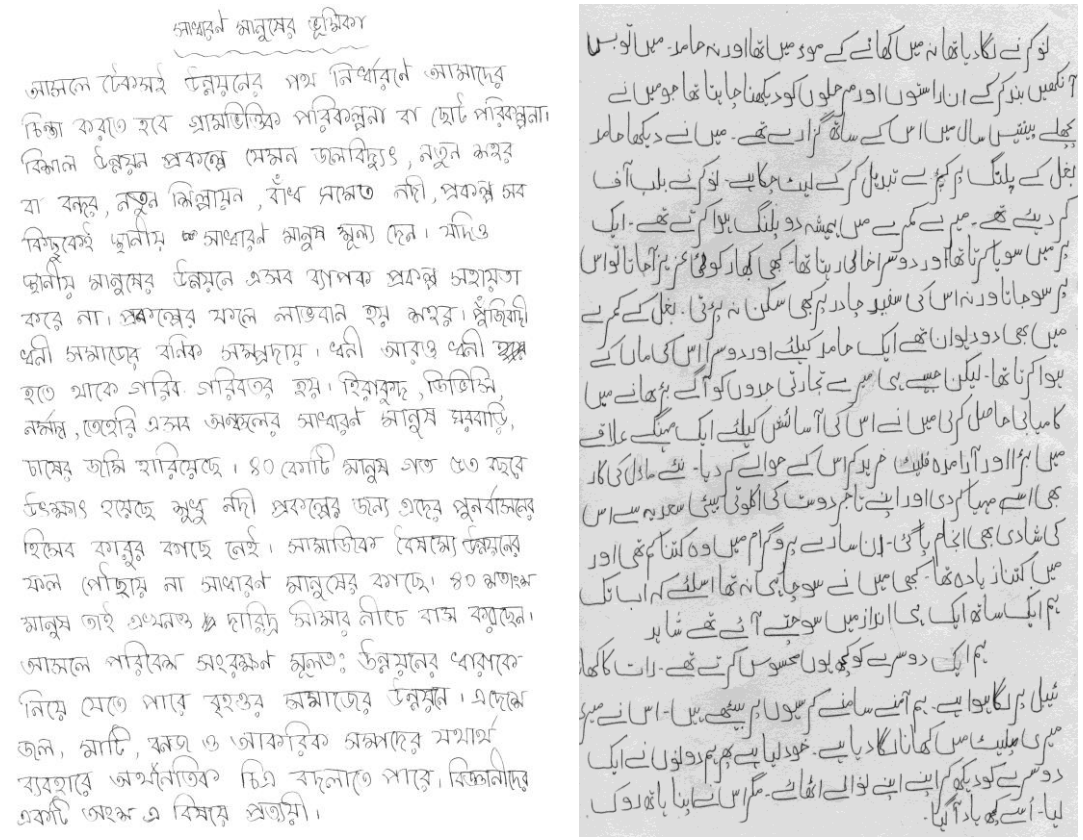
Table 5.11 Script separation: using three different features and three different classifiers, measured in terms of sensitivity, specificity and accuracy (all in %). The number of scripts considered are eleven, i.e. Bangla, Devanagari, Gujarati, Gurumukhi, Kannada, Malayalam, Oriya, Roman, Tamil, Telugu, Urdu.

Feature	Classifier	Sensitivity	Specificity	Accuracy
FGA	RF	93.90	97.05	95.68
	BN	92.95	96.02	94.68
	MLP	91.61	96.61	94.43
CED	RF	72.38	89.54	82.06
	BN	67.80	97.79	84.72
	MLP	69.14	97.79	85.30
LT	RF	72.19	77.76	75.33
	BN	75.61	89.54	76.49
	MLP	35.42	70.25	55.05

HANDWRITTEN SCRIPT IDENTIFICATION – A MULTI-LEVEL FRAMEWORK

In our survey we have found that, all the script identification works are categorized into several levels namely: page-level, block-level, line-level, word-level and character-level. Our claim is also supported by the review presented by Ghosh et al. [8]. Although each of these levels has their own applicability, no experimental or empirical support is provided till date towards considering a particular level of work. To bridge this gap, we propose a multi-level framework for handwritten script identification. In this experiment, we consider the same document into page, block, line and word-level and studied the script identification performance at each level. In next section some sample images are shown from our dataset. Figure 5.2 shows two page-level documents of Bangla and Urdu script. Block-level images obtained from the same page are shown in Figure 5.3. Figure 5.4 shows four line-level images of Bangla and Urdu scripts where the upper two are from Bangla and bottommost two are from Urdu. These lines are

extracted from the same page-level images as shown in Figure 5.2. In Figure 5.5 sample word-level images are shown, which are again extracted from the same pages as shown in Figure 5.2.



(a) (b)

Figure 5.2 Sample page-level document (a) Bangla, (b) Urdu

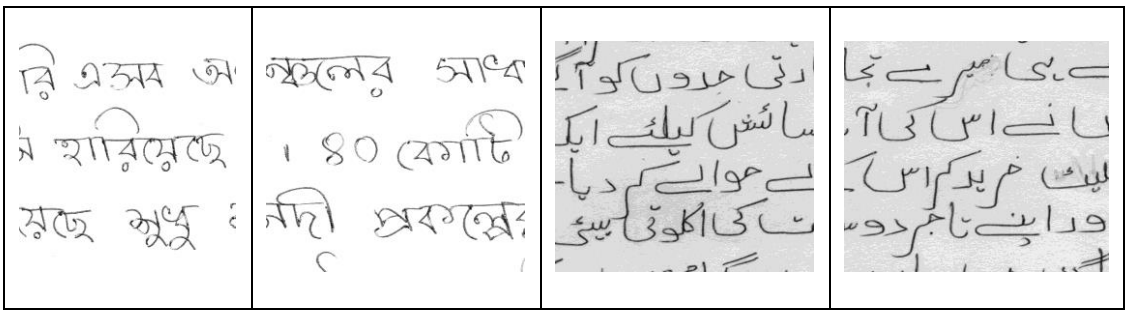


Figure 5.3 Sample block-level document extracted from the same page as shown in Figure 5.2

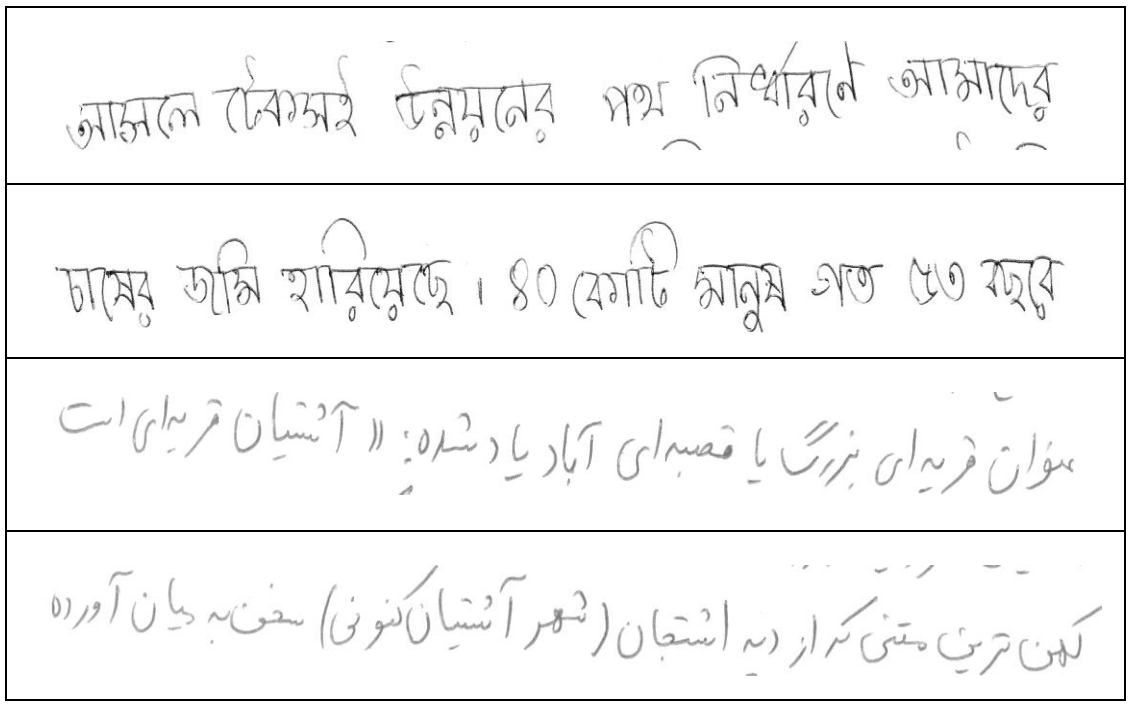


Figure 5.4 Sample line-level document extracted from the same page as shown in Figure 5.2

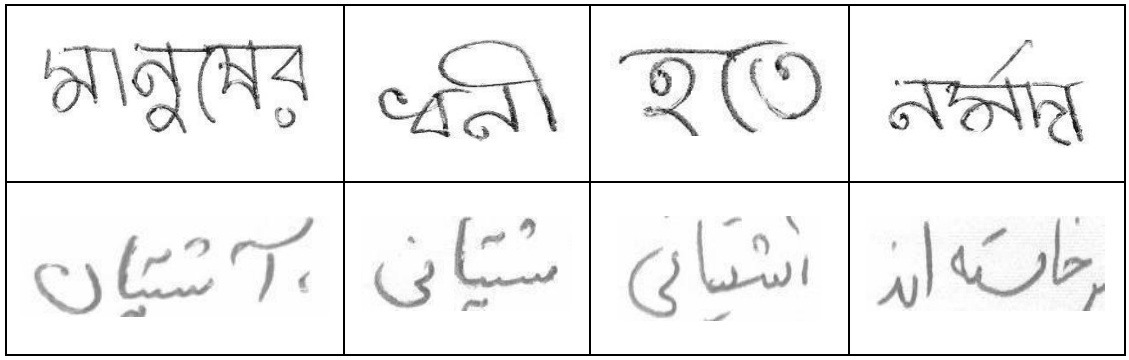


Figure 5.5 Sample word-level document extracted from the same page as shown in Figure 5.2

The dataset distribution for the experiment is as follows: total 440 pages, i.e. 40 pages from each of the eleven scripts, 2200 blocks, i.e. 200 blocks from each of the eleven scripts, 3300 lines, i.e. 300 lines from each of the eleven scripts and 6600 words, i.e. 600 words from each of the eleven scripts are considered. So, the distribution of page, block, line and word-level data is in a ratio of 1:5:7.5:15. This ratio means, from a single page on an average, 5 blocks, 7.5 lines and 15 words are considered. In this dataset, the Bangla pages are collected from CAMTER [4] and Kannada pages are considered from

KHTD dataset [68]. The rest of the images of remaining nine scripts are collected from different sources across the country [52]. We have already discussed the data collection process in Chapter 2. Table 5.12 shows the dataset distribution at different level.

Table 5.12 Dataset distribution of page, block, line and word-level documents

Multi-level Experiment Dataset Statistics				
	Page-level	Block-level	Line-level	Word-level
Bangla	40	200	300	600
Devanagari	40	200	300	600
Gujarati	40	200	300	600
Gurumukhi	40	200	300	600
Kannada	40	200	300	600
Malayalam	40	200	300	600
Oriya	40	200	300	600
Roman	40	200	300	600
Tamil	40	200	300	600
Telugu	40	200	300	600
Urdu	40	200	300	600
Total	**440**	**2200**	**3300**	**6600**

As we have already mentioned in this experiment, our objective is to study the effect of document segmentation into page, block, line and word-level towards script identification performance.

Feature extraction and design of feature set

Two types of features are considered in this work: (i) script dependent feature and (ii) global texture feature. These features are briefly pointed out below.

Script dependent features:

- $\boldsymbol{FS_{SVA}}$ **–** Structural and Visual Appearance based feature set. Overall feature dimension is 42 [Refer Chapter 3, Table 3.1]
- $\boldsymbol{FS_{FGA}}$ **–** A topological feature of dimension 2 [Refer Chapter 3, Table 3.1]

- $\boldsymbol{FS_{DSI}}$ **–** Directional Stroke Identification based feature. Morphological operations: erosion and dilation were used to generate the feature vector. The DSI feature dimension is 12 in our experiment [Refer Chapter 3, Table 3.1].

Final feature set for present page-level script identification problem:

$\boldsymbol{FS_{SVA \cup FGA \cup DSI}}$ $\boldsymbol{= FS_{SVA} \cup FS_{FGA} \cup FS_{DSI}}$

= 56 dimensions

Script Independent features:

- $\boldsymbol{FS_{SE}}$ **–** Feature based on spatial energy. Overall feature dimension is 04 [Refer Chapter 3, Table 3.2]
- $\boldsymbol{FS_{WRT}\#1}$ **–** It is a fusion of wavelet and radon transform. Overall feature dimension is 52 [Refer Chapter 3, Table 3.2]

Final feature set for present page-level script identification problem:

$\boldsymbol{FS_{SE \cup WET\#1}}$ $\boldsymbol{= FS_{SE} \cup FS_{WRT\#1}}$

= 56 dimensions

For simplicity, in rest of the chapter $FS_{SVA \cup FGA \cup DSI}$ and $FS_{SE \cup WET\#1}$ are abbreviated as FS_{SD} and FS_{SI} respectively.

Details about these features are discussed in Chapter 3. To make a fair comparison the feature dimension were kept same for both types of features, i.e. 56 dimensional features from both the categories. Finally, to evaluate the performance, we consider two state-of-the-art classifiers: multilayer perceptron (MLP) and random forest (RF) because of their promising performance in our earlier work [12]. Table 5.13 shows the performance of script dependent (SD) feature at page, line, block and word-level. Here we have shown the average multi-script (11 script in our experiment) identification accuracy (μ) and standard deviation (σ) along with individual script identification performance. In a similar manner, Table 5.14 shows the same performance of script independent (SI) features at the same four levels.

Experimental results and analysis

Based on state-of-the-art feature-classifier combination we need to investigate the script identification performance at different real life scenarios considering all the levels. The outcome is reported in the following section.

MULTI-SCRIPT IDENTIFICATION

As mentioned earlier, in general scenario, we performed the multi-script identification, so all the eleven scripts are considered together. Performances are evaluated using two types of features: script dependent (FS_{SD}), Script independent (FS_{SI}) and two state-of-the-art classifiers: multilayer perceptron (MLP), random forest (RF). Total number of experiment for multi-script identification will be four, i.e. one experiment for each level. The abbreviation for the scripts mentioned in rests of the paper is as follows: Ben- Bangla, Dev- Devanagari, Guj- Gujarati, Gur- Gurumukhi, Kan- Kannada, Mal- Malayalam, Ory- Oriya, Rom- Roman, Tam- Tamil, Tel- Telugu, Urd- Urdu.

Table 5.13 and Table 5.14 show the performance of FS_{SD} and FS_{SI} features accordingly at page, line, block and word-level. Here we have shown the average multi-script (11 script in our experiment) identification accuracy (μ) and standard deviation (σ) along with individual script identification performance. In Table 5.13, we have observed highest average identification accuracies using FS_{SD} feature as follows: for Page-level: 94.32%, 93.19%, Block-level: 94.05%, 92.23%, Line-level: 93.73%, 94.61%, Word-level: 79.00%, 88.83%, using MLP and RF accordingly. Similarly, from Table 5.14, we have noticed highest average accuracies using FS_{SI} features as follows: for Page-level: 83.64%, 85.00%, Block-level: 87.32%, 86.32%, Line-level: 93.19%, 92.70%, Word-level: 91.04%, 86.29%, using MLP and RF accordingly. While comparing the performance of MLP and RF we have found that, at page and block-level MLP performs better than RF, line-level is almost comparable, and RF outperforms MLP at word-level. On the other hand, using Script independent feature, MLP outperforms RF at block, line and word-level. In our experiment, MLP is the top performer in most of the situations independent of the feature type. This is why we have carried out the

remaining experiments using MLP classifier. The accuracy chart for feature-classifier combination is shown by Figure 5.6.

Table 5.15 is summarization of Table 5.13 and Table 5.14. In addition with individual feature performance, in this table we have also reported the performance of feature combination i.e. performance of $FS_{SD} \cup FS_{SI}$ features at page, line, block and work-level. It is evident that, MLP outperforms RF in most of the scenarios irrespective of the feature chosen. So, rest of analysis we have done is on the result produced by MLP classifier. The standard deviation (σ) of both the features (FS_{SD} and FS_{SI}) at each level is computed as shown by Figure 5.7. In terms of the performance of feature independency: line-level > block-level > page-level > word-level. So the observation is: line-level data are more stable irrespective of the features chosen. Block and page-level data are comparatively similar and performance of word-level data are very much feature dependent In our experiment, we have found the highest standard deviation of 14.5 at word-level for FS_{SD} and FS_{SI} features, whereas for the same features the lowest value of 0.38 found at line-level. Now, while considering individual feature type, we found that, script dependent features are more suitable at page-level and texture feature are more suitable at word-level. In Table 5.15, the highest accuracy produced by individual feature types is underlined. We have also carried out the experiment using feature combination (i.e.$FS_{SD} \cup FS_{SI}$) to observer if word-level identification accuracies could be improved to some extent. Using feature combination, i.e. using $FS_{SD} \cup FS_{SI}$ features, we obtained average accuracies of 98%, 98.75%, 97.50% and 93.93% at page, line, block and word-level correspondingly. As we expected, here we have found a notable improvement on word-level performance. In earlier, we got word-level accuracy of 70.54% applying only FS_{SD} feature. The same is increased to 93.93% while FS_{SD} features are used in combination with FS_{SI} features. In addition with that, Figure 5.7 shows the highest value of σ at word-level which is 14.5 (significantly high). But at the same level, while we combine both FS_{SD} and FS_{SI} features, we have found the value of σ as 2.14, which is significantly less compared to if individual features are applied. So, suitable feature combination has remarkable effect on the overall performance of script identification. Finally, our observation is: word-level script

identification is more challenging in terms of accuracy compared to page, line and block-level.

Table 5.13 Individual script-wise identification accuracy of Script dependent ($\boldsymbol{FS_{SD}}$) feature at page, block, line and word-level using MLP, RF and SVM classifier for multi-script scenario (11-script)

	Script dependent features ($\boldsymbol{FS_{SD}}$)							
	Page-level		Block-level		Line-level		Word-level	
	MLP	RF	MLP	RF	MLP	RF	MLP	RF
Ben	97.50	97.50	96.00	93.50	96.66	96.33	87.33	90.83
Dev	87.50	87.50	87.00	85.00	85.66	92.00	57.00	63.66
Guj	100	100	100	100	99.66	100	92.16	95.16
Gur	95.00	100	93.50	94.00	90.00	90.66	66.00	82.16
Kan	92.50	85.00	92.00	90.00	91.66	91.00	59.16	67.00
Mal	100	100	90.50	87.00	94.66	95.00	61.83	63.66
Ory	92.50	92.50	93.50	90.50	87.33	91.00	62.50	77.50
Rom	100	97.50	97.50	98.00	99.00	98.66	68.33	80.16
Tam	87.50	85.00	94.00	88.00	91.66	92.00	68.00	65.50
Tel	87.50	82.50	91.50	88.50	95.00	95.00	74.33	86.16
Urd	97.50	97.50	99.00	100	99.66	99.00	79.00	88.83
μ	94.32	93.19	94.05	92.23	93.73	94.61	70.54	78.26
σ	5.13	6.90	3.86	5.27	4.88	3.51	11.49	11.63

Table 5.14 Individual script-wise identification accuracy of Script independent ($\boldsymbol{FS_{SI}}$) feature at page, block, line and word-level using MLP and RF classifier for multi-script scenario (11-script combination in our case).

	Script independent features ($\boldsymbol{FS_{SI}}$)							
	Page-level		Block-level		Line-level		Word-level	
	MLP	RF	MLP	RF	MLP	RF	MLP	RF
Ben	75.00	72.50	90.00	87.50	96.00	93.33	94.33	88.50
Dev	87.50	85.00	84.50	82.50	90.66	94.33	79.50	70.33
Guj	100	100	99.50	99.00	98.66	98.66	95.16	95.83
Gur	75.00	100	85.50	89.50	96.00	95.33	88.50	77.83
Kan	90.00	85.00	94.50	91.50	98.66	97.00	97.66	90.50
Mal	87.50	92.50	93.00	91.00	93.66	94.33	88.66	81.66
Ory	80.00	77.50	78.50	78.00	90.33	95.00	85.83	84.16
Rom	100	100	98.00	97.00	99.00	99.33	94.16	92.16
Tam	82.50	80.00	76.00	62.00	79.33	79.66	87.00	79.83
Tel	72.50	62.50	75.50	90.50	87.66	81.66	98.00	97.16
Urd	70.00	80.00	85.50	81.00	95.00	91.00	92.50	91.16
μ	83.64	85.00	87.32	86.32	93.19	92.70	91.04	86.29
σ	10.39	12.25	8.45	10.28	5.92	6.41	5.66	8.27

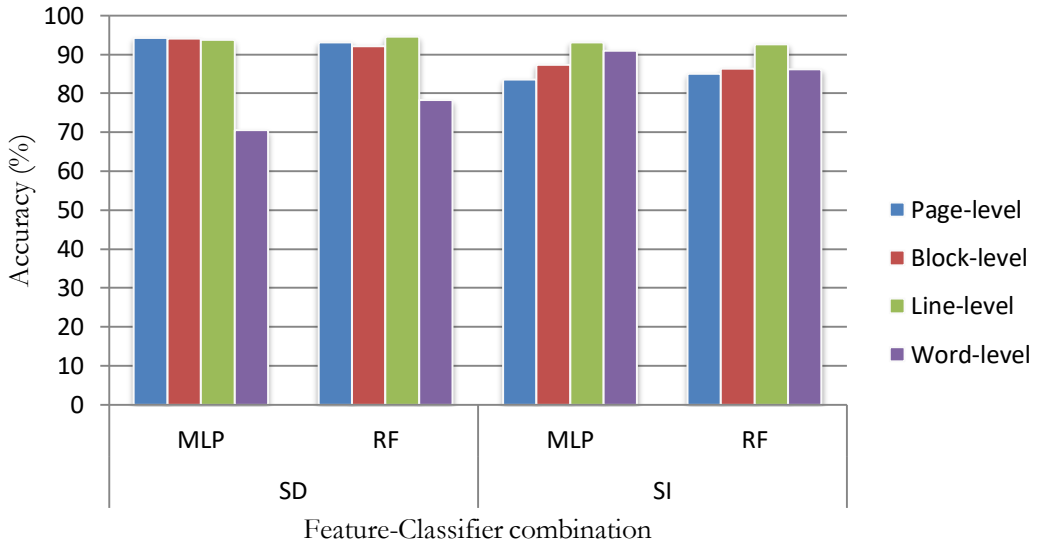


Figure 5.6 Performance of MLP and RF classifier in multi-script identification for different feature-classifier combinations at page, block, line and word-level

Table 5.15 Performance at page, block, line and word-level documents for multi-script scenario (11-script in our case), Feature: Script dependent (FS_{SD}) and Script independent (FS_{SI}) and their combination

Document type & Features		**Accuracies (%) with MLP and RF**	
Level	**Feature**	**MLP**	**RF**
Page	FS_{SD}	94.32	93.19
	FS_{SI}	83.64	85.00
	$FS_{SD} \cup FS_{SI}$	98.00	96.37
Block	FS_{SD}	94.05	92.23
	FS_{SI}	87.32	86.32
	$FS_{SD} \cup FS_{SI}$	97.50	96.14
Line	FS_{SD}	93.73	94.61
	FS_{SI}	93.19	92.70
	$FS_{SD} \cup FS_{SI}$	98.75	98.27
Word	FS_{SD}	70.54	78.26
	FS_{SI}	91.04	86.29
	$FS_{SD} \cup FS_{SI}$	93.93	89.37

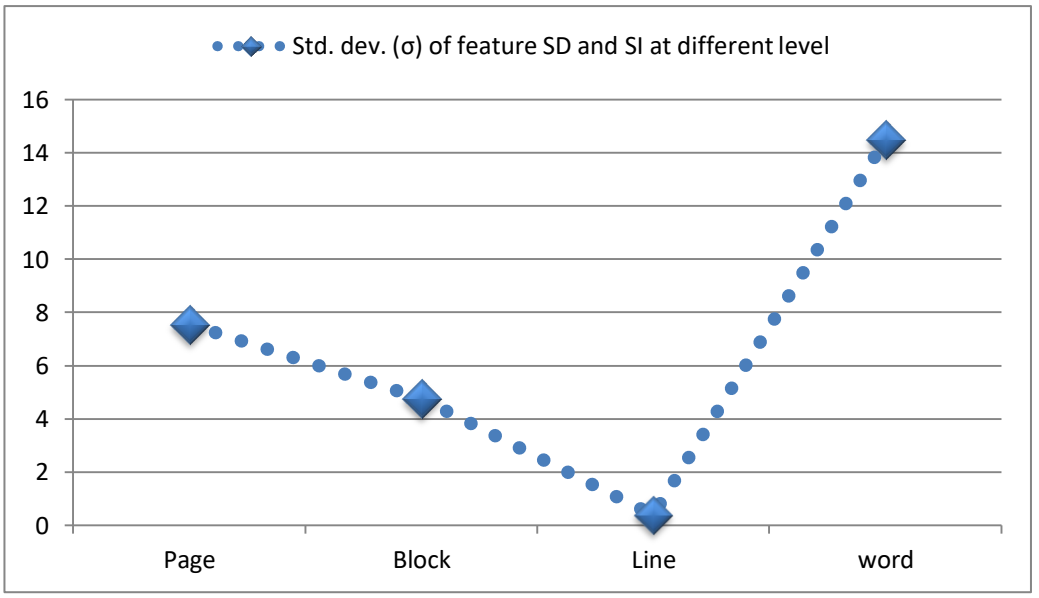


Figure 5.7 Standard deviation (σ) of FS_{SD} and FS_{SI} features at page, block, line and word-level is computed. Conclusion: with respect to feature independency: Line-level > Block-level > Page-level > Word-level

BI-SCRIPT IDENTIFICATION

Most of the Indian people are aware of more than one languages/scripts and that reflects on the documents when they write. Bi-script postal document is very common example of such document. So, bi-script identification is very much essential and obtaining good identification accuracy is crucial. We have eleven scripts, so total bi-script combinations will be ${}^{11}C_2$ or 55. The number of bi-script experiments we have carried out is 110 per level (55 for each of $\boldsymbol{FS_{SD}}$ and $\boldsymbol{FS_{SI}}$ features), so altogether 440 bi-script experiments we have conducted. Table 5.16, Table 5.17, Table 5.18 and Table 5.19 shows the page, block, and line and word-level bi-script identification performance reported in percentage.

Table 5.16 Bi-script identification accuracies (%) at Page-level using MLP classifier, the upper triangular part of the matrix provides the results with Script dependent (FS_{SD}) features and lower triangular part provides results with Script independent (FS_{SI}) features

Script	Ben	Dev	Guj	Gur	Kan	Mal	Ory	Rom	Tam	Tel	Urd	Avg. Bi-script identification accuracy and standard deviation using FS_{SD} feature is **98.93% & 1.79**
Ben		100	98.75	100	98.75	100	95.00	100	98.75	100	98.75	
Dev	98.75		100	92.50	98.75	100	97.50	98.75	100	100	97.50	
Guj	100	100		100	100	100	100	100	100	100	100	
Gur	98.75	95.00	100		100	100	98.75	100	98.75	100	100	
Kan	96.25	97.50	100	97.50		100	100	97.50	96.25	92.50	100	
Mal	97.50	100	100	98.75	100		97.50	100	95.00	100	100	
Ory	93.75	100	100	98.75	98.75	96.25		98.75	98.75	100	98.75	
Rom	98.75	100	100	98.75	100	100	98.75		100	98.75	100	
Tam	97.50	95.00	97.50	98.75	100	96.25	98.75	98.75		96.25	98.75	
Tel	96.25	96.25	100	98.75	100	91.25	85.00	100	91.25		100	
Urd	93.75	96.25	100	95.00	100	88.75	86.25	93.75	93.75	82.5		
Avg. Bi-script identification accuracy and standard deviation using FS_{SI} feature is **97.00% & 4.01**												

Table 5.17 Bi-script identification accuracies (%) at Block-level using MLP classifier, the upper triangular part of the matrix provides the results with Script dependent (FS_{SD}) features and lower triangular part provides results with Script independent (FS_{SI}) features

Script	Ben	Dev	Guj	Gur	Kan	Mal	Ory	Rom	Tam	Tel	Urd	Avg. Bi-script identification accuracy and standard deviation using FS_{SD} feature is **98.81% & 1.47**
Ben		96.00	100	100	99.50	97.75	97.25	99.75	99.25	99.50	100	
Dev	96.75		99.75	93.50	98.75	97.00	97.50	99.75	98.00	99.75	99.25	
Guj	100	100		99.75	100	100	99.75	99.00	99.75	100	100	
Gur	100	91.25	100		100	98.50	98.00	99.75	99.50	100	100	
Kan	99.00	98.75	99.75	100		99.00	97.50	99.00	97.50	94.5	100	
Mal	99.50	96.75	100	97.50	100		95.75	99.75	97.00	98.00	100	
Ory	95.75	97.50	100	99.50	98.00	98.25		99.25	97.00	98.50	100	
Rom	100	98.25	100	98.25	100	99.75	97.00		100	99.00	99.00	
Tam	100	90.00	99.75	99.50	100	94.25	91.00	99.00		98.00	100	
Tel	98.00	96.50	100	100	100	94.75	92.50	98.50	85.50		99.75	
Urd	98.50	94.75	100	97.75	98.50	96.50	94.50	96.75	93.50	95.00		
Avg. Bi-script identification accuracy and standard deviation using FS_{SI} feature is **97.57% & 3.08**												

Table 5.18 Bi-script identification accuracies (%) at Line-level using MLP classifier, the upper triangular part of the matrix provides the results with Script dependent ($\boldsymbol{FS_{SD}}$) features and lower triangular part provides results with Script independent ($\boldsymbol{FS_{SI}}$) features

Script	Ben	Dev	Guj	Gur	Kan	Mal	Ory	Rom	Tam	Tel	Urd	Avg. Bi-script identification accuracy and standard deviation using $\boldsymbol{FS_{SD}}$ feature is **99.24% & 1.28**
Ben		99.33	100	99.16	99.33	100	96.66	100	99.66	99.16	99.83	
Dev	99.00		100	92.50	99.83	100	99.50	100	99.33	99.66	99.66	
Guj	100	99.83		100	100	100	99.50	99.66	100	100	100	
Gur	100	97.83	99.83		99.83	99.83	98.33	100	99.83	99.33	99.83	
Kan	99.00	99.83	100	99.83		99.50	98.33	99.50	96.83	96.16	100	
Mal	99.83	99.33	99.83	99.50	100		98.33	100	98.00	99.16	100	
Ory	98.16	99.16	99.16	99.50	99.00	99.33		99.66	98.16	98.50	99.50	
Rom	100	100	100	100	100	100	99.83		100	99.00	100	
Tam	99.83	94.16	98.83	99.16	99.83	97.00	96.33	100		98.16	100	
Tel	99.00	97.16	100	100	99.50	95.33	99.66	100	90.83		99.83	
Urd	99.33	99.33	99.83	100	100	99.00	98.50	99.50	98.33	97.33		
Avg. Bi-script identification accuracy and standard deviation using $\boldsymbol{FS_{SI}}$ feature is **99.01% & 1.65**												

Table 5.19 Bi-script identification accuracies (%) at Word-level using MLP classifier, the upper triangular part of the matrix provides the results with Script dependent ($\boldsymbol{FS_{SD}}$) features and lower triangular part provides results with Script independent ($\boldsymbol{FS_{SI}}$) features

Script	Ben	Dev	Guj	Gur	Kan	Mal	Ory	Rom	Tam	Tel	Urd	Avg. Bi-script identification accuracy and standard deviation using $\boldsymbol{FS_{SD}}$ feature is **94.86% & 3.35**
Ben		95.66	99.58	97.25	96.08	96.25	97.58	98.00	98.91	97.82	95.66	
Dev	98.91		97.58	84.00	92.08	92.08	90.08	91.50	93.50	94.23	94.00	
Guj	99.41	99.00		98.00	96.41	98.50	97.50	98.33	97.66	99.16	98.66	
Gur	99.83	88.66	99.75		94.58	95.08	94.50	93.41	94.91	97.58	96.33	
Kan	99.50	100	99.83	100		85.33	92.33	91.91	92.66	93.23	94.25	
Mal	99.75	94.25	100	98.66	100		88.75	95.00	87.33	93.48	96.58	
Ory	97.41	96.16	98.16	96.58	100	97.91		95.08	90.25	94.23	97.41	
Rom	99.91	97.58	100	98.16	100	99.50	97.25		95.08	96.15	96.75	
Tam	99.75	93.41	98.41	97.75	100	94.50	94.66	99.33		95.32	97.58	
Tel	100	98.58	100	99.83	100	100	97.91	98.08	100		96.49	
Urd	99.66	96.33	99.50	96.66	99.91	96.00	98.16	99.25	98.33	100		
Avg. Bi-script identification accuracy and standard deviation using $\boldsymbol{FS_{SI}}$ feature is **98.40% & 2.16**												

Table 5.20 shows the summarization of the bi-script results as we obtained from four levels. Using script dependent feature, we have obtained accurecies of 98.93%, 98.81%, 99.24%, 94.84% at page, block, line and word-level accordingly. So, identification accuracy wise sequence is: line-level>page-level>block-level>word-level for script dependent features. Though line-level is the best performer, but close inspection reveals that, page and block-level performances are similar. The $\sigma_{\text{page-level}}$, $\sigma_{\text{block-level}}$, $\sigma_{\text{line-level}}$

and $\sigma_{word\text{-}level}$ values we have obtained as 1.79, 1.47, 1.28 and 3.35 accordingly. So, in terms of consistency of the results, we have found line, block and page-level results are more consistent compared to word-level. While considering the Script independent feature, the accuracies we have obtained are 97.00%, 97.57%, 99.01% and 98.40% at page, block, line and word-level. In this case: $\mu_{line\text{-}level}>\mu_{word\text{-}level}>\mu_{block\text{-}level}>\mu_{page\text{-}level}$. Again line-level shows the best performance among all. One interesting point is to be noted here, the word-level performance is more promising compared to page and block-level using Script independent feature. The bi-script performance graph is shown by Figure 5.8. Close inspection of this figure reveals that, bi-script performance is not exceptional compared to multi-script identification pattern as we explained in Section 3.5.1 (see Chpater 3). Finally we conclude that, in bi-script scenario, line-level shows best performance independent of the feature chosen, followed by page/block-level. Script independent feature are more suitable than script dependent feature in case of word-level data.

Table 5.20 Summary of the bi-script results in terms of μ and σ

Level of document image considered	Features and parameters			
	FS_{SD}		FS_{SI}	
	μ	σ	μ	σ
Page-level	98.93	1.79	97.00	4.01
Block-level	98.81	1.47	97.57	3.08
Line-level	99.24	1.28	99.01	1.65
Word-level	94.84	3.35	98.40	2.16

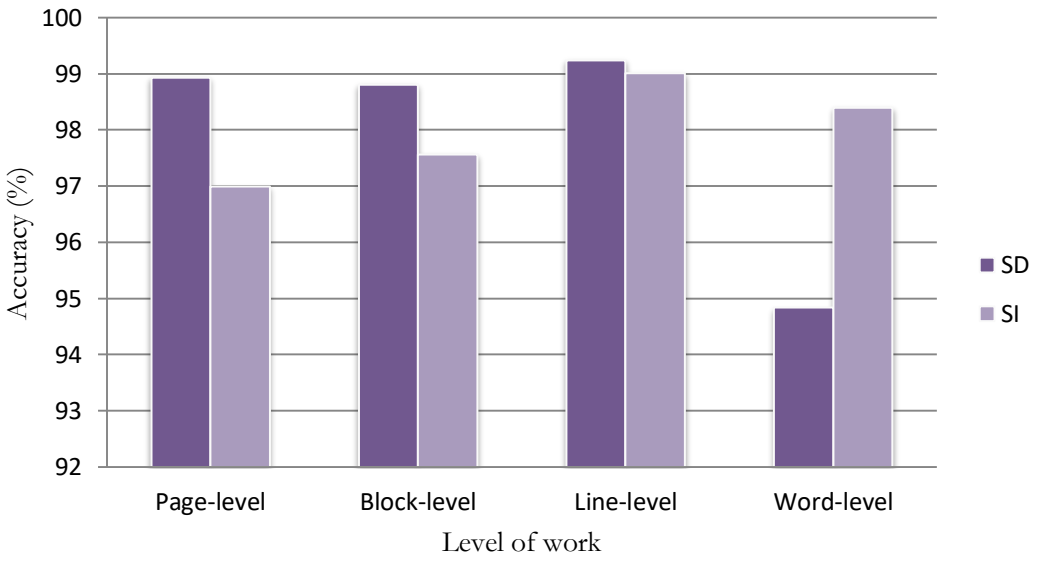


Figure 5.8 Comparison of bi-script performances at Page, Block, Line and Word-level performance using FS_{SD} and FS_{SI} features and MLP classifier

TRI-SCRIPT IDENTIFICATION

Devanagari and Roman are two most popular and widely use script in India. A good number of official Indian documents contain three different scripts, the state's official script and Devanagari, Roman. Thus the tri-script identification is become inevitable. In this experiment we have categorized among these triplates. In all these triplates, Devanagari and Roman are common, while the state's official scripts varies. Total nine such triplates are possible here: Devanagari, Roman common and any one of the nine other regional scripts. The numbers of tri-script experiments are as follows: 9 experiments using each of the $\boldsymbol{FS_{SD}}$ and $\boldsymbol{FS_{SI}}$ features, total 4 levels, so 9x2x4 =72 total experiments. The tri-script identification results are reported in Table 5.21. The μ and σ are the average and standard deviation of the identification rate. Figure 11 shows the comparison of identification accurecies of both the features $\boldsymbol{FS_{SD}}$ and $\boldsymbol{FS_{SI}}$. Using $\boldsymbol{FS_{SD}}$ features the page, block, line and word-level average identification accurecies were reported to be 98.42%, 98.50%, 98.70% and 88.46% respectively. So, in tri-script

scenario, $\boldsymbol{FS_{SD}}$ feature is consistant for page, block and line-level. On the other hand, the word-level accuracy drops at same rate for tri-script identification too, as we observed in multi-script and bi-scripts scenarios. While considering $\boldsymbol{FS_{SI}}$ features, we received the average identification accurecies of 96.76%, 95.83%, 99.03% and 96.60% for page, block, line and word-level respectively. From performance graph of Figure 5.9, we found that the performances of line-level data are more consistent irrespective of the feature choosen. In our experiment, we have received the average line-level identification accuracies of 98.70% and 99.03% respectively for $\boldsymbol{FS_{SD}}$ and $\boldsymbol{FS_{SI}}$ features, so it proves the feature independence. On the other hand, at word-level there is a huge performance drop of 8.14% from $\boldsymbol{FS_{SI}}$ feature to $\boldsymbol{FS_{SD}}$ feature. So, the performance of word-level data are more dependent on the particular feature used, making the task more challenging.

Table 5.21 The tri-script identification performance at page, block, line and word-level using MLP classifier

Script (with Dev-Rom)	Document level & Feature							
	Page-level		Block-level		Line-level		Word-level	
	FS_{SD}	FS_{SI}	FS_{SD}	FS_{SI}	FS_{SD}	FS_{SI}	FS_{SD}	FS_{SI}
Ben	100	99.16	97.66	95.83	99.66	99.88	91.11	98.33
Guj	100	100	99.66	99.33	99.66	99.88	92.33	98.38
Gur	95.00	91.66	95.50	94.00	93.22	98.66	82.27	94.11
Kan	96.66	98.33	98.33	98.83	99.22	99.77	86.44	98.61
Mal	99.16	97.50	98.33	96.66	99.44	99.77	87.66	94.83
Ory	97.50	96.66	99.16	94.33	99.11	99.00	86.61	95.72
Tam	98.33	96.66	99.50	92.00	99.33	97.00	88.44	94.77
Tel	100	98.33	99.83	95.66	99.11	98.33	90.99	98.11
Urd	99.16	92.50	98.50	95.83	99.55	99.00	90.33	96.50
μ	98.42	96.76	98.50	95.83	98.70	99.03	88.46	96.60
σ	1.74	2.87	1.34	2.30	2.07	0.95	3.13	1.80

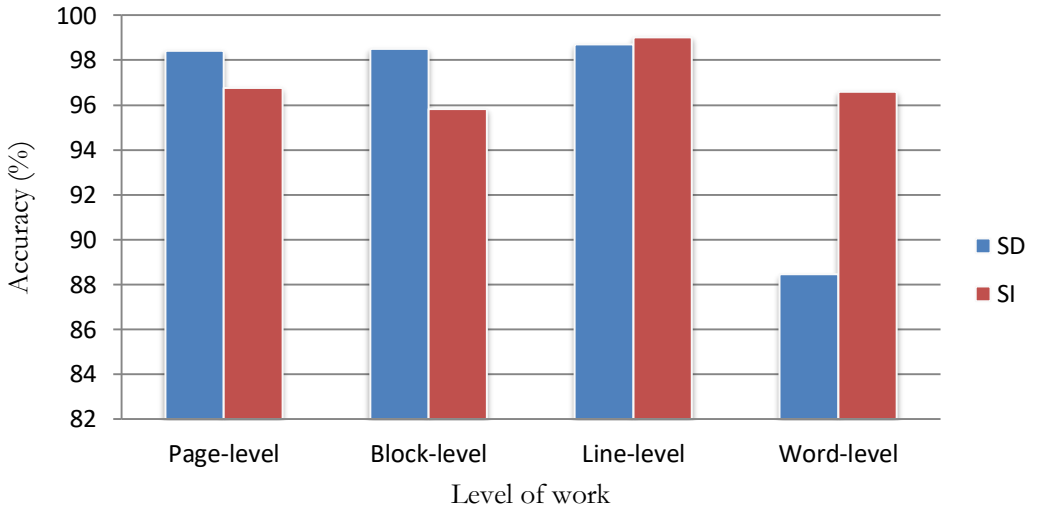


Figure 5.9 Comparison of tri-script performances at Page, Block, Line and Word-level performance using $\boldsymbol{FS_{SD}}$ and $\boldsymbol{FS_{SI}}$ features and MLP classifier

SUMMARY OF THE RESULTS

Table 5.22 shows the summarization of script identification results at page, block, line and word-level for $\boldsymbol{FS_{SD}}$ & $\boldsymbol{FS_{SI}}$ features and MLP classifier. From our experimental results, we can draw the following conclusions:

- Script identification from all official Indic scripts has been carried out in this work. In addition with that, we propose a novel multi-level script identification framework.
- Among the four levels (i.e. page, block, line and word), line-level performance is more consistent irrespective of the features (SD or SI) or identification scenarios (i.e. multi-script, bi-script or tri-script). We got the lowest value of σ_{line} as 2.86 and the highest value of σ_{word} as 10.19.
- While comparing among the script identification scenarios (i.e. multi-script, bi-script or tri-script), Bi-script performance is more consistent compared to other two. In our experiment we found, $\sigma_{bi\text{-}script}$ as 2.08 and 0.88 for $\boldsymbol{FS_{SD}}$ and $\boldsymbol{FS_{SI}}$

respectively. The highest value we got for $\sigma_{\text{multi-script}}$ as 11.74 and 4.20 for $\boldsymbol{FS_{SD}}$ and $\boldsymbol{FS_{SI}}$ respectively.

- While comparing the two features (i.e. $\boldsymbol{FS_{SD}}$ and $\boldsymbol{FS_{SI}}$), we found that script dependent features are more suitable at page, block and line level, Whereas script independent features perform comparably well at word-level.

Table 5.22 Summary of the page, block, line and word-level script identification results, feature: FS_{SD} & FS_{SI}, Classifier: MLP

Level vs. Script identification	Multi-script		Bi-script		Tri-script	
	FS_{SD}	FS_{SI}	FS_{SD}	FS_{SI}	FS_{SD}	FS_{SI}
Page-level	94.32	83.64	98.93	97.00	98.42	96.76
Block-level	94.05	87.32	98.81	97.57	98.50	95.83
Line-level	93.73	93.19	99.24	99.01	98.70	99.03
Word-level	70.54	91.04	94.84	98.40	88.46	96.60

5.1.3 NUMERAL SCRIPT IDENTIFICATION

All the works available in literature are mainly based on script identification on alphabetic characters. Till date, very few works have been reported on *HNSI* (Handwritten Numeral Script Identification) and no work considering four numeral Indic scripts, which inspired us to carry out the present work [86]. *HNSI* has its applicability in different domains of 'smart computing' like automatic sorting of postal documents based on PIN code script, automatic classification of application forms, examination forms etc. written by native languages based on a numeral strings. The present work proposes an intelligent *NSI* technique from handwritten document images written by any one of the four popular Indic scripts, namely *Bangla, Devanagari, Roman and Urdu*. We have also applied different feature combinations along with different script combinations to make the system more robust. The experiment was carried out on *Numeral_db* dataset which is discussed in Chapter 2.

Design of feature set

During feature extraction, first, visual observations are made on Indic scripts to study the nature of different graphemes of different scripts. The main features considered are based on spatial and wavelet energy. The features considered for this work are as follows:

- $\boldsymbol{FS_{SE}}$ **–** Feature based on spatial energy. Feature dimension is 04 [Refer Chapter 3, Table 3.2]
- $\boldsymbol{FS_{WE}\#2}$ **–** Feature based on wavelet energy. The feature dimension is 51 [Refer Chapter 3, Table 3.2]

Final feature set for present page-level script identification problem:

$$\boldsymbol{FS_{SE \cup WE\#2} = FS_{SE} \cup FS_{WE}\#2}$$

= 55 dimensions

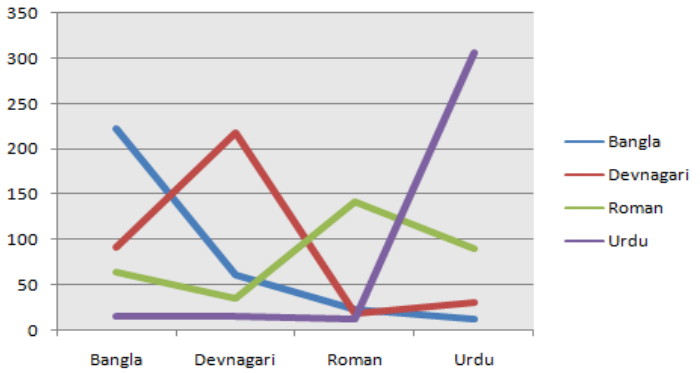
Among the classifiers we considered: NBTree, PART, LIBLinear, Random Forest, SMO, Simple Logistic and MLP. *HNSI* from document images is a challenging task because number of samples in each script are limited to ten (zero to nine) and many samples are common to each other for different script pairs. We have statically analysed this similarity percentage from real life handwritten numeral data. It has been found that Bangla-Devanagari group has almost 40% numerals which are visually and structurally similar. So here, total number of numeral samples of these two scripts reduces to 16 instead of 20. This fact leads to higher misclassification rate between these two scripts. In a similar manner, Bangla-Roman, Devanagari-Roman and Roman-Urdu script pairs have digit similarity percentage of 30%, 40% and 20% respectively. It has been found that only Urdu script characters are almost distinct in visually and structurally from other three scripts. So whenever there is a pair of Urdu and any other script then reasonable outcome has been found. The effect of the similarity percentage among different scripts has direct impact on the identification rate. That is why numeral

script identification from handwritten document images is a real challenge in terms of successful identification rate, and is still far from complete solution.

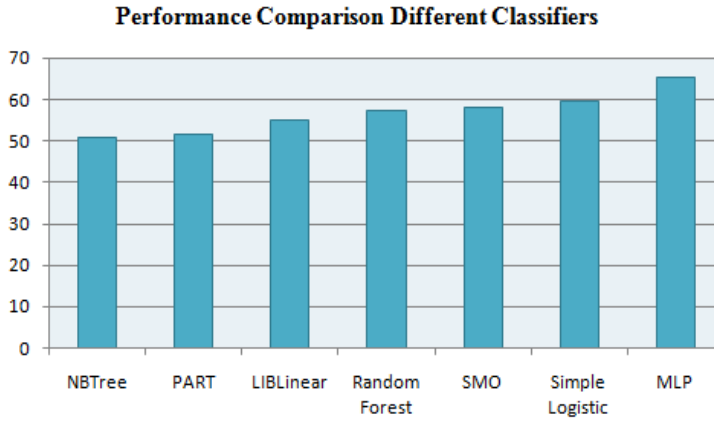
Extensive experimentation has been carried out for the present work. Performances of bi-scripts, tri-scripts, four-scripts combinations are measured. Table 5.23 shows confusion matrix of four-scripts on the test dataset. The last row shows four-scripts identification rate which is 65.4% for the present experiment. Maximum misclassification is found among Bangla, Devanagari and Roman scripts. This is due to the similar shaped digits of those scripts as mentioned in the introduction section. As per our expectation, we got encouraging identification performance for Urdu script. This is because except the numeral 'one' in Urdu which is very much similar with numeral 'one' in Roman, all other digits are visually and structurally distinct in comparison with other three. The evidence of our observation can be seen from misclassification rate of Urdu script, which is almost equal with other three scripts (Bangla: 4.2%, Devanagari: 4.2% and Roman: 3.4%). Figure 5.10 (a) shows graphical representation of Table 5.23, whereas the comparative graph of seven classifiers for four-scripts average accuracy rate is shown by Figure 5.10 (b). The performances of tri-scripts and bi-scripts combinations can be found from Table 5.24 and Table 5.25 respectively. From Table Table 5.24, average tri-scripts accuracy rate of 71.8% can be found for ${}^4C_3$ or four different combinations. Highest accuracy rate is reported by the {Bangla, Roman, Urdu} combination which is 2.8% more than tri-scripts average rate. Whereas the tri-script combination of {Bangla, Devanagari, Roman} reports the lowest accuracy rate among all which is 3.6% below the average rate. Among the ${}^4C_2$ or six bi-scripts combinations, as shown in Table 5.25, highest accuracy is found for the script combination {Bangla, Urdu} (90.9%) combination, followed by {Devanagari, Urdu} (89.6%) and {Roman, Urdu} (82.1%). Lowest bi-script accuracy rate is found for {Bangla, Devanagari} script combination which is 10.4% below the average bi-scripts accuracy rate (82.2%). The pattern of the result is also similar here in comparison with Table 5.23 and Table 5.24.

Table 5.23 Confusion matrix on the test dataset after splitting the whole dataset into 2:1 training and testing set ratio

Classified As	Bangla	Devanagari	Roman	Urdu
Bangla	222	62	23	13
Devanagari	91	218	18	30
Roman	65	36	142	91
Urdu	15	15	12	307
Average four-scripts identification Rate: 65.4%				



(a)



(b)

Figure 5.10 (a) The graphical representation of the confusion matrix on the test dataset using MLP; (b) Performance comparison of seven different classifiers by Average Accuracy Rate (%) measured using True Positive Values.

Table 5.24 Tri-script identification rate using MLP classifier on the test dataset of $^4C_3$ sets.

Sl. No.	Scripts Combination	AAR (%)
1	{Bangla, Roman, Urdu}	74.6
2	{Devanagari, Roman, Urdu}	73
3	{Bangla, Devanagari, Urdu}	71.4
4	{Bangla, Devanagari, Roman}	68.2
5	***Avg. tri-script Acc. Rate***	***71.8***

Table 5.25 Bi-script identification rate using MLP classifier on the test dataset of $^4C_2$ sets.

Sl. No.	Scripts Combination	AAR (%)
1	{Bangla, Urdu}	90.9
2	{Devanagari, Urdu}	89.6
3	{Roman, Urdu}	82.1
4	{Bangla, Roman}	80.2
5	{Devanagari, Roman}	78.6
6	{Bangla, Devanagari}	71.8
7	***Avg. bi-script Acc. Rate***	***82.2***

Exact comparative study with the work of other fellow researches is not possible right now as no work is reported on *HNSI* considering Bangla, Devanagari, Roman and Urdu scripts. Availability of benchmark database is another problem in this field, that's why we have taken the effort to prepare our own image dataset. The present result can be considered as a benchmark for these Bi-scripts, Tri-scripts and Four-script combinations.

Error Analysis:

It is already discussed that handwritten script identification is more challenging compared to printed one. This is because, besides the inherent problems of some visual and structural smililary, due to the varying writing patterns sometimes components from different scripts look quite similar. For the experiment considered in Section 5.1.1, page level script identification from eleven official scripts (results shown in Table 5.7), we found out of 220 Devanagari pages, 03 are misclassified with Bangla and 03 are

misclassified with Gurumukhi. This is because Devanagari script shares most visual similarity with these two scripts. From the same table we found that, 02 Oriya scripts are misclassified with Bangla, and 01 each with the Devanagari, Kannada and Malayalam. This is due to the structural similarity of the Oriya charactes with these scrits. In general Oriya characters are rounded in shape, which is similar with few of the characters of Bangla, Devangari, Kannada and Malayalam characters.

In another experiment where we have done the script identification at multiple levels in Section 5.1.2, we found that, misclassification rate varies with the features considered. Table 5.13 show the script identification results using script dependent features where, we found Devanagari are mostly misclassified with Gurumukhi, Gurumukhi are misclassified with Devanagari, Oriya are misclassified with Bangla. The results are almost comparable at Page, block and line level. But most misclassification occurs at word level. Still the pattern of misclassification is same, i.e. most of the Devanagari words are misclassified with Gurumukhi. Table 5.14 shows the script identification results using script dependent features. Here we found the most misclassified instances for Urdu script, 04 Urdu pages are misclassified with Oriya. So, we can conclude that the misclassified patterns and count is many times directly dependent on the feature set choosen.

From the result of numeric script identification as found in Table 5.23, Roman is the mostly misclassified numeral scripts with a maximum misclassification instances with Urdu followed Bangla and Devanagari. This is due to the limited digit set in numeral domain (ten only) and out of this set many digits looks alike among these scripts. Four numeral digits of Roman are almost structurally and visually similar with Bangla and Devanagari script. For Urdu we found maximum accuracy because only one Urdu numeral digit looks similar with other three.

Figure 5.11 shows some sample images depicting the possible cause of misclassification among various Indiac scripts.

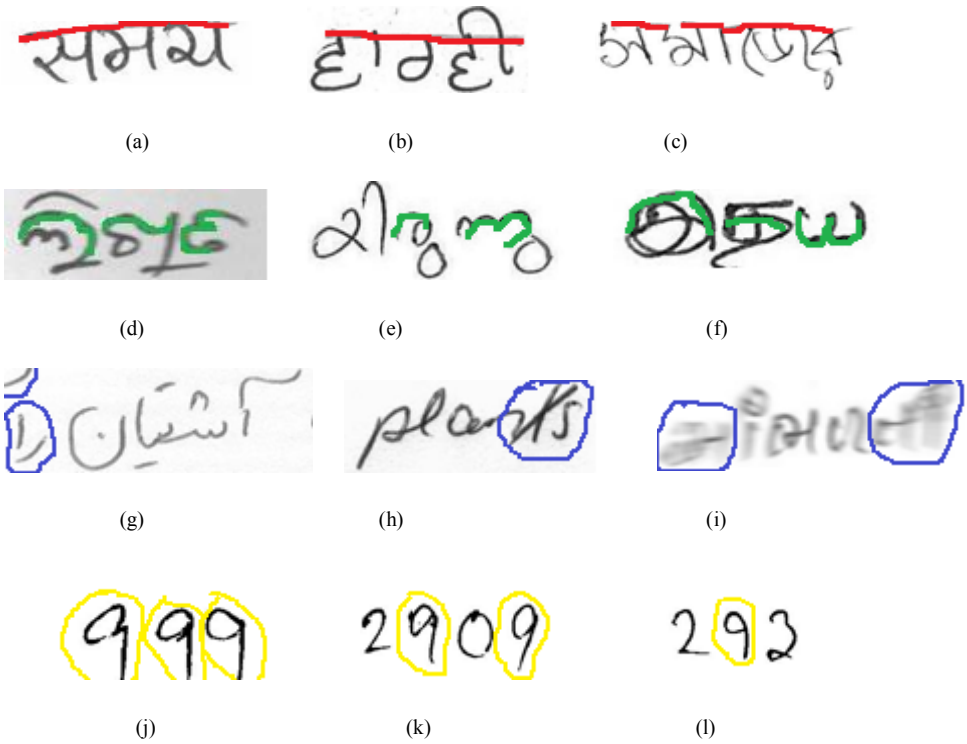


Figure 5.11 Sample images which show the possible cause of misclassification (a-c) Devanagari, Gurumukhi and Bangla scripts bearing similar 'matra' like component, (d-f) Oriya, Malayalam and Tamil words, they share quite similar visual shape (f-h) sample noisy images, (g) improper segmented Urdu word, (g) complex writing of Roman word look it like other script, (h) Gujarati noisy word due to blur, (j-l) sample numeral images from Urdu, Roman and Devanagari though they looks alike in many characters

5.2 CONCLUSION

Script identification from handwritten document is a challenging task due to several factors: different writing styles from people of diversified cultures across India, asymmetric nature of handwritten characters compared to printed ones, presence of skew at word, line or document-level, the presence of very similar characters within a single word even from a single writer, and different spacing between different words, lines and characters in handwritten document. In this chapter, we have discussed about handwritten script identification at various levels with different number of scripts. Page-level handwritten script identification from 11 official Indic scripts is discussed. Block, line and word-level script identification using different numbers of scripts is discussed. The idea of 'matra' based script separation using some low dimensional feature set is presented. We proposed the idea of multi-level script identification where

the same document is considered at page, line, block and word-level. Then the multi-script identification performance is studied for all the levels for different categories of features. Suitability of a particular level (i.e. page/line/block/word) of script identification and the feature which performs optimally at that level is analysed. Finally the importance and results of numeral script identification is discussed.

Overall in this chapter, we have addressed different aspects of handwritten script identification starting from traditional frameworks to some upcoming script identification problems like: numeral script identification, multi-level script identification. As we know the unavailability of standard dataset in this field, so all the results we obtained can be considered as benchmark on the used dataset.

CHAPTER SIX

CONCLUSION

6.1 CONTRIBUTION OF THE THESIS

The work presented in this thesis addresses various problems related to Indic script identification. To be more specific, in this endeavour, we have carried out the following tasks:

- Statistical analysis of different Indic languages and scripts with their demographic distribution
- Survey of Indic script identification techniques with their limitations
- Preparing handwritten script dataset for 11 Indic scripts which actually cover all the Indic languages, preparing handwritten numeral scripts dataset
- Printed and handwritten script identification from 11 official scripts with the analysis of bi-script, tri-script and multi-script performance
- Handwritten numeral script identification
- Study of the effect of document segmentation for script identification performance.

The statistical analysis of different Indic languages and scripts with their demographic distribution has been presented in Section 1.2 of Chapter 1. As shown in Table 1.1, there are 23 official languages (including English) in India as per 8th schedule of the Indian constitution. These 23 languages are written using 11 different scripts, which means that, there are many languages which are written by a single script. Examples of such languages are Bangla (used to write Bengali, Assamese and Manipuri languages),

Devanagari (used to write Hindi, Sanskrit, Nepali languages). Demographically speaking, Roman is the most popular script, followed by Devanagri and Bangla.

Some of the intrinsic properties of Indic scripts are as follows:

- Scripts like Bangla and Devanagari contain a special topological property known as 'matra' or 'shirorekha'.
- Oriya and Malayalam scripts have components of a more circular shape than others.
- Considering Tamil script, most of the characters contain a 'T' like shape in their structure.
- Urdu script have maximum dot ('.') like small components. This script looks quite unlike other Indic scripts. Many characters of Urdu contain directional strokes with an orientation of around 75^0.
- There are many vertical, horizontal and slanting (45^0) strokes in Roman script.
- Kannada and Telugu scripts are quite similar, except a 'tick' like symbol present in Telugu script which is not there in Kannada. Similarly, Tamil and Malayalam characters are very much similar. Tamil and Malayalam characters have downward concavities and Kannada and Telugu characters have upward concavities as shown in Figure 1.7 of Chapter 1.

The writing system of India follows an alphabetic writing system, which is divided into three main categories: abjad, abugida and true alphabetic. Urdu script, which has its origin in the Indo-European family falls under the abjad catagory. The Roman script follows true alphabetic system. The rest of the nine scripts belong to the Brahmic family of scripts, which fall under the abugida catagory. Brahmic family of scripts is divided into three classes: gupta, kadamba and grantha. All the eastern and northern Indian scripts are from gupta family. The four main south Indian scripts belong to the kadamba and grantha family.

In Section 1.3 of Chapter 1, we have presented a survey of handwritten script identification techniques. The flow diagram has been presented in Figure 1.9, which shows different offline script identification techniques at different levels: page, block, line, word and character. Table 1.3 summarizes the offline script identification from handwritten or handwritten-printed mixed type documents. In Table 1.4, level wise distribution of all these works has been reported. It is found that, most of the works has been carried out at word and block level. Very few works are at page, line and character level. Motivated by this fact, we have prepared a complete 11-script page-level handwritten dataset and performd the script identification task. At line-level, initially, we proposed a script identification technique from eight official scripts. Later during multi-level script identification, we performed script identification from all official Indic scripts at four major levels: page, block, line and word.

Availability of standard dataset for all official Indic scripts has been a real challenge for the script identification work. The issue of dataset development for script identification has been discussed in Chapter 2. In Section 2.2, State-of-the-art techniques on handwritten dataset development considering Indic scripts has been discussed and the summary has been shown in Table 2.1. It can be deduced that, till date, handwritten Indic scripts dataset development has been restricted to a maximum of three scripts. This dataset is known as PBOK dataset, which consists of a total four scripts: Persian, Bangla, Oriya and Kannada, out of which last three are Indic scripts. This information has been presented in Table 2.4. Table 2.2 shows the global demographic distribution of different Indic scripts. We found that, the Indic scripts considered in this thesis not only concerns of India but for researchers outside India too. In Section 2.4, we discussed about the proposed dataset as a part of this thesis work. Although we have collected and used different printed/ handwritten datasets throughout this work, our main contribution in this chapter has been three handwritten datasets: (i) *PHDIndic_11* : a complete page-level handwritten dataset from 11 official Indic scripts (ii) word-level printed dataset from 13 different languages and 11 official scripts and (ii) *Numeral_db*: a handwritten numeral script dataset from four most popular Indic scripts. Section 2.4.1 presents PHDIndic_11 dataset, which consists of total 1458 pages from 11 different

scripts written by 463 different writers and distributed over approximately 15010 lines and 124279 words. It was collected over duration of more than two years from different parts across the country. From Figures 2.2 to 2.12, two sample images from each of scripts have been shown. Finally, in Table 2.4, we have compared the proposed dataset with few of existing ones and illustrated the effectiveness of the proposed one. In Section 2.4.2, we have proposed a word-level printed document image dataset from 13 different languages and 11 different scripts. The dataset consists of total 39k words, 3k words from each language. The script identification result on this dataset has been discussed in Chapter 4. In Section 2.4.3, we proposed the *Numeral_db* dataset, which is a handwritten numeral script dataset from four popular Indic scripts: Bangla, Devanagari, Roman and Urdu. This dataset consists of 5659 numeral strings written by 43 different writers. A comparative analysis of *Numeral_db* with other state-of-the-art numeral image datasets has been shown in Table 2.6. The number of scripts covered by *Numeral_d*b dataset is 75% more compared to existing numeral datasets. Additionally, we also proposed benchmark results for script identification on these datasets. These results have been reported in Chapter 5.

Different methods used in the present work have been discussed in Chapter 3. Ghosh et al. [8] reported that, no universal feature exists which can effectively classify all the Indic scripts. Features are in general script/application dependent. Hence, for optimum performance, there might arise a need to combine different features through a heuristic feature selection approach. In Section 3.1, different feature extraction techniques used in this thesis have been discussed. First, we studied the shape and structural property of different scripts. Then based on our observation, we computed different structural features: number of small components, presence of directional strokes, circularity, rectangularity and convexity of connected components, topological property etc. Here, one of our major contributions is optimizing the dimension of one of the topological feature, i.e. proposing a 1-dimensional fractal dimension (only one attribute is considered in this feature). During the work of 'matra' and without 'matra' separation, we compared the proposed 1-dimensional fractal dimension with two of the state-of-

the-art techniques: canny edge detector and line transform. The effectiveness of the proposed features has been supported by experimental results as shown in Chapter 5. Fractal dimensions are used in handwritten script identification along with other features to obtain promising results. Another important contribution is the directional stroke based feature as discussed in Section 3.1.1.3. Observing the presence of different directional strokes, we have defined four directional kernels, and feature values are computed applying different morphological operators. In Section 3.1.2, we have described different script independent features. Some state-of-the-art texture features namely: gray-level co-occurrence matrix, gabor filter bank, spatial energy, wavelet energy and radon transform have been studied. One of our contributions is optimizing the performance of wavelet features by making a feature fusion with radon transform i.e. we proposed wavelet radon transform or WRT. Experimental results show the effectiveness of WRT in compared to normal wavelet transform. Another feature fusion based technique is used based oninterpolated morphological transform or IMT. It is a fusion of interpolation and morphological operations. In Chapter 4, we discussed the outcome of printed script identification. The present literature suggests that, most of works have been carried out on printed documents [8]. This is due to less complexity of printed documents in comparison to handwritten one. In Section 4.2, two different printed script identification problems have been addressed. In the first one, we have carried out page-level script identification from eleven official Indic scripts. As no page-level printed dataset were available, we conducted the experiment on our collected dataset. Mainly structural features or shape based features are used in this work. Performance of different classifiers are compared and random forest classifier has been found to be the best performer with an average multi-script identification accuracy of 98.99%, followed by LibLINEAR and MLP with 98.19% and 98.00 % respectively. This result can be considered as a benchmark on the dataset used in this work. In another problem as described in Section 4.2.2, we have discussed the word-level script identification from eleven official Indic scripts (number of languages considered are thirteen). A dataset of volume 39k words, with equal distribution for each of the languages had been considered for the purpose of experiment. Three different features: spatial energy, wavelet energy and radon transform, three state-of-the-art classifiers:

MLP, FURIA and random forests were used here. Two bi-script scenarios: (i) keeping Roman common with other languages (ii) keeping Devanagari common with other languages were considered. In scenario (i) we received an average accuracy of 98.38% using MLP, while in scenario (ii) 99.24% average accuracy was obtained using the same classifier. During tri-script identification (keeping both Roman and Devanagri common), we have obtained an average accuracy of 98.19% using MLP.

In Chapter 5, we have discussed about the handwritten script identification. The notable work reported in this chapter are: (i) Block-level script identification from six official Indic scripts (ii) Line-level script identification from eight official Indic scripts (iii) Page-level script identification from eleven official Indic scripts (iv) Numeral script identification from four popular Indic scripts (v) Script separation of 'matra' based scripts from scripts without 'matra' and (vi) Multi-level handwritten script identification from all official Indic script. Numeral script identification is a new direction of work in this field as it will help in different applications like: sorting of postal documents, arranging multi-lingual application forms or examination sheets based on the roll number written in candidates own scripts. We conducted the experiment to separate scripts with 'matra' from scripts without 'matra' and used it as a precursor for script identification. Reduced 1-dimenstional fractal dimension has been used as the sole pertinent feature in this work. Finally, we performed multi-level script identification from all the eleven official Indic scripts. In the literature, all the works had been carried out only at a single level. There is no theoretical or experimental justification available till date about choosing a particular level of work. So, here we have prepared a multi-level dataset, i.e. the same document has been considered at page, line, block and word level. Then, two different types of features: script dependent (structural) and script independent (global texture) have been applied at each level for script identification. So, in this work our objective is not only to study about the effect of segmentation on the performance of script identification but also to analyze which types of features are suitable at which level. Our observations are as follows: (i) line level data are more consistent irrespective of the features chosen. Block and page level data are

comparatively similar and performance of word level data is very much feature dependent. (ii) Suitable feature combination has a remarkable effect on the overall performance of script identification (iii) Word level script identification is more challenging in terms of accuracy compared to page, line and block level identification. Hence, here we have attempted to provide an experimental justification for choosing a particular level of work along with suitability of different features at different level of work. We feel that, this is a new direction for future script identification work.

6.2 SCOPE OF THE FUTURE WORK

The work reported in this thesis can be further extended in several directions for future research. These have been pointed out below:

- **ONLINE SCRIPT IDENTIFICATION:**

Online script identification system handles real time handwritten data and processes them for identification. As compared to offline script identification techniques, much work has not been reported in online environment particularly in the handwritten domain. So, this area can be further explored by the researcher.

- **DYNAMIC SCRIPT IDENTIFICATION:**

Video based script identification has several applications like: automatic content based information retrieval, indexing and searching. Video texts normally vary in size, orientation and pace making the identification task more challenging compared to normal offline or online text images. Figure 6.1 shows few sample video frame images from our dataset.

Figure 6.1 Sample multi-script video frame image

- **SCRIPT IDENTIFICATION FROM THE SCENE IMAGES:**

Text detection from scene images is one of the recent areas of interests among researchers. It has several applications like: tracking license plate from moving vehicles, development of driver less automatic vehicles, building software for a blind person for freely walking on the road, building biometric devices, extracting GPS information from Google map etc.

- **MOBILE/ HANDHELD DEVICE BASED SCRIPT IDENTIFICATION:**

Mobile or other handheld devices have very less computational resources compared to a PC based system. Developing algorithms for this type of application have serious computational recourse restrictions. Language or script identification from images captured using these devices is a very challenging task, and is an area of research interest in the near future.

- **SCRIPT IDENTIFICATION FROM ARTISTIC DOCUMENTS:**

Figure 6.2 shows sample artistic document images where multi-script occurs at character level. Finding scripts from these images is a challenging task due to segmentation problem, complex background, and uneven contrast information.

Figure 6.2 Different words showing multiple scripts at character level

- **SCRIPT IDENTIFICATION FROM DEGRADED/NOISY DOCUMENTS:**

Degraded/noisy document processing is one of the most challenging issues [112]. In future, we are also interested to explore the possibility of script identification from different kind of degraded/noisy historical documents, manuscripts etc.

- **MULTI-SCRIPT SIGNATURE IDENTIFICATION:**

Signature recognition is a behaveoural biometric that tries to identify a person uniquely from his/her signature [113] [114]. As India is a multi-script country, the scope of signature identification has broadened into mult-script scenario. Not much works have been reported on this. So, scopes are there to work in this area.

- **OPTIMIZING SCRIPT IDENTIFICATION PERFORMANCE:**

In future, we plan to work on optimization issues using nature inspired algorithm and soft computing paradigms [115] [116] for boost up the overall performance of script identification particularly at word-level.

The work presented in this thesis can be considered as an important step towards automation of document processing in the multi-script scenario. Here, the author proposes a pre-processing step (i.e. script identification) before supplying the document to script specific OCR. Some key issues of the script identification have been discussed with special emphasis on handwritten script identification. One of the important outcomes of this thesis is presenting different frameworks and techniques for script identification, as well as, the development of benchmark handwritten datasets.

REFERENCES

[1] J. Mantas, "An overview of Character Recognition Methodologies," *Pattern Recognit.*, vol. 19, pp. 425–430, 1986.

[2] "OCR System: A Literature Survey." [Online]. Available: http://shodhganga.inflibnet.ac.in/bitstream/10603/4166/10/10_chapter 2.pdf.

[3] "Eight_Schedule." [Online]. Available: http://mha.nic.in/hindi/sites/upload_files/mhahindi/files/pdf/Eighth_Schedule.pdf.

[4] R. Sarkar, N. Das, S. Basu, M. Kundu, M. Nasipuri, and D. K. Basu, "CMATERdb1: a database of unconstrained handwritten Bangla and Bangla–English mixed script document image," *Int. J. Doc. Anal. Recognit.*, vol. 15, no. 1, pp. 71–83, 2012.

[5] K. Roy and U. Pal, "Word-wise Hand-written Script Separation for Indian Postal Automation," in *10th International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, 2006, pp. 521–526.

[6] "Map_States," 2016. [Online]. Available: http://commons.wikimedia.org/wiki/File:States_of_South_Asia.png.

[7] S. M. Obaidullah, S. K. Das, and K. Roy, "A System for Handwritten Script Identification from Indian Document," *J. Pattern Recognit. Res.*, vol. 8, no. 1, pp. 1–12, 2013.

[8] D. Ghosh, T. Dube, and S. P. Shivprasad, "Script Recognition – A Review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2142–2161, 2010.

[9] "Writting System of India." [Online]. Available: http://en.wikipedia.org/wiki/Writing_system.

[10] "Abugida Writing System," 2016. [Online]. Available: available: http://en.wikipedia.org/wiki/Abugida.

[11] M. Paul, "Ethnologue: Languages of the World," Dallas: SIL International, 2009.

[12] S. M. Obaidullah, C. Goswami, K. C. Santosh, C. Halder, N. Das, and K. Roy, "Separating Indic scripts with `matra' -- a precursor to script identification in multi-script documents," in *IAPR International Conference on Computer Vision & Image Processing*, 2016, p. In Press.

[13] J. J. Lee, J. H. Kim, and M. Nakajima, "A Hierarchical HMM Network-based Approach for Online Recognition of Multi-lingual Cursive Handwritings," *Inst. Electron. Inf. Commun. Eng. Trans. Inf. Syst.*, vol. E81–D(8), pp. 881–888, 1998.

[14] R. Rani, R. Dhir, and G. S. Lehal, "Script Identification for Pre-segmented Multi-font Characters and Digits," in *12th International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 2010–1154.

[15] C. V Lakshmi and C. Patvardhan, "An optical character recognition system for printed Telugu text," *Pattern Anal. Appl.*, vol. 7, no. 2, pp. 190–204, 2004.

[16] Zhu, X. Yu, Y. Li, and D. Doermann, "Language Identification for Handwritten Document Images Using A Shape Codebook," *Pattern Recognit.*, vol. 42, pp. 3184–3191, 2009.

[17] S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri, and D. K. Basu, "A Novel Framework for Automatic Sorting of Postal Documents with Multi-script Address Blocks," *Pattern Recognit.*, vol. 43, no. 10, pp. 3507–3521, 2010.

[18] V. Singhal, N. Navin, and D. Ghosh, "Script-based Classification of Hand-written Text Documents in a Multi-lingual Environment," in *13th International Workshop on Research Issues in Data Engineering: Multi-lingual Information Management*, 2003, pp. 47–54.

[19] M. Hangarge and B. V. Dhandra, "Offline Handwritten Script Identification in Document Images," *Int. J. Comput. Appl.*, vol. 4, no. 6, pp. 6–10, 2010.

[20] G. Rajput and H. B. Anita, "Handwritten Script Recognition using DCT and Wavelet Features at Block Level," *Int. J. Comput. Appl. Spec. Issue Recent Trends Image Process. Pattern Recognit.*, vol. 3, pp. 158–163, 2010.

[21] K. Roy, A. Banerjee, and U. Pal, "A System for Word Wise Handwritten Script Identification for Indian Postal Automation," in *IEEE India Annual Conference*,

2004, pp. 266–271.

[22] R. Sarkar, N. Das, S. Basu, M. Kundu, M. Nasipuri, and D. K. Basu, "Word level Script Identification from Bangla and Devanagri Handwritten Texts Mixed with Roman Script," *J. Comput.*, vol. 2, no. 2, pp. 103–108, 2010.

[23] S. Chanda, K. Franke, and U. Pal, "Identification of Indic Scripts on Torn-Documents," in *International Conference on Document Analysis and Recognition*, 2011, pp. 713–717.

[24] M. Hangarge, K. C. Santosh, and R. Pardeshi, "Directional discrete cosine transform for handwritten script identification," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2013, pp. 344–348.

[25] P. K. Singh, R. Sarkar, N. Das, S. Basu, and M. Nasipuri, "Identification of Devnagari and Roman Scripts from Multi-script Handwritten Documents," in *5th International Conference Pattern Recognition and Machine Intelligence*, 2013, pp. 509–514.

[26] R. Pardeshi, B. B. Chaudhuri, M. Hangarge, and K. C. Santosh, "Automatic Handwritten Indian Scripts Identification," in *2014 14th International Conference on Frontiers in Handwriting Recognition*, 2014, pp. 375–380.

[27] P. K. Singh, I. Chatterjee, and R. Sarkar, "Page-level handwritten script identification using modified log-Gabor filter based features," in *IEEE 2nd International Conference on Recent Trends in Information Systems*, 2015, pp. 225–230.

[28] P. K. Singh, R. Sarkar, M. Nasipuri, and D. Doermann, "Word-level script identification for handwritten Indic scripts," in *13th International Conference on Document Analysis and Recognition*, 2015, pp. 1106–1110.

[29] S. Kanoun, A. Ennaji, Y. L. Courtier, and A. M. Alimi, "Script and Nature Differentiation for Arabic and Latin Text Images," in *8th International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, 2002, pp. 309–313.

[30] L. Zhou, Y. Lu, and C. L. Tan, "Bangla/English Script Identification Based on Analysis of Connected Component Profiles," in *2nd International Workshop on Document Analysis Systems*, 2006, pp. 243–254.

[31] K. Roy, U. Pal, and B. B. Chaudhuri, "Neural Network based Word-wise Handwritten Script Identification System for Indian Postal Automation," in

International Conference on Intelligent Sensing and Information Processing, 2005, pp. 240–245.

[32] J. Hochberg, K. Bowers, M. Cannon, and P. Kelly, "Script and Language Identification for Handwritten Document Images," *Int. J. Doc. Anal. Recognit.*, vol. 2, no. (2/3), pp. 45–52, 1999.

[33] M. Benjelil, S. Kanoun, R. Mullot, and A. M. Alimi, "Arabic and Latin Script Identification in Printed and Handwritten Types Based on Steerable Pyramid Features," in *Steerable Pyramid Features, International Conference on Document Analysis and Recognition (ICDAR)*, 2009, pp. 591–595.

[34] K. Roy, A. Alaei, and U. Pal, "Word-Wise Handwritten Persian and Roman Script Identification," in *12th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2010, pp. 628–633.

[35] Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–697, 1986.

[36] S. B. Moussa, A. Zahour, A. Benabdelhafid, and A. M. Alimi, "Fractal-Based System for Arabic/Latin, Printed/Handwritten Script Identification," in *International Conference on Pattern Recognition*, 2008, pp. 1–4.

[37] G. G. Rajput and H. B. Anita, "Handwritten Script Identification from a Bi-Script Document at Line Level using Gabor Filter," in *International Workshop on Soft Computing Applications and Knowledge Discovery*, 2011, pp. 94–101.

[38] S. Huang and N. Sargur, "Word Segmentation of Offline Handwritten Documents, Document Recognition and Retrieval," in *(XV) The International Society of Optics and Photonics Annual Symposium*, 2008, p. 6815(68150E).

[39] G. Louloudisa, B. Gatosb, I. Pratikakisb, and C. Halatsisa, "Text Line Detection in Handwritten Documents," *Pattern Recognit.*, vol. 41, pp. 3758–3772, 2008.

[40] G. Louloudisa, B. Gatosb, I. Pratikakisb, and C. Halatsisa, "Text line and word segmentation of handwritten documents," *Pattern Recognit.*, vol. 42, pp. 3169–3183, 2009.

[41] U. Pal, S. Sinha, and B. B. Chaudhuri, "Multi-script Line Identification from Indian Documents," in *7th International Conference on Document Analysis and*

Recognition (ICDAR), 2003, pp. 880–884.

[42] A. Alaei, U. Pal, and P. Nagabhushan, "A New Scheme for Unconstrained Handwritten Text-line Segmentation," *Pattern Recognit.*, vol. 44, pp. 917–928, 2011.

[43] S. Vajda, K. Roy, U. Pal, B. B. Choudhury, and A. Belaid, "Automation of Indian Postal Documents Written in Bangla and English," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 8, pp. 1599–1632, 2009.

[44] A. Khandelwal, P. Choudhury, R. Sarkar, S. Basu, M. Nasipuri, and N. Das, "Text Line Segmentation for Unconstrained Handwritten Document," in *3rd International Conference on Pattern Recognition and Machine Intelligence*, 2009, pp. 369–374.

[45] B. V Dhandra and M. Hangarge, "Global and Local Features Based Handwritten Text Words and Numerals Script Identification," in *International Conference on Computational Intelligence and Multimedia Applications*, 2007, pp. 471–475.

[46] Spitz L, "Determination of the Script and Language Content of Document Images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, pp. 234–245, 1997.

[47] J. Hochberg, P. Kelly, T. Thomas, and L. Kerns, "Automatic Script Identification from Document Images using Cluster-based Templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, pp. 176–181, 1997.

[48] J. Ding, L. Lam, and C. Y. Suen, "Classification of Oriental and European Scripts by Using Characteristic Features," in *4th International Conference Document Analysis and Recognition*, 1997, pp. 1023–1027.

[49] T. N. Tan, "Rotation Invariant Texture Features and Their Use in Automatic Script Identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 7, pp. 751–756, 1998.

[50] S. Chanda, O. R. Terrades, and U. Pal, "SVM Based Scheme for Thai and English Script Identification," in *9th International Conference on Document Analysis and Recognition*, 2007, pp. 551–555.

[51] J. J. Lee and J. H. Kim, "A Unified Network-based Approach for Online Recognition of Multi-Lingual Cursive Handwritings," in *5th International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, pp. 393–397.

[52] S. M. Obaidullah, C. Halder, K. C. Santosh, N. Das, and K. Roy, "PHDIndic_11: Page-level handwritten document image dataset of 11 official Indic scripts for script identification," *Multimed. Tools Appl.*, p. doi:10.1007/s11042-017-4373-y, 2017.

[53] S. M. Obaidullah, K. C. Santosh, C. Halder, N. Das, and K. Roy, "Word-level thirteen official Indic languages database for script identification in multi-script documents," in *International Conference on Recent Trends in Image Processing and Pattern Recognition (RTIP2R 2016)*, 2016, p. (accepted).

[54] S. M. Obaidullah, C. Halder, N. Das, and K. Roy, "A New Dataset of Word-level Offline Handwritten Numeral Images from Four Official Indic Scripts and Its Benchmarking using Image Transform Fusion," *Int. J. Intell. Eng. Informatics*, vol. 4, no. 1, pp. 1–20, 2016.

[55] R. Wilkinson *et al.*, "The First Census Optical Character Recognition Systems," in *Conference #NISTIR 4912 (The U.S. Bureau of Census and the National Institute of Standards and Technology, Gaithersburg, MD*, 1992.

[56] C. Y. Suen, C. Nadal, R. Legault, T. Mai, and L. Lam, "Computer recognition of unconstrained handwritten numerals," in *Proceedings of IEEE*, 1992, p. 80(7):1162-1180.

[57] J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 550–554, 1994.

[58] Y. L. Cun, L. Bottou, Y. Bengio, and P. Haffiner, "Gradient based learning applied to document recognition," in *Proceedings of IEEE*, 1998, p. 86(11):2278-2324.

[59] U. Marti and H. Bunke, "A full English sentence database for off-line handwriting recognition," in *Proceedings of the International Conference on Document Analysis and Recognition*, 1999, pp. 705–708.

[60] U. Marti and H. Bunke, "An English Sentence Database for Off-line Handwriting Recognition," *Int. J. Doc. Anal. Recognit.*, vol. 5, no. 39–46, 2002.

[61] M. Zimmermann and H. Bunke, "Automatic Segmentation of the IAM Off-line Database for Handwritten English Text," in *Proceedings of the International Conference*

on Pattern Recognition, 2000, p. 4:35-39.

[62] "http://www.iam.unibe.ch/fki/databases/iam-handwriting-database," 2016. .

[63] B. Gatos, N. Stamatopoulos, and G. Louloudis, "Handwriting segmentation contest," in *Proceedings of the International Conference on Document Analysis and Recognition*, 2009, pp. 1393–1397.

[64] U. Bhattacharya and B. B. Chaudhuri, "Databases for research on recognition of handwritten characters of Indian scripts," in *Proceedings of the International Conference on Document Analysis and Recognition*, 2005, pp. 789–793.

[65] B. B. Chaudhuri, "A complete handwritten numeral database of Bangla-A major Indic script," in *10th International Workshop on Frontiers of Handwriting Recognition (IWFHR), La Baule, France*, 2006, pp. 379–384.

[66] M. W. Saqheer, C. L. He, N. Nobile, and C. Y. Suen, "A New Large Urdu Database for Off-Line Handwriting Recognition," in *15th International Conference on Image Analysis and Processing*, 2009, pp. 538–546.

[67] B. Nethravathi, C. P. Archana, K. Shashikiran, A. G. Ramakrishnan, and V. Kumar, "Creation of a huge annotated database for Tamil and Kannada OHR," in *International Workshop on Frontiers of Handwriting Recognition IWFHR*, 2010, pp. 415–420.

[68] A. Aleai, P. Nagabhushan, and U. Pal, "A Benchmark Kannada Handwritten Document Dataset and Its Segmentation," in *International Conference on Document Analysis and Recognition (ICDAR)*, 2011, pp. 140–145.

[69] A. Raza, I. Siddiqi, A. Abidi, and F. Arif, "An Unconstrained Benchmark Urdu Sentence Database with Automatic Line Segmentation," in *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2012, pp. 491–496.

[70] A. Raza, I. Siddiqi, A. Abidi, and F. Arif, "QUWI: An Arabic and English Handwriting Dataset for Offline Writer Identification," in *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2012, pp. 746–751.

[71] V. J. Dongre and V. H. Mankar, "Development of Comprehensive Devnagari Numeral and Character Database for Offline Handwritten Character Recognition," *Appl. Comput. Intell. Soft Comput.*, vol. 2012, no. Article ID 871834, p. 5 pages, 2012.

[72] A. Aleai, P. Nagabhushan, and U. Pal, "Dataset and Ground truth for Handwritten Text in Four Different Scripts," *Int. J. Pattern Recognit. Artif. Intell. World Sci.*, vol. 26, no. 4, p. 1253001 (25 pages), 2012.

[73] M. Diem, S. Fiel, F. Kleber, and R. Sablatnig, "CVL-Database: An Off-line Database for Writer Retrieval, Writer Identification and Word Spotting," in *12th International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 560–564.

[74] S. Thadchanamoorthy, N. D. Kodikara, H. L. Premaretne, U. Pal, and F. Kimura, "Tamil Handwritten City Name Database Development and Recognition for Postal Automation," in *12th International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 793–797.

[75] N. Das, K. Acharya, R. Sarkar, S. Basu, M. Kundu, and M. Nasipuri, "A benchmark image database of isolated Bangla handwritten compound characters," *Int. J. Doc. Anal. Recognit.*, vol. 17, no. 4, pp. 413–431, 2014.

[76] N. Das, R. Sarkar, S. Basu, P. K. Saha, M. Kundu, and M. Nasipuri, "Handwritten Bangla Character Recognition using a Soft Computing Paradigm embedded in Two pass Approach," *Pattern Recognit.*, vol. 48, no. 6, pp. 2054–2071, Dec. 2014.

[77] "https://www.ethnologue.com/browse/names," 2016. .

[78] S. M. Obaidullah, K. C. Santosh, C. Halder, N. Das, and K. Roy, "Word-Level Multi-Script Indic Document Image Dataset and Baseline Results on Script Identification," *Int. J. Comput. Vis. Image Process.*, vol. 7, no. 2, pp. 81–94, 2017.

[79] S. M. Obaidullah, A. Mondal, N. Das, and K. Roy, "Script Identification from Printed Indian Document Images and Performance Evaluation Using Different Classifiers," *Appl. Comput. Intell. Soft Comput.*, vol. 2014, p. 12 pages, 2014.

[80] S. M. Obaidullah, C. Goswami, K. C. Santosh, C. Halder, N. Das, and K. Roy, "Separating Indic scripts with `matra' for effective handwritten script identification in multi-script documents," *Int. J. Artif. Intell. Pattern Recognit.*, vol. 31, no. 4, p. 1753003 (17 pages), 2017.

[81] B. B. Mandelbrot, *The Fractal Geometry of Nature (New York: Freeman)*. 1982.

[82] S. M. Obaidullah, C. Halder, K. C. Santosh, N. Das, and K. Roy, "Automatic Line-Level Script Identification From Handwritten Document Images - A Region-Wise Classification Framework For Indian Subcontinent," *Malaysian J. Comput. Sci.*, p. (accepted).

[83] M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-3, pp. 610–621, 1973.

[84] S. M. Obaidullah, N. Das, and K. Roy, "Gabor Filter Based Technique for Offline Indic Script Identification from Handwritten Document Images," in *International Conference on Devices, Circuits & Communications (ICDCCom-2014)*, 2014, pp. 1–6.

[85] S. M. Obaidullah, C. Halder, N. Das, and K. Roy, "An Approach for Automatic Indic Script Identification from Handwritten Document Images," in *2nd Doctoral Symposium on Applied Computation and Security Systems*, 2015, pp. 37–51.

[86] S. M. Obaidullah, C. Halder, N. Das, and K. Roy, "Numeral Script Identification from Handwritten Document Images," *Procedia Comput. Sci. J.*, vol. 54C, pp. 585–594, 2015.

[87] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet rep- resentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, 1989.

[88] S. M. Obaidullah, C. Halder, N. Das, and K. Roy, "Indic Script Identification from Handwritten Document Images – An Unconstrained Block-level Approach," in *IEEE 2nd International Conference on Recent Trends in Information Systems*, 2015, pp. 213–218.

[89] S. R. Deans, *Applications of the Radon Transform*. . Wiley Interscience Publications, New York, 1983.

[90] A. Kaehler and G. R. Bradski, *Learning OpenCV*. 2008.

[91] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Mach. Learn.*, vol. 29, no. 2–3, pp. 131–163, 1997.

[92] R. E. Fan, K.-W. Chang, C.-J. Hsieh, X. R. Wang, and C.-J. Lin, "LIBLINEAR: a library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, 2008.

[93] C. S. V and J. Ghosh, "Scale-based clustering using the radial basis function network," *IEEE Trans. Neural Networks*, vol. 7, no. 5, pp. 1250–61, 1996.

[94] A. J. Howell and H. Buxton, "RBF network methods for face detection and attentional frames," *Neural Process. Lett.*, vol. 15, no. 3, pp. 197–211, 2002.

[95] M. Sumner, E. Frank, and M. Hall, "Speeding up Logistic Model Tree Induction," in *Proceedings of the European Conference on Principles and Practice of Knowledge Discovery in Databases*, 2005, pp. 675–683.

[96] J. Huhn and E. Hullermeier, "FURIA: an algorithm for unordered fuzzy rule induction," *Data Min. Knowl. Discov.*, vol. 19, no. 3, pp. 293–319, 2009.

[97] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[98] B. Patil and N. V. Subareddy, "Neural Network Based System for Script Identification in Indian Scripts," *Sadhana-Academy Proc. Eng. Sci. IAS Springer*, vol. 27, no. 1, pp. 83–97, 2002.

[99] A. M. Elgammal and M. A. Ismail, "Techniques for Language Identification for Hybrid Arabic-English Document Images," in *IEEE Proceedings of the Sixth International Conference on Document Analysis and Recognition*, 2001, pp. 1100–1104.

[100] B. V. Dhandra, P. Nagabhushan, M. Hangarge, R. Hegadi, and V. S. Malemath, "Script Identification Based on Morphological Reconstruction in Document Images," in *18th International Conference on Pattern Recognition*, 2006, pp. 950–953.

[101] U. Pal and B. B. Chaudhuri, "Identification of Different Script Lines from Multi-script Documents," *Image Vis. Comput.*, vol. 20, no. 13–14, pp. 945–954, 2002.

[102] M. C. Padma and P. A. Vijaya, "Wavelet packet based texture features for automatic script identification," *Int. J. Image Process.*, vol. 4, no. 1, pp. 53–88, 2010.

[103] G. D. Joshi, S. Garg, and J. Sivaswamy, "Script identification from Indian documents," in *7th International Association of Pattern Recognition Workshop on Document Analysis Systems*, 2006, pp. 255–267.

[104] D. Dhanya, A. G. Ramakrishna, and P. B. Pati, "Script Identification in Printed Bilingual Documents," *Sadhana*, vol. 27, no. 1, pp. 73–82, 2002.

[105] P. B. Pati and A. G. Ramakrishnan, "Word-level multi-script identification,"

Pattern Recognit. Lett., vol. 29, no. 9, pp. 1218–1229, 2008.

[106] S. M. Obaidullah, R. Karim, S. Shaikh, C. Halder, N. Das, and K. Roy, "Transform Based Approach for Indic Script Identification from Handwritten Document Images," in *3rdInternational Conference on Signal Processing, Communications and Networking*, 2015, pp. 1–7.

[107] J. Demsar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

[108] M. Hollander and D. A. Wolfe, *Nonparametric Statistics*. J. Wiley New York, 1973.

[109] P. K. Singh, S. K. Dalal, R. Sarkar, and M. Nasipuri, "Page-level script identification from multi-script handwritten documents," in *Proceedings of the Third International Conference Computer, Communication, Control and Information Technology*, 2015, pp. 1–6.

[110] A. Busch, W. W. Boles, and S. Sridharan, "Texture for script identification," *EEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1720–1732, 2005.

[111] S. M. Obaidullah, C. Halder, N. Das, and R. Roy, "Bangla and Oriya Script Lines Identification from Handwritten Document Images in Tri-script Scenario," *Int. J. Serv. Sci. Manag. Eng. Technol.*, vol. 7, no. 1, pp. 43–60, 2016.

[112] B. Gatos, I. Pratikakis, and S. J. Perantonis, "Efficient Binarization of Historical and Degraded Document Images," in *8th International Workshop on Document Analysis Systems*, 2008, pp. 447–454.

[113] A. Alaei, S. Pal, U. Pal, and M. Blumenstein, "An Efficient Signature Verification Method Based on an Interval Symbolic Representation and a Fuzzy Similarity Measure," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 10, pp. 2360–2372, 2017.

[114] A. Das, M. Ferrer, U. Pal, S. Pal, M. Diaz, and M. Blumenstein, "Multi-script versus single-script scenarios in automatic off-line signature verification," *IET Biometrics*, vol. 5, no. 4, pp. 305–313, 2016.

[115] N. Das, R. Sarkar, S. Basu, P. Saha, M. Kundu, and M. Nasipuri, "Handwritten Bangla character recognition using a soft computing paradigm embedded in two pass approach," *Pattern Recognit.*, vol. 48, no. 6, pp. 2054–2071, 2015.

[116] R. Sarkhel, N. Das, A. Saha, and M. Nasipuri, "A multi-objective approach towards cost effective isolated handwritten Bangla character and digit recognition," *Pattern Recognit.*, vol. 58, pp. 172–189, 2016.

Gmail **Sk Obaidullah <sk.obaidullah@gmail.com>**

Acceptance Confirmation Of Manuscript: AUTOMATIC LINE-LEVEL SCRIPT IDENTIFICATION FROM HANDWRITTEN DOCUMENT IMAGES - A REGION-WISE CLASSIFICATION FRAMEWORK FOR INDIAN SUBCONTINENT

1 message

editormjcs staff <editormjcs@um.edu.my> Tue, Oct 25, 2016 at 7:59 AM
To: Sk Md Obaidullah <sk.obaidullah@gmail.com>, Chayan Halder <chayan.halderz@gmail.com>, nibaranju@gmail.com, Kaushik Roy <kaushik.mrg@gmail.com>, santosh.kc@usd.edu
Cc: tutut mjcs <tutut.mjcs@gmail.com>, ramdr staff <ramdr@um.edu.my>

Dear Sk Md Obaidullah, Chayan Halder, K. C. Santosh, Nibaran Das & Kaushik Roy,

We are pleased to confirm that your paper *Automatic Line-Level Script Identification From Handwritten Document Images - A Region-Wise Classification Framework For Indian Subcontinent* has been accepted for publication in the Malaysian Journal of Computer Science (MJCS).

We will inform you the volume and number of the publication and send you the final edited version of your manuscript for author to proofread before we publish it.

Thank you for submitting your work to our journal.

Regards

Executive Editor
Malaysian Journal of Computer Science
Faculty of Computer Science & Information Technology
University of Malaya
Kuala Lumpur, 50603
http://ejum.fsktm.um.edu.my/
editormjcs@um.edu.my

International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 31, No. 4 (2017) 1753003 (17 pages)

DOI: 10.1142/S0218001417530032

World Scientific
www.worldscientific.com

Separating Indic Scripts with *matra* for Effective Handwritten Script Identification in Multi-Script Documents

Sk Md Obaidullah* and Chitrita Goswami†

Departmemt of Computer Science & Engineering
Aliah University Kolkata, West Bengal, India
**sk.obaidullah@gmail.com*
†*chtrgswm@gmail.com*

K. C. Santosh‡

Department of Computer Science
The University of South Dakota, SD, USA
santosh.kc@usd.edu

Nibaran Das

Department of Computer Science & Engineering
Jadavpur University, Kolkata, India
nibaran@gmail.com

Chayan Halder§ and Kaushik Roy¶

Department of Computer Science
West Bengal State University, Kolkata, India
§*chayan.halder@gmail.com*
¶*kaushik.mrg@gmail.com*

Received 28 July 2016
Accepted 19 September 2016
Published

We present a novel approach for separating Indic scripts with 'matra', which is used as a *precursor* to advance and/or ease subsequent handwritten script identification in multi-script documents. In our study, among state-of-the-art features and classifiers, an optimized fractal geometry analysis and random forest are found to be the best performer to distinguish scripts with 'matra' from their counterparts. For validation, a total of 1204 document images are used, where two different scripts with 'matra': Bangla and Devanagari are considered as positive samples and the other two different scripts: Roman and Urdu are considered as negative samples. With this precursor, an overall script identification performance can be advanced by more than 5.13% in accuracy and 1.17 times faster in processing time as compared to conventional system.

‡Corresponding author.

Keywords: Handwritten script identification; scripts with 'matra'; *precursor*; fractal geometry analysis; two-pass approach.

1. Introduction

We are living in the digital age. Documents are being digitized, as we move towards a 'paperless' world. For such 'paperless' world, first we need to convert the physical documents into digital form. To make the digitized documents editable and searchable, we need OCR to recognize the characters/texts. Before that, script recognition is an equally important pre-requisite, since OCR is script dependent. It becomes even more important in the scenario of multi-lingual document processing. In India alone, we have 13 different official scripts (including Roman) and 23 different languages (including English), which are often used in combination. Therefore, there exits a need of robust script identification to make the OCR efficient, when considering multi-scripts documents.[11] In this paper, we study the effect of separating scripts having 'matra' from those which do not have, on script identification performance. A 'matra' is a topological property in few of the major Indian scripts namely Bangla, Devanagari. With this script separation, we can thus ease and/or advance the subsequent script identification performance, and therefore we call it a *precursor*.

State-of-the-art works on script identification based on Indic and non-Indic scripts have been reported in literature since last decade.[4,5,8,13–15,17,18,21] Ghosh *et al.*[4] proposed a comprehensive review on script identification for Indic and non-Indic scripts. Hochberg *et al.*[8] proposed a page level script identification technique using some textual features and cluster-based template matching. Zhu *et al.*[21] proposed a shape codebook-based technique for script identification from few Indic and non-Indic scripts. Rotation invariant texture features using multi-channel Gabor filtering and gray level co-occurrence matrix was employed by Singhal *et al.*[18] to identify Devanagari, Bangla, Telugu and Roman scripts. DCT and wavelet-based feature was used by Rajput and Anita[14] for block level script identification. Sarker *et al.*[17] proposed handwritten word-level script identification from mixed type of documents using horizontal, foreground background transitions and 'shirorekha' (present in Devanagari and Bangla script) based features. Among the recent works, Hangarge *et al.*[5] applied directional discrete cosine transform-based approach to classify six Indic scripts namely Roman, Devanagari, Kannada, Tamil, Telugu, Malayalam. Rani *et al.*[15] implemented a character-level script identification technique on Gurumukhi and Roman scripts using Gabor filter, gradient-based feature with SVM classifier. Pardeshi *et al.*[13] reported word-level handwritten script identification using collective features of discrete cosine transform, discrete wavelet transform, radon transform, and statistical filter. But all the above mentioned works are carried out in a single pass i.e. features are globally extracted on all the scripts and then classified scripts into their corresponding classes. These works did not consider different optimization factors such as feature dimensionality and time

complexity that effect the overall performance of any script identification system. Besides and very importantly no work about script separation has been reported in the literature.

While stating aforementioned works, in this paper, we present a novel framework which can be used as a *precursor* to ease subsequent Indic scripts identification problem. The proposed method, used as a *precursor*, also optimizes two important factors: feature dimension and time complexity. Our proposed system follows a two-pass approach. During the first pass, we separate Indic scripts into two different classes by using a topological property: (i) scripts with 'matra' and (ii) scripts without 'matra'. In the second pass bi-script classification is done. We then compared an overall script identification performance of the proposed work with the conventional system. At this point, we mention that this work is the thorough extension of our earlier proof-of-concept work reported in the conference proceedings.[12] To make it more clear, our key contribution can be highligted as follows:

- Separating handwritten Indic scripts with 'matra' from their counterpart i.e. scripts without 'matra' eases and/or advances the subsequent script identification task.
- For validation, a line-level document image dataset from four demographically popular Indic scripts (two scripts with 'matra': Bangla and Devanagari, and rest two scripts without 'matra': Roman and Urdu) was prepared.[a]
- The proposed work is compared with conventional script identification system and we conclude that it outperforms the conventional one in both accuracy and processing time.

Our proposed method can be explained as shown in Fig. 1. It starts with feature extraction for line-level documents, and three different classifiers to make a decision for separating scripts with and without 'matra'. For feature extraction, fractal geometry analysis (FGA), Canny edge detector (CED) and morphological line transform (LT) are used. Similarly, for script separation task, three different classifiers such as multi-layer perceptron (MLP), Bayesnet (BN) and random forest (RF) are used. We carefully check which combination (i.e. feature-classifier) performs the best. Script separation is followed by script identification process. Note that script separation process is completely different than the script identification process. In the script identification process, like we have mentioned earlier, the main aim is to show how useful the script separation is.

As stated before, there are 13 scripts (including Roman) in India and few of the major scripts can be classified on the basis of a special topological property known as 'matra'. A 'matra' is a horizontal line present on the upper part of scripts such as Bangla and Devanagari. When user starts writing with pen or pencil he/she draws the line at the top and then starts writing the graphemes below this line with some touching component in between. As an example, Fig. 2 shows the presence of 'matra'

[a]The dataset is available for research purpose, upon request.

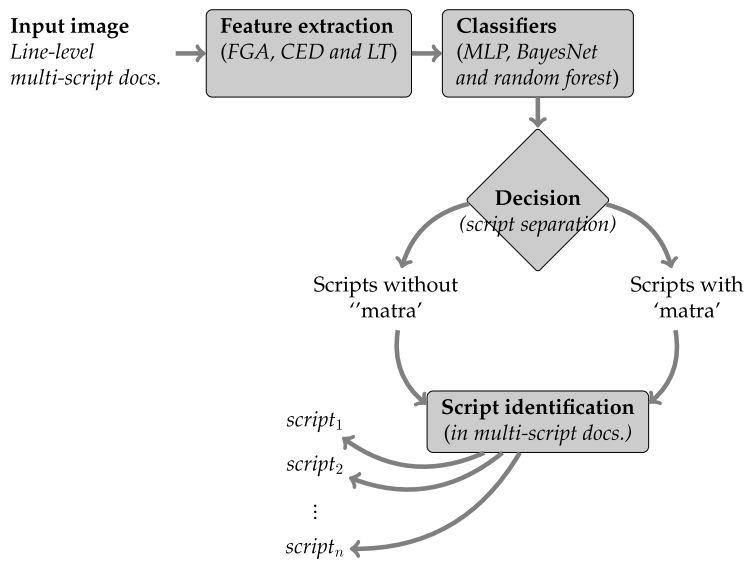


Fig. 1. Workflow: it starts with feature extraction for line-level documents, and three different classifiers to make a decision for separating scripts with and without 'matra'. Script separation is followed by script identification process. Script separation is just to group scripts into two different categories: scripts with 'matra' and without 'matra'. In the latter process i.e. script identification process, the main aim is to show how useful the script separation is.

on Bangla and Devanagari scripts, and absence of the same in Roman and Urdu scripts. A horizontal line on the top can be clearly identified, which is drawn over the connected graphemes. In India, there are several scripts with and without 'matra', and for our current work, we have considered two scripts with 'matra': Bangla and

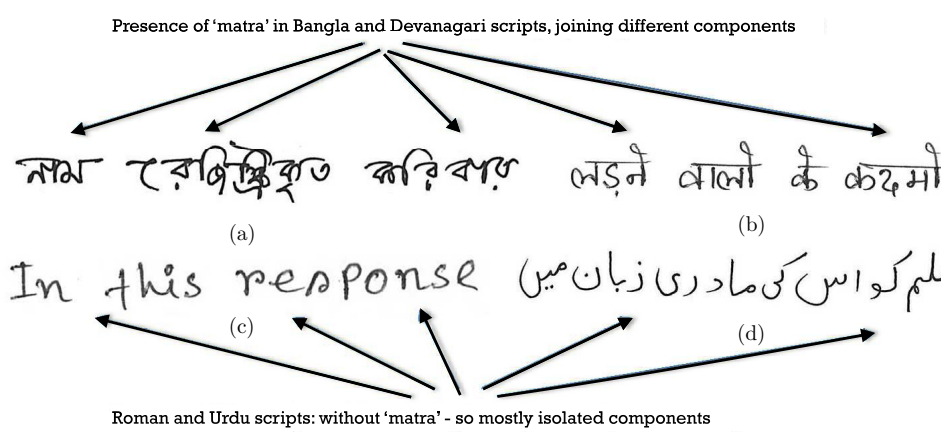


Fig. 2. Presence of 'matra' in (a) Bangla, (b) Devanagari scripts and the same is absent in (c) Roman, (d) Urdu scripts in Roman. 'matra' joins different character resulting in a large connected component (in case of (a) and (b)), whereas, component size is relatively smaller for scripts without 'matra' (in case of (c) and (d)).

Devanagari and the other two scripts without 'matra': Roman and Urdu. These four scripts were chosen observing their wide demographic distribution in India.[3]

The remainder of the paper is organized as follows. In Sec. 2, proposed methodology is discussed. Section 3 provides experimentation details including data collection, pre-processing, feature extraction process, comparison of the proposed method with the state-of-the-arts. We state our conclusion in Sec. 4.

2. The Proposed Method

2.1. *Feature extraction*

The problem of script identification depends on the fact that different scripts have unique visual attributes and spatial pixel distribution which make it easy to distinguish from one to the other. So, the primary task associated with script identification is to devise a technique to identify these features from a document image and then classify document's script accordingly. As our problem solely relies on 'matra' separation and then script classification, we choose such state-of-the-art techniques which are able to do so. We considered three features:

(1) fractal geometry analysis (FGA),
(2) Canny edge detector (CED) and
(3) morphological line transform (LT).

In what follows, we explain them in brief.

2.1.1. *1D FGA*

Inspired by the previous work,[10,16] we optimize the feature dimension of FGA (to 1-D) and propose a faster algorithm as it directly extracts features from the topological distribution of the pixels (presence or absence of 'matra'). Fractal geometry is a mathematical idea that is used to describe, model and analyze complex forms.[10] Mathematically, a fractal is defined as a set for which Hausdorff–Besicovich dimension is strictly greater than the topological dimension. The Hausdorff–Besikovich dimension (D_H) is defined by

$$D_H = \lim_{\varepsilon \to 0^+} \frac{\ln N_\varepsilon}{\ln 1/\varepsilon},$$

where N_ε is the number of elements of ε diameter required to cover the object in the embedded space.

For discrete data, we are interested to find a deterministic fractal and the associated fractal dimension (D_f). Following the above equation, D_f can be defined as the ratio of the number of self-similar pieces, N with magnification factor $1/r$ into which an image may be segmented. However, objects cannot be described with an integer value (in our experiment these objects are basically connected components in handwritten documents). These objects are said to have a "fractional" dimension,

$D_f = \frac{\ln N}{\ln 1/r}$. D_f may be a noninteger value, unlike objects that lie stringently in Euclidean space, which have only an integer value. This is because, handwritten scripts tend to have more crooked lines than straight one that resembles non-Euclidean geometry. We have used box-counting algorithm to compute D_f. More specifically, a box is equivalent to one pixel value. In our study, if pixel density of the connected components of different scripts with 'matra' and without 'matra' is computed, we observe the significant difference in D_fs that are computed from upper part and lower part of the image components. We can summarize it in three steps:

(1) Compute D_f from both upper (D_f^u) and lower (D_f^l) parts of each image component.
(2) Take the average of both upper and lower components: $D_{f,\text{avg}}^u$ and $D_{f,\text{avg}}^l$, respectively.
(3) Compute their ratio: $D_{f,\text{avg}}^u / D_{f,\text{avg}}^l$.

In Fig. 3, sample results are shown for Bangla, Devanagari, Roman and Urdu scripts.

Fig. 3. Illustrating fractal dimension of (a) Bangla, (b) Devanagari, (c) Roman and (d) Urdu scripts, where topmost part shows original line level document image, middle and lower part show fractal dimension (D_f) of upper profile and lower profile, respectively for each of the four scripts (a)–(d).

2.1.2. *Canny edge detector*

The process of Canny edge detection algorithm[19] can be broken down to five different steps.

(1) *Apply smoothing.* It refers to blurring, which is aimed to remove noise. For this, a Gaussian filter is applied to convolve with the image, $G(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}}$.

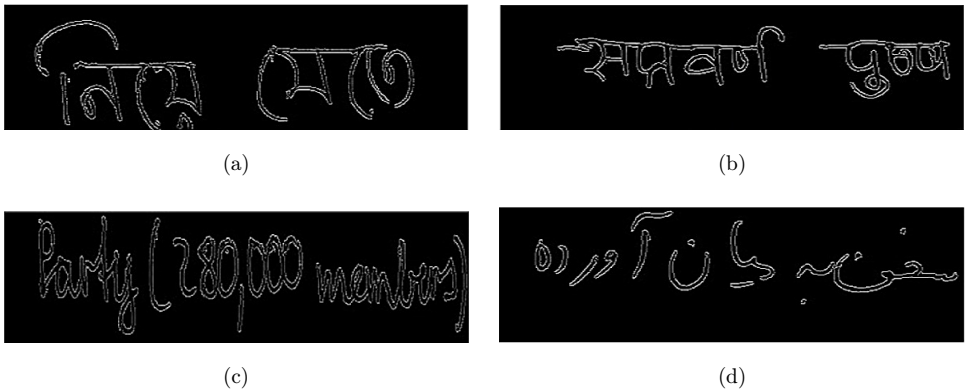
(2) *Compute the intensity gradients.* An edge in an image may point in a variety of directions. In case of Canny algorithm, four filters are used to detect horizontal, vertical and diagonal edges in the blurred image. The edge gradient and directions (by using G_x and G_y) can be determined: $G = \sqrt{G_x^2 + G_y^2}$ and $\theta = \text{atan2}(G_y, G_x)$. Note that the edge direction angle is rounded to one of four angles representing vertical, horizontal and the two diagonals: $0, \pi/4$, $\pi/2$ and $3\pi/2$.

(3) *Apply nonmaximum suppression.* It is an edge thinning technique, helps get rid of spurious response to edge detection.

(4) *Apply double threshold.* It determine potential edges, by using two different thresholds: high and low that are empirically set. High threshold yields strong edges, and in the same way, low threshold yields weak edges. Edges are suppressed if the pixel value is smaller than the low threshold value.

(5) *Track edge by hysteresis.* It finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

In our case, we apply CED on script image, as shown in Fig. 4. Since we are interested in separating scripts with 'matra', we calculate pixel density from the upper block.



(a) (b)

(c) (d)

Fig. 4. Sample output after applying CED algorithm on (a) Bangla, (b) Devanagari, (c) Roman and (d) Urdu scripts.

2.1.3. *Line transform*

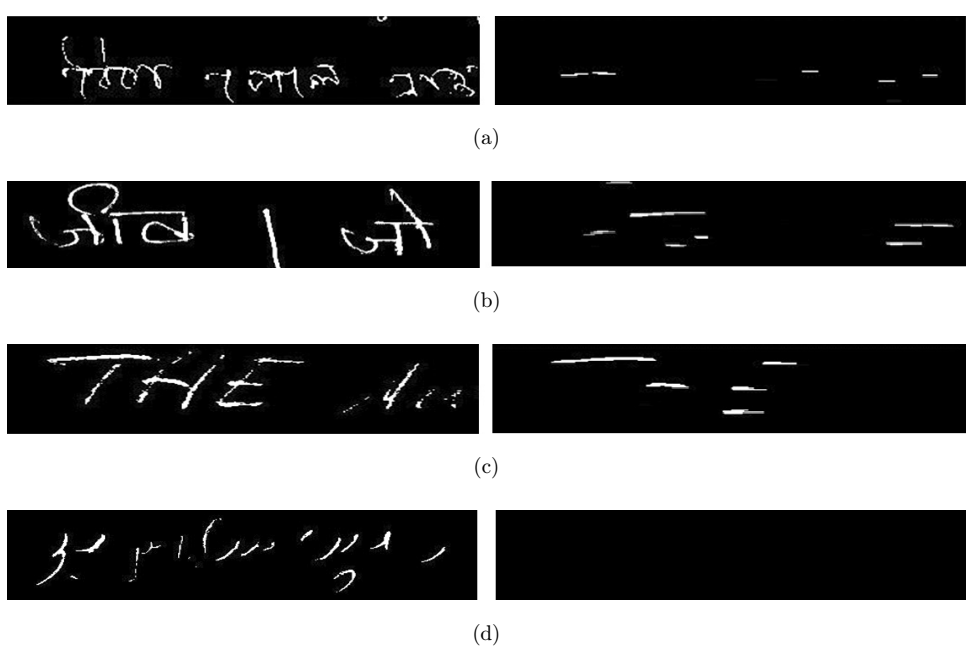
Considering 'matra' in our script, we aim to extract by using LT. For this, we convolve an original image with a kernel. The kernel is defined as a linear structuring element that decides the nature of morphological operations: *erosion* and *dilation* are considered. In this study, to duplicate 'matra'-like image component, our kernel (linear structuring element) of size 10 is 1×10 (i.e. [1 1 1 1 1 1 1 1 1 1]). Consider an image $I(x, y)$ and a kernel $K(u, v)$, both operations: erosion and dilation can be generally expressed as,

$$I_{\text{ero.}} = I \ominus K = \min\{I(x+u, y+v) - S(u, v)\} \quad \text{and}$$
$$I_{\text{dil.}} = I \oplus K = \max\{I(x-u, y-v) + K(u, v)\},$$

respectively. In our study, we apply this technique on image component and calculate pixel density as in CED. Figure 5 provides outputs of LT from four different scripts.

2.2. *Classification*

In our study, we take three state-of-the-art classifiers, aiming to find best feature-classifier combination. In what follows, we discuss them in brief.



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

Fig. 5. Illustration of LT output on (a) Bangla, (b) Devanagari, (c) Roman and (d) Urdu scripts. The first column shows original image and second one shows output image after applying LT.

(1) *Multilayer Perceptron* (*MLP*). It is a feed forward neural network, which has been widely used since decades[9] for pattern recognition applications. MLP uses layer wise connected nodes to build the architecture of the model. Each node (except for the input nodes) can be viewed as a neuron with a nonlinear activation function. In this paper, we use the sigmoid function as the activation function: $\sigma(x) = \frac{1}{1+\exp(-(\omega * x+v))}$, where the weight vector w and bias vector b in each layer pair are trained by the back propagation algorithm. We optimized the parameters for MLP using learning rate of 0.3, momentum 0.2, and empirically chose number of hidden layers.

(2) *Bayesnet* (*BN*). It is a well-known Bayesian classifier. For present work, to search the network structure we have used K2, a popular score-based search algorithm[3] which recovers the underlying graphic structure based on a predetermined order of nodes in a greedy fashion.

(3) *Random forests* (*RF*). It operates by constructing a group of decision trees at training time and outputting the class that is the mode of the classes output by individual trees.[2,7] Consider a training set $X = \{x_1, \ldots, x_n\}$ with corresponding responses $Y = \{y_1, \ldots, y_n\}$, we continuously select samples from the training set and fit trees to the samples, using bagging approach. In general, for $b = 1, \ldots, B$, we sample (with replacement) n training samples from X, Y, we call these X_b, Y_b. We then train a decision or regression tree f_b on X_b, Y_b. After training, predictions for unseen samples $\acute{x}$ can be made by averaging the predictions from all the individual regression trees on $\acute{x}$ or by taking the majority vote in the case of decision trees: $\hat{f} = \frac{1}{B}\sum_{b=1}^{B} \hat{f}_b(\acute{x})$.

3. Experiments

3.1. *Dataset*

For this work, we have prepared a dataset of 1204 line level handwritten document images. Out of which, 525 lines belong to scripts with 'matra' and remaining 679 lines are from scripts without 'matra'. Among the scripts with 'matra' there are 325 lines from Bangla and 200 lines from Devanagari script. On the other hand, for scripts without 'matra', Roman and Urdu contribute 370 and 309 lines, respectively. The dataset was collected from different persons with varying age, sex, educational background and demographic location. Table 1 shows the statistical overview of the dataset. More importantly, the dataset is available for research purpose, upon request. Figure 6 shows few sample images from our dataset.

3.2. *Experimental setup and evaluation metrics*

For validation, like in the conventional system, 5-fold cross-validation approach was considered during the training and testing process. This means that the entire dataset was distributed over a ratio of 4:1 (i.e. training:testing) and it was repeated

Table 1. Statistical distribution of the dataset.

Category: (Topological Property)	Script	# of Images
With 'matra'	Bangla	325
	Devanagari	200
Without 'matra'	Roman	370
	Urdu	309
Total		1204

five times such that all the instances participate in the decision of training and testing.

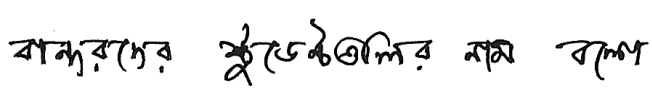
To evaluate the performance, the following three metrics are used: (i) sensitivity, (ii) specificity and (iii) accuracy that can be computed as follows:

$$\text{Sensitivity} = \frac{T_P}{T_P + F_N},$$

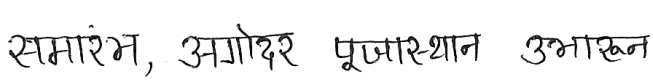
$$\text{Specificity} = \frac{T_N}{F_P + T_N}, \quad \text{and}$$

$$\text{Accuracy} = \frac{T_N + T_P}{F_P + T_N + T_P + F_N},$$

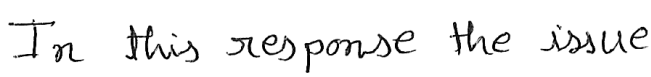
where T_P (true positive) is the total number of objects correctly classified, F_P (false positive) is the total number of objects of other classes falsely recognized as its own, T_N (true negative) is the total number objects of the other classes truly rejected as intruders and F_N (false negative) is the total number of objects falsely rejected. Note



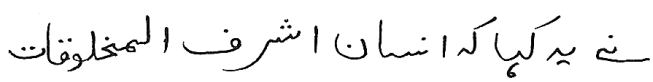
(a)



(b)



(c)



(d)

Fig. 6. Sample images from our dataset. Scripts with 'matra' (a) Bangla, (b) devanagari and without 'matra' (c) Roman, (d) Urdu.

that sensitivity refers to the probability that the classifier says an object belongs to a particular class and actually that one belongs to that particular class. Specificity is the probability that the test says an object does not belong to a particular class, when in fact, it does not belong to that class.

3.3. *Results*

Our primary goal, in this study, is to design an efficient *precursor* to advance/ease the subsequent Indic script identification. Therefore, in what follows, we first demonstrate how robust is our script separation task, and then address its usefulness in script identification process.

3.3.1. *Script separation: with and without 'matra'*

The main task of the *precursor* is to separate scripts with 'matra' from their counterpart. For simplicity, we call it as a two class problem. In our dataset, as mentioned in Sec. 3.1, the scripts with 'matra' (i.e. Bangla and Devanagari) are labeled as class 1, and the scripts without 'matra' (i.e. Roman and Urdu) are labeled as class 2. Different features (i.e. FGA, CED and LT) and classifiers (i.e. MLP, BN, RF) as explained in Sec. 2 are evaluated. In Table 2, we provide their results.

In Table 2, we achieved the highest possible accuracies of 95.68%, 85.30% and 76.49% for FGA-RF, CED-MLP and LT-BN feature-classifier combination, respectively. FGA outperforms other features when combining with RF classifier, and overall we have FGA-RF > CED-MLP > LT-BN. This can be supported by computing standard deviation of accuracies by these three classifiers for FGA, CED and LT as 0.6614, 1.7276 and 12.0517, respectively. It implies that the FGA is more robust as compared to others. The primary reason about 'why FGA' is due to the fact that it works on the principle of non-Euclidean geometry, and handwritten scripts tend to have more crooked lines than straight ones resemble non-Euclidean geometry. Between CED and LT, CED performs better, since LT cannot classify script like Roman.

Table 2. Script separation: using three different features and three different classifiers, measured in terms of sensitivity, specificity and accuracy (all in %).

Feature	Classifier	Sensitivity	Specificity	Accuracy
FGA	RF	93.90	97.05	**95.68**
	BN	92.95	96.02	94.68
	MLP	91.61	96.61	94.43
CED	RF	72.38	89.54	82.06
	BN	67.80	97.79	84.72
	MLP	69.14	97.79	**85.30**
LT	RF	72.19	77.76	75.33
	BN	75.61	89.54	**76.49**
	MLP	35.42	70.25	55.06

As stated before, our next experiment is to check whether the proposed *precursor* helps advance and/or ease subsequent script identification.

3.3.2. *Script identification*

In India, most of the multi-script documents are bi-script in nature, and therefore bi-script tests were done to prove that script separation can be considered as a *precursor* to effective script identification. To handle this, we have the following setup for script identification: (i) with and (ii) without *precursor*. The latter setup refers to conventional script identification system.

Script identification with precursor

We call script separation a *precursor* to script identification. In Table 2, we separated scripts into two different classes: c1 and c2, where c1 refers to scripts with 'matra' and c2 refers to scripts without 'matra'. Taking precursor into consideration, in Table 3, we provide script identification results between two different classes: c1 and c2, using four possible bi-script combinations: Bangla versus Roman, Bangla versus Urdu, Devanagari versus Roman and Devanagari versus Urdu. To evaluate the performance in terms of sensitivity, specificity and accuracy, we have considered the FGA-RF, CED-MLP and LT-BN feature-classifier combination since these are considered as the best performers as reported in Table 2. From Table 3, we observe that FGA+RF, feature-classifier combination performs better as compared to others.

Between CED and LT, we found that LT outperforms CED because LT analyzes well for those scripts with horizontal lines. For example, an absence of Roman script boosts its performance. On the other hand, CED takes pixels from the top-row, as a consequence, there exists false positives with 'matra'. In brief, scripts without 'matra'

Table 3. Script identification with *precursor*: using three different features and three different classifiers, in terms of sensitivity, specificity and accuracy (all in %) for possible bi-script combination.

Feature	Classifier	Script Combination	Sensitivity	Specificity	Accuracy
FGA	RF	Ben versus Rom	95.07	95.67	95.39
		Ben versus Urd	97.84	97.08	97.47
		Dev versus Rom	94.00	94.15	94.07
		Dev versus Urd	95.50	98.38	97.24
		Average (FGA-RF)	**95.60**	**96.32**	**96.04**
CED	MLP	Ben versus Rom	84.61	100.0	92.80
		Ben versus Urd	86.15	98.38	92.11
		Dev versus Rom	74.00	98.10	89.64
		Dev versus Urd	73.50	97.73	88.20
		Average (CED-MLP)	79.56	98.55	90.68
LT	BN	Ben versus Rom	76.92	59.45	67.62
		Ben versus Urd	98.76	99.35	99.05
		Dev versus Rom	70.00	74.59	72.98
		Dev versus Urd	92.00	99.35	96.46
		Average (LT-BN)	84.42	83.18	84.02

Table 4. Script identification without *precusor*: using FGA feature and RF classifier, in terms of sensitivity, specificity and accuracy (all in %) for possible bi-script combination from four different scripts.

Script Combination	Sensitivity	Specificity	Accuracy
Ben versus Rom	95.07	95.67	95.39
Ben versus Dev	96.30	86.50	92.57
Ben versus Urd	97.84	97.08	97.47
Rom versus Dev	94.00	94.15	94.07
Rom versus Urd	72.43	63.43	68.33
Dev versus Urd	95.50	98.38	97.24
Average	91.85	89.20	90.84

will have lesser density of horizontal lines, and LT outperforms CED. Another observation is that, Bangla versus Roman and Devanagari versus Roman has the highest confusion rate because letters like E, F, I, J, T and Z may produce the same effect as that of a top horizontal line, that is 'matra'.

Script identification without precursor

It refers to the conventional script identification problem, where we consider all possible b-script combination: $C_2^4 = 6$. These six different combinations are Bangla versus Roman, Bangla versus Devanagari, Bangla versus Urdu, Roman versus Devanagari, Roman versus Urdu and Devanagari versus Urdu. Using exactly similar evaluation setup and metrics as mentioned before, script identification result is shown in Table 4. Like before, FGA-RF (feature-classifier) combination yields best results, and therefore, we provide the same. Average scores of sensitivity, specificity and accuracy are 91.85%, 89.20% and 90.84%, respectively.

Comparison

At this point, one needs to raise a question: is the proposed script identification system (i.e. script identification with *precursor*) effective in comparison to the conventional ones? To address this, we compare performance from both systems: script identification with and without *precursor*. Note that, to make fair comparison, our experimental setup remains exactly the same for all tests. We used code blocks 12.11 software with OpenCV 2.2.0 library in the machine with Intel core i3 2.13 GHz processor and 4GB memory. In Table 5, we summarize their outcomes.

As shown in Table 5, we consider two different terms: accuracy and processing time into account, for comparison.

Table 5. Comparison: script identification with and without *precursor*, in terms of sensitivity, specificity and accuracy (all in %) and processing time (in seconds).

Script Identification	Sensitivity	Specificity	Accuracy	Time
With *precursor*	**95.26**	**96.46**	**95.97**	**0.82**
Without *precursor*	91.85	89.20	90.84	0.96

(1) Accuracy:
The use of script separation in script identification (as a *precursor*) advances the performance by more than 5.13%. Note that the comparison has been made between the system with and without script separation. It holds the same for sensitivity and specificity.

(2) Processing time:
Beside accuracy, processing time matters if we consider huge data. blue As reported in Table 3, the precursor (i.e. script separation) step avoids all possible bi-script combinations (C_2^4) to 4. This means that the conventional approach considers all of them (see Table 4). Therefore, in our test, the proposed system takes 0.82 s, on average. In contrast, without *precursor*, it takes 0.96 s, on average. This concludes that the precursor helps speed up the script identification process i.e. 1.17 times faster than conventional one.

Considering our study (and/or based on our results), we proved the usefulness of script separation for script identification task.

4. Conclusion

In this paper, we have presented a novel idea on separating Indic scripts, we have proved that it can be used as a *precursor* to advance and/or ease subsequent handwritten script identification in multi-script documents. In our study, we have used among state-of-the-art features and classifiers, an optimized FGA and RF are found to be the best performer to distinguish scripts with 'matra' from their counterparts. For validation, a total of 1204 document images are used, where two different scripts with 'matra': Bangla and Devanagari are considered as positive samples and the other two different scripts: Roman and Urdu are considered as negative samples. With this precursor, an overall script identification performance can be advanced by more than 5.13% in accuracy and 1.17 times faster in processing time as compared to conventional system.

In our further work, we will be analyzing some of the misclassified instances/scripts, and to recover this, we may combine other script-dependent features. But then a realistic trade-off between feature dimension and accuracy rate will be considered. Increasing the size of the dataset is another work. Note that, even though few benchmark handwritten datasets were reported in literature,[1] but majority of the scripts are still not available.

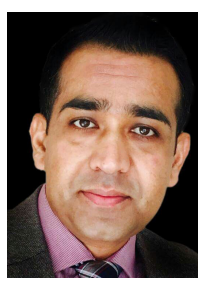
References

1. A. Aleai, U. Pal and P. Nagabhushan, Dataset and ground truth for handwritten text in four different scripts, *Int. J. Pattern Recogn. Artif. Intell.* **26**(4) (2012) 1253001.
2. G. J. Burghouts, Soft-assignment random-forest with an application to discriminative representation of human action in videos, *Int. J. Pattern Recogn. Artif. Intell.* **27**(4) (2013) 1350009.

3. G. Cooper and E. Herskovitz, A Bayesian method for the induction of probabilistic networks from data, *Mach. Learn.* **9** (1992) 330–347.
4. D. Ghosh, T. Dube and S. P. Shivprasad, Script recognition - A review, *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(12) (2010) 2142–2161.
5. M. Hangarge, K. C. Santosh and R. Pardeshi, Directional discrete cosine transform for handwritten script identification, in *Proc. Int. Conf. Doc. Anal. Recogn. ICDAR* (2013), pp. 344–348.
6. S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd edn. (Prentice Hall, 1998).
7. T. K. Ho, Random Decision Forests (PDF), in *Proc. 3rd Int. Conf. Document Analysis and Recognition* (1995) pp. 278–282.
8. J. Hochberg, K. Bowers, M. Cannon and P. Kelly, Script and language identification for handwritten document images, *In. J. Doc. Anal. Recogn.* **2** (2/3) (1999) 45–52.
9. M. Jinwen, A neural network approach to real-time pattern recognition, *Int. J. Pattern Recogn. Artif. Intell.* **15** (2001) 937–947.
10. B. B. Mandelbrot, *The Fractal Geometry of Nature* (Freeman, New York, 1982).
11. S. M. Obaidullah, S. K. Das and K. Roy, A system for handwritten script identification from Indian document, *J. Pattern Recogn. Res.* **8** (2013) 1–12.
12. S. M. Obaidullah, C. Goswami, K. C. Santosh, C. Halder, N. Das and K. Roy, Separating Indic scripts with 'matra' – A precursor to script identification in multi-script documents, *IAPR Int. Conf. Computer Vision & Image Processing* (2016).
13. R. Pardeshi, B. B. Chaudhuri, M. Hangarge and K. C. Santosh, Automatic handwritten indian scripts identification, *2014 14th Int. Conf. Frontiers in Handwriting Recognition* (2014), pp. 375–380.
14. G. Rajput and H. B. Anita, Handwritten script recognition using DCT and wavelet features at block level, *International Journal of Computer Application* **3** (2010) 158–163.
15. R. Rani, R. Dhir and G. S. Lehal, Script identification for pre-segmented multi-font characters and digits, *12th Int. Conf. Document Analysis and Recognition (ICDAR)* (2013), pp. 2010–1154.
16. K. Roy and U. Pal, Word-wise hand-written script separation for indian postal automation, *10th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR)* (2006), pp. 521–526.
17. R. Sarkar, N. Das, S. Basu, M. Kundu, M. Nasipuri and D. K. Basu, Word level script identification from bangla and devanagri handwritten texts mixed with roman script, *J. Comput.* **2**(2) (2010) 103–108.
18. V. Singhal, N. Navin and D. Ghosh, Script-based classification of hand-written text documents in a multi-lingual environment, *13th Int. Workshop on Research Issues in Data Engineering: Multi-lingual Information Management* (2003) pp. 47–54.
19. M. Thomas, *Image and Video Processing* (2008).
20. S. Vajda, K. Roy, U. Pal, B. B. Choudhury and A. Belaid, Automation of Indian postal documents written in Bangla and English, *Int. J. Pattern Recogn. Artif. Intell.* **23**(8) (2009) 1599–1632.
21. X. Zhu, Y. Y. Li and D. Doermann, Language identification for handwritten document images using a shape codebook, *Pattern Recogn.* **42** (2009) 3184–3191.

Sk Md Obaidullah has completed B. E. in Computer Science & Engineering from Vidyasagar University, M.Tech in Computer Science & Application from University of Calcutta in the year 2004 and 2009, respectively. He is a registered Ph.D. candidate in the Department of Computer Science & Engineering, Jadavpur University since November 2014. Presently he is working as an Assistant Professor in the Department of Computer Science & Engineering, Aliah University, Kolkata. He has published more than 25 research papers in reputed peer reviewed journal and national/international conferences. His research interests are document image processing, pattern recognition and computer vision.

Chitrita Goswami has completed B.Tech in Computer Science & Engineering from Aliah University, in the year 2016. She is currently employed as a web engineer, and is also continuing research work under the guidance of her mentor. She is extremely interested to continue research work and complete her higher studies and is looking for the right opportunity. Her research interests include image processing, computer vision and machine learning.



K. C. Santosh is an Assistant Professor at the University of South Dakota in Computer Science department. Before that, from 2013 to 2015, Dr. K. C. worked as a research fellow at the U. S. National Library of Medicine (NLM), National Institutes of Health (NIH). He worked as a postdoctoral research scientist at the LORIA research centre, Universite de Lorraine in direct collaboration with industrial partner ITESOFT, France, for 2 years. He also worked as a research scientist at the INRIA Nancy Grand Est research centre for 3 years, until 2011. Dr. K. C. has demonstrated expertise in pattern recognition, image processing, computer vision and machine learning with various applications in handwriting recognition, graphics recognition, document information content exploitation, medical image analysis and biometrics. Dr. K. C. published more than 70 research papers, including a book section in encyclopedia of electrical and electronics engineering. Dr. K. C. is an Association Editor of Int. J. of Machine Learning & Cybernetics, Springer.

Nibaran Das received his B.Tech degree in Computer Science and Technology from Kalyani Goverment Engineering College under Kalyani University, in 2003. He received his M.C.S.E and Ph.D.(Engg.) degree from Jadavpur University, in 2005 and 2012, respectively. He joined Jadavpur University as a faculty member in 2006. His areas of current research interest are OCR of handwritten text, optimization techniques and image processing. He has been an editor of Bengali monthly magazine "Computer Jagat" since 2005.

Chayan Halder has completed M.Sc in Computer Science West Bengal State University in 2010. He is currently working as a full time Ph. D. research scholar at Department of Computer Science, West Bengal State University, Barasat, India. He is recipient of the DST Inspire Fellowship, DST, Government of India. He has published more than 15 research papers in reputed conferences and peer reviewed journal. His research interest includes document image processing, computer vision and pattern recognition.

Kaushik Roy, B.E., M.E., Ph.D., is currently working as a Professor and Head, Department of Computer Science, West Bengal State University, Barasat, India. He has published more than 100 research papers/book chapters in reputed conferences and journals. His research interest includes Pattern Recognition, Document Image Processing, Medical Image Analysis, etc.