Offline Handwritten Malayalam Word Recognition using Machine Learning Techniques

Ph.D Thesis submitted to

COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

in partial fulfilment of the requirements

for the award of the degree of

DOCTOR OF PHILOSOPHY

under the Faculty of Technology by

JINO P J
(Reg.No.3762)

Under the guidance of

Dr. B. KANNAN

Department of Computer Applications
Cochin University of Science and Technology
Kochi - 682 022, Kerala, India

June 2018

Offline Handwritten Malayalam Word Recognition using Machine Learning Techniques

Ph.D. thesis

Author:
JINO P J
Department of Computer Applications
Cochin University of Science and Technology
Kochi - 682 022, Kerala, India
Email: jinopallickal@gmail.com

Supervising Guide:
Dr. B. Kannan
Professor & Head
Department of Computer Applications
Cochin University of Science and Technology
Kochi - 682 022.
Email: mullayilkannan@gmail.com

Cochin University of Science and Technology
Kochi - 682 022, Kerala, India.
June 2018

Dr. B. Kannan
Professor and Head
Department of Computer Applications
CUSAT
Kochi - 682 022

14th June 2018

Certificate

Certified that the work presented in this thesis entitled " *Offline Handwritten Malayalam Word Recognition using Machine Learning Techniques*" is based on the authentic record of research carried out by Shri. Jino P J under my guidance in the Department of Computer Applications, Cochin University of Science and Technology, Kochi- 682 022 and has not been included in any other thesis submitted for the award of any degree.

Dr. B. Kannan

Phone : +91 484 2576253

Declaration

I hereby declare that the work presented in this thesis entitled “ *Offline Handwritten Malayalam Word Recognition using Machine Learning Techniques* ” is based on the original research work carried out by me under the supervision and guidance of Dr. B. Kannan, Professor & Head, Department of Computer Applications, Cochin University of Science and Technology, Kochi - 682 022 and has not been included in any other thesis submitted previously for the award of any degree.

Kochi– 682 022 Jino P J
14th June 2018

Dr. B. Kannan
Professor and Head
Department of Computer Applications
Kochi- 682 022
CUSAT

June 14, 2018

Certificate

Certified that the work presented in this thesis entitled "Offline Handwritten Malayalam Word Recognition using Machine Learning Techniques" submitted to Cochin University of Science and Technology by Sri. Jino P J for the award of degree of Doctor of Philosophy under the faculty of technology, contains all the relevant corrections and modifications suggested by the audience during the pre-synopsis seminar and recommended by the Doctoral Committee.

Dr. B. Kannan

Acknowledgements

Thankfulness is the beginning of gratitude and gratitude is the completion of thankfulness. I wish to express my profound gratitude to Dr. B. Kannan, Professor and Head, Dept. Of Computer Applications, Cochin University of Science and Technology for spending his precious time and giving me valuable suggestions at the right time as my research supervisor. His perennial support, motivation, loving attitude and healthy criticism helped me to fulfill my dream successfully

I express my heartfelt gratitude to Dr.Ujjwal Bhattacharya, Associate Professor, Computer Vision and Pattern Recognition Unit, Indian Statistical Instiute, Kolkata for his unflinching support and inspiration in all my works.

I extend my sincere gratitude to Dr.M.Jathavedan, Dr. K. V. Pramod, Dr. A. Sreekumar, Ms. Malathi S, Dr.M. V.Judy and Dr.Sabu M K, Faculty members of the Department Of Computer Applications for their solid support and necessary corrections to accomplish my research goals.

I would like to thank Dr.Jomy John, Asst. Professor, KKTM Govt. College and Partha Sarathi Mukherjee, Project Linked Person, Indian Statical Institute for their selfless service and sheer dedication to share their technical expertise.

The endless support, affection and timely help extended by all the office staff, technical staff and librarian in the Department of Computer Applications are remembered with great sense of gratitude.

The constant support,sincere love and timely help extended by all my wellwishers,friends, co-researchers - Ramkumar R., Binu V.P., Bino Sebastian V., Santhoshkumar M. B., Aneurin Salim A.L., Vijith T.K., Vinu

V.S., Reshmi S., Sukrith B.,Jestin Joy.,Rajesh Kumar R., Krishnakumar M., Tibin Thomas., Arun Madhu., Jasir M. P., Sariga Raj., Smiju I.S., Soumya S., Bindu J.S., Parameswaran R., Sunil Kumar R., Merin Cheriyan., Shyam Sunder Iyer., Soumya George., Sruthi S., Simily Joseph., Cini Kurian., Sindhumol S., Remya A. R., Soumya T.V., Sruthi K S.,Rekha K.S., Ajlisa O.A., Haritha K., Shailesh S., Shernasmol A.A., Kumareshan K., Reshma Sanu and Divya Sindhulekha are remembered with much pleasure and token of gratitude. I am grateful to them for their unconditional support and encouragement

I am obliged to my parents for their sincere love and blessings. Thanks a million to my family especially my beloved Vini Rose, HSA, St.Joseph School, Kidangoor, Ankamaly and my son Richard for their true love, understanding, patience and inspiring words.

The financial support received from University is gratefully acknowledged.

Beyond all, I bow to God Almighty, the ultimate power enlightening every human being. From the core of my heart I thank you all for your unfailing support and motivation-that radiated a source of energy within me.

Jino P J

Contents

List of Figures

List of Tables

List of Acronyms

HOG	Histogram Of Oriented Gardient
PHOG	Pyramid Histogram of Oriented Gradient
SVM	Support Vector Machine
PCA	Principal Component Analysis
RF	Random Forest
MLP	Multi Layer Perceptron
CNN	Convolutional Neural Network
LSTM	Long Short Term Memory
BLSTM	Bidirectional Long Short Term Memory
CTC	Connectionist Temporal Classification
CER	Character Error Rate
WER	Word Error Rate
ED	Edit Distance
HMM	Hidden Markov Model
PHOC	Pyramidal Histogram Of Characters
RNN	Recurrent Neural Network
DL	Deep Learning
IID	Independent and Identically Distributed
BN	Batch Normalization
RBF	Radial Basis Function
JMHRDB	Jino Malayalam Handwriting Recogniton Database
CLC	CNN-LSTM-CTC

Abstract

Handwriting recognition is an important application of Pattern Recognition. Unfortunately, study of the same is rare as far as the Malayalam script is concerned. In this thesis, we discuss various methodologies for automatic recognition of offline handwritten Malayalam words and also describe about script independent recognition. Samples in our database were collected from a group of 199 natives belonging to different sections of the population with respect to age, sex, education, profession and income. Writers of its samples had age in the range 10 to 70. The set of writers consists of both left handed and right handed ones. They were asked to write the words of a given lexicon on a specific form printed on A4 size paper. Header part of this form was used to collect information about the writer such as name, age, qualification, signature etc. So the present database can be used for several other applications of handwriting analysis. This dataset consists of 31,020 handwritten Malayalam word samples.In addition to that 50 actual and 50 redesigned birth certificate forms are also included in this dataset. All the data collected is digitized to 300 dpi tiff image format. The major feature extraction methods used are HOG (Histogram of Oriented Gradient), PHOG (Pyramidal Histogram of Oriented Gradient), Wavelet and CNN (Convolutional Neural Network). Classification methods are MLP (Multi Layer Perceptron), SVM (Support Vector Machine), Random Forest, and BLSTM (Bidirectional Long Short Term Memory)-CTC(Connectionist Temporal Classification). Major contributions to the thesis include development of a moderately large database of handwritten word samples of Malayalam, a novel deep convolutional neural network (CNN) architecture for the purpose of automatic feature generation and recognition of handwritten Malayalam words. We investigated both lexicon specific and lexicon free approaches of offline handwriting of Malayalam and other Indian scripts. Also we propose a prototype for Birth Certificate form recognition.

Chapter 1

Introduction

"Innovation is hard. It really is. Because most people don't get it. Remember, the automobile, the airplane, the telephone, these were all considered toys at their introduction because they had no constituency. They were too new."

Nolan Bushnell

1.1 Prologue

Handwriting recognition can be either online or offline. The former case needs pen trajectory movements and in later case features of the image are considered for transcription. As a subsidiary of online recognition the method of in-air [1] is also getting popularity, where the movement of the finger can translate to a character or a word. Recognition of offline handwriting remains an open research problem for decades. Some extensive studies [2, 3, 4] of the problem, have led to remarkable success in recent past on a number of scripts of developed countries. Also, several handwriting recognition studies of a few Indian scripts such as Devanagari [5], Bangla [6], Tamil [7], Oriya [8], Malayalam [9] etc. are found in the literature. A survey of handwriting recognition of Indian scripts can be found in the studies of Pal et.al[10]. Although a majority of the works on Indian scripts considered only isolated characters, a few others [11, 12, 13] considered offline handwriting recognition problem in word level. Recognition of offline handwritten Malayalam words has not been explored by the researchers. Also, to the best of our knowledge, there is no available benchmark samples dataset of handwritten Malayalam words for comparison purposes. Handwritten image transcription is an important application of Image Processing, Pattern Recognition and Machine Learning. In Malayalam handwritten texts, the characters are not well separated, but the most of the words are separated by space. So the recognition of handwritten text on the real time data can start with word is an obvious choice.

The character of a human being can be identified through the style of the handwriting because it is unique and represents the individuality of a person [14]. The problem of handwritten recognition is not easy when compared with machine printed text. The major difficulty arised is the variability in the shapes of the characters in the documents, when the same person writes in different situations. In some extreme cases even humans feel difficult to identify and read the documents properly. There exists several scripts and languages across world. The recognition process can be either script dependent or independent. In this work we focus on script independent character and word recognition system with Malayalam as the prime language considered for the recognition. The following chapter discussed about objectives of the research work, Motivation behind the selection of the topic, Offline handwriting recognition methods/ approaches are explained in grassroots level, contributions of this research work is explained in detail, challenges faced during the period of research, publications related to the work, brief outline of the remaining chapters and finally summarize the chapter.

1.2 Objectives

- Develop a benchmarking dataset of Handwritten Malayalam words.
- The main objective of this research is to develop recognition methods for handwritten Malayalam words.
- Develop methods for script independent handwritten recognition.
- To identify the best Deep Learning approach for the recognition of handwritten documents.

1.3 Motivation

The following factors motivated us to select the present problem as our current research topic. Evaluation of voluminous handwritten assignments and answer scripts are done manually even in this era of information technological boom. The Availabilty of automatic handwriting recognition systems assist automation of various important services such as postal, courier etc.. Now-a-days the Government of Kerala is encouraging the use of Malayalam as the official language across the state. Thus, a lot of handwritten official documents are getting created everyday and the possible use of an efficient Malayalam handwriting recognizer should lead to better management of official works through fast retrieval of various information. Another factor for motivation is, according to the nature of Malayalam script, it is a challenging task that can be solved through machine learning methodologies.

1.4 Offline Handwriting Recognition Method

Numerous word/ string transcription methods have been developed since 1970's. Many of these systems have reported fairly high recognition accuracies, sometimes for vocabularies with tens of thousands of words. Automation of Handwriting recognition cannot written by a programmer by implementing some algorithms without data. The recognition method can be either holistic or analytic [15]. In holistic each word is considered as a recognition unit and in analytic, character or sub part of the words are considered as a recognition unit.

The method of handwriting recognition includes feature extraction and classification method. Selection of the relevant features and suitable classifier ensure a proper pattern matching algorithm that leads to the transcription of image to editable text. Representation of the pattern can be either statistical or structural. In this research work pattern is the handwritten word image[16]. From the machine learning perspective the method of learning can be classified into supervised, unsupervised, semi-supervised and sequence based. For lexicon specific recognition, we use supervised learning methods and for lexicon free recognition, learning methods are sequential and semi-supervised[17].

1.5 Origin of Indic Scripts and Its Charcateristics

Eighth schedule in constitution of India officialy declared 22 languages. They are: Assamese, Bangla(Bengali), Gujarati, Hindi, Kannada, Kashmiri, Manipuri(Meitei), Malayalam, Konkani, Marathi, Nepali, Oriya, Punjabi (Gurumukhi), Sanskrit, Sindhi, Tamil, Telugu, Urdu, Santhali, Bodo (Boro), Maithili and Dogri. These languages are written using 10 scripts. They are Bengali, Gujarathi, Devanagari, Malayalam, Kannada, Odia, Gurumukhi, Tamil, Telugu and Urdu. Classification of the languages in terms of its popularity is shown in Table 1.1 [18]

Table 1.1: List Of Scheduled Languages/Usage Wise- World wide♣

SI.No	Language	Offcial Language	Script	Users
1	Hindi	Bihar Chattisgarh Haryana Himachal Pradesh Jharkhand Madhya Pradesh Mizoram Rajasthan Uttar Pradesh Uttarakhand	Devanagari	534,271,550
2	Bengali	WestBengal	Bengali	261,862,630
3	Urdu	Jammu & Kashmir	Urdu	163,211,530
4	Telugu	Andhrapradesh	Telugu	79,771,240
5	Tamil	Tamilnadu	Tamil	74,678,890
6	Marathi	Maharashtra	Devanagari	74,796,800
7	Gujarati	Gujarat	Gujarati	46,888,670
8	Malayalam	Kerala	Malayalam	35,247,100
9	Maithili		Devanagari	34,085,000
10	Odia	Odisha	Odia	32,139,520
11	Punjabi(Gurumukhi)	Punjab	Gurumukhi	29,537,970
12	Sindhi		Devanagari	24,546,460
13	Nepali	Sikkim	Devanagari	24,131,000
14	Assamese	Assam	Bengali	12,828,310
15	Santhali		Bengali	6,220,280
16	Kannada	Karnataka	Kannada	46,752,570
17	Konkani	Goa	Devanagari	2,423,330
18	Dogri		Devanagari	2,280,000
19	Manipuri	Manipur	Bengali	1,485,000
20	Bodo(Boro)	Assam	Devanagari	1,334,380
21	Kashmiri		Devanagari	5,485,780
22	Sanskrit		Devanagari	211,100

♣ 'https://www.ethnologue.com/language/'

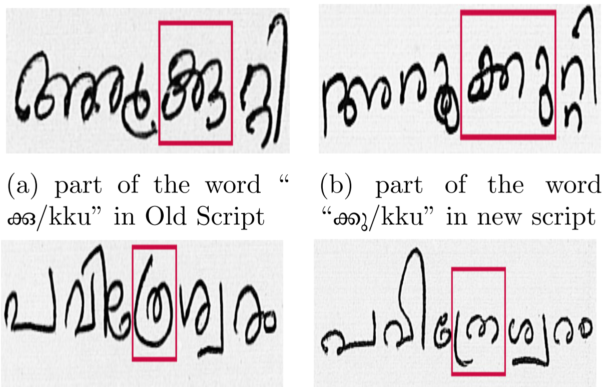
1.6 Research Problem

To develop script independent methodologies for automatic recognition of offline handwritten Malayalam words involving the entire character set of the reformed script [19].

1.7 Challenges

Malayalam is an agglutinative language, so the words can form in many ways. Develop a good language model is also challenging. The major challeges are listed below.

- Lack of proper benchmarking data set
- Word Extraction from the document.
- Use of new and old script in a mixed manner as shown in Figure 1.1a and 1.1b, part of the word in old and new script are marked in red rectangle.
- More number of characters in the form of compound characters, modifiers and consonants. Combining all the charcters from old and new script it will be around 600.
- variations in the shapes of handwritten words/characters. Writing style of the same person writing in different situations are not exactly in same pattern as shown in Figure 1.2a and Figure 1.2b.
- Scanned document quality also decided the recognition accuracy
- Intra class variablity and inter class similarity of characters and words.
- Segmentation to the basic unit or Grapheme level is difficult. In Figure 1.3a character “ട/T” and ൂ /uu, Figure 1.3b character “ര/ra” and ി/i, Figure 1.3c character “യ/ya” and “ ് ”, Figure 1.3d character



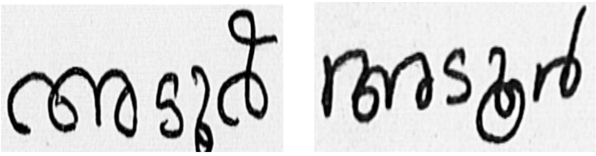
(a) part of the word “ക്കു/kku” in Old Script

(b) part of the word “ക്കു/kku” in new script

(c) part of the word “ത്ര/thra” in Old Script

(d) character “ത്ര/thra” in new script

Figure 1.1: Example For MixedScript- Character “ക്കു/kku” and “ത്ര/thra” in Old Script



(a) First Attempt

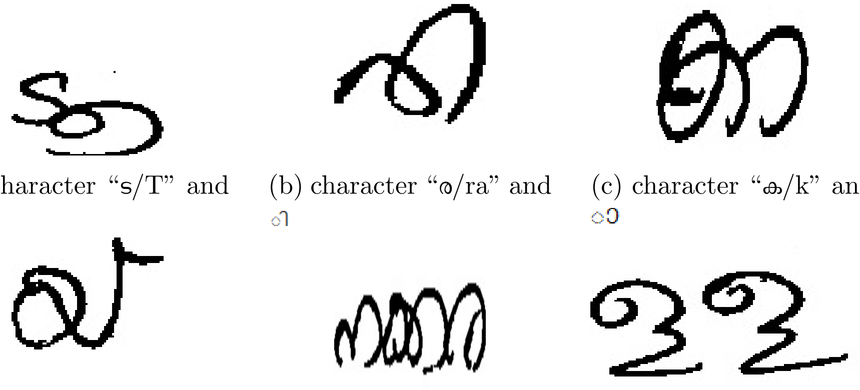
(b) Second Attempt

Figure 1.2: Challenges-Writing Style of Individual

“ക/k” and ാ, Figure 1.3e character “ന/na”,“ക്ക/kk” and “ര/ra” are joined together. In Figure 1.3f character “ള്ള” /LL oversegmented

1.8 Contributions

- Development of a moderately large database of 31020 handwritten word samples of Malayalam and 100 birth certificate application form images.
- Proposed a deep architecture of Convolutional Neural Network (CNN) for the purpose of automatic feature generation and recognition of



(a) character “ട/T” and ◌ൃ

(b) character “ര/ra” and ◌ി

(c) character “ക/k” and ◌ാ

(d) character “യ/ya” and “ ് ”

(e) character “ന/na”,“ ക്ക/kk” and “ര/ra”

(f) character “ള്ള”/LL oversegmented

Figure 1.3: Challenges Related Segmentation

handwritten Malayalam words. The same architecture has been found to improve the existing state-of-the-art of offline handwriting recognition of several major Indian scripts.

- Introduced a Recurrent Neural Network architecture (RNN) for handwritten Malayalam word recognition.

- Studied both lexicon specific and lexicon free approaches of offline handwriting Malayalam word recognition.

- This is the first major work on handwritten Malayalam word recognition involving the entire character set of the reformed script.

- Implementation of various other state-of-the-art feature extraction methods based on Wavelet, HOG (Histogram of Oriented Gradient), and PHOG (Pyramidal HOG) for the purpose of comparisons.

- Comparison of a few other state-of-the-art classifiers with the proposed deep architecture.

- Proposed a two-stage classification method for handwritten Malayalam word recognition.

1.9 Publications

Journals

- Offline Handwritten Malayalam Word Recognition using Wavelet Transform,Jino P J, Kannan Balakrishnan, International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 2, Issue 5, pp.948-954, September-October 2017

- Offline Handwritten Recognition of Malayalam District Name-A Holistic Approach, Jino P J, Kannan Balakrishnan, International Journal of Engineering and Technology, ISSN:2319-8613, Vol 9 No 2,pp.987-994, April-May 2017, Engg Journals Publications.

Springer/IEEE International Conference Proceedings

- Offline Handwritten Malayalam Word Recognition Using A Deep Architecture, Jino P J, Kannan Balakrishnan and Ujjwal Bhattacharya, SocProS 2017 at IIT Bhubaneswar, Advances in Intelligent Systems and Computing series of Springer, ISSN:2194-5357**(Received Best Paper Award)**

- Offline Handwritten Malayalam character Recognition using stacked LSTM, Jino P J, Jomy John and Kannan Balakrishnan, IEEE 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT) at Vimal Jyothi College, Chemberi, Kerala, pp:1587-1590 ISBN: 978-1-5090-6106-8

- Combined Approach for Binarization of Offline Handwritten Documents, Jino P J and Kannan Balakrishnan, IEEE 2017 4th International Conference on Communications and systems at Karpagam collge of engineering, Coimbatore, Thamilnad, pp:23-27, ISBN: 978-1-5090-3355-3

National Conference Proceedings

- Deep Architectures for Offline Handwritten Recognition - Necessity of a benchmark dataset in Indian Languages, Jino P J and Kannan Balakrishnan, Three day National seminar on Indian Language Technology:State and Prospects at University of Kerala, March 2018.
- Offline handwritten Malayalam character recognition:A Convolutional Neural Network Approach, Jino P J and Kannan Balakrishnan, National Conference on Indian Language Computing(NCILC) at Cochin University, pp:33-37, ISBN:978-81-936217-1-4, March 2018.
- HWR for Indian Languages: A Comprehensive Survey, Jino P J, Kannan Balakrishnan, NCILC at Cochin University, February 2014. published in CSI digital library.

1.10 Thesis Outline

Remaining of the thesis organized into the following chapters.

Chapter 2: Literature Review: Existing literature on handwriting recognition with special emphasis on Indic scripts has been studied. The study includes existing benchmark databases, preprocessing, feature extraction / selection and classification strategies.

Chapter 3: Details of the development of a new moderately large dataset of 31020 handwritten Malayalam word samples have been presented.

Chapter 4: Presents the proposed Deep Learning based lexicon specific recognition of offline handwritten Malayalam words.

Chapter 5: Presents a detailed comparison of the proposed recognition approach with a few traditional recognition strategies.

Chapter 6: Presents a study of lexicon free recognition of unconstrained offline handwriting in Malayalam.

Chapter 7: Presents a prototype system towards an application of offline Malayalam handwriting recognition.

Chapter 8: Concludes the work with a discussion on prospective directions of future studies.

1.11 Summary

This chapter describes the objectives and motivations of the present research study. A brief overview about the indic scripts and its characterestics are mentioned. The task of handwritten recognition process can be considered as a core Machine Learning problem to overcome the challenges explained in previous Section:1.7. Thesis contributions are also described.

Chapter 2

Literature Review

"Every new beginning comes from some other beginning's end."

Seneca

2.1 Introduction

Comprehensive research is done in the area of handwriting recognition. This chapter explores the techniques/ methodologies applied on various scripts with special emphasis on Indic scripts. This review focuses on various papers published in the last 10 years on the reputed journals and conference proceedings. The Word recognition problem can be treated in three different categories. First method is to consider the handwritten image as a whole word, second one tries to recognize the character by character, where the image should properly segmented to characters and the last method is segmentation free, where the word is predicted from the sequence labelling method. This review covers all the methods discussed above.

The study includes existing benchmark datasets as well as preprocessing, feature extraction / selection and classification strategies. There are several methods found in literature for handwritten word recognition, viz. 1) Template matching 2) HMM (Hidden Markov Model) based 3) Machine Learning /Deep Learning based methods. In these Recurrent Neural Network Deep Learning based methods are popular because it is the state-of-the-art architecture [20] with unconstrained handwriting recogntion and generic model recognitions.

2.2 History of Handwriting Recognition

A lot of research and development happened in the area of handwriting recognition in various languages for the last four decades. As part of the digitization, in 1870 Carey contributed the techonlogy for retina scanner. Followed by 1890, Nipkow comes with the sequential scanner. Earliest works are focused on the machine printed characters/words[21]. In 1990 as a break through in the process of handwriting recognition Prof.Ching Yee Suen initiated the conference International Workshop on Frontiers In Handwriting Recognition(IWFHR) on this subject and it attracted lot of researchers across the world. The major concern of this conference was the need of good standard dataset. In 1992, N.D Gorsky and T. Caesar observed that Hidden Markov Model can be used for the document recognition [22] . In 1993, M. Hamnak et.al introduced a method for Kanji Character recognition,widely used in Japanese writing system. In 1994, C.Y Suen and D Gullevic suggested a sentence level recognition of legal amounts in the bank checks written in cursive manner. A. EI Yoacoubi et.al put forward city name recognition on the mail system. A method for cursive word recognition was introduced by H Bunke in 1995[23].

2.3 Existing Benchmark Datasets

2.3.1 Non-Indic Datasets

IRONFF Dataset consists of French and English word (Latin script) images [24]. Total sample size of the database is 31,346 and lexicon size is 196. In one of the experiments with this dataset , the training sample size is 20,898 and test sample size is 10,448 [25].

IFN/ENIT dataset contains Tunisian Words in Arabic Script contributed by 411 writers. Total sample size of the dataset is 32492 words with the lexicon size of 2100 [26].

KHATT is an arabic word with meaning as “handwriting” and it stands for KFUPM Handwritten Arabic TexT and it can be categorized on the basis of age, gender and handedness[27].

Multilingual Automatic Document Classification Analysis and Translation(MADCAT) also known as OpenHaRT [28] developed by University of Pennsylvania. It contains more than 46000 Arabic handwritten documents from 453 writers. All the documents are scanned with 600 dpi in grayscale format.

Two editions of NIST dataset exists. In the first edition [29] it has forms, fields and characters. Final release of NIST dataset contributed by 3600 writers and it consits of 81000 charcters extracted from the forms and 91,500 text and phrases [30]. CENPARMI Handwritten digit dataset consists of 17000 digits of zip codes of US Postal department written by 3400 writers [31]. In Urdu offline handwriting the dataset consists of dates, digits, alphabets, numral strings and words. Lexicon size of word dataset is 57 and the total sample size is 19432[32]. CEDAR consists of 10000 words, 10000 zip codes and 50000 alpha numeric characters [33].

MNIST is the widely used benchmarking dataset for the recognition of handwritten digits and suitable for deep learning experiments with a huge

60000 training samples and 10000 testing samples[34]. The best result reported for this dataset is 99.77 % [35].

IAM dataset consists of 1539 pages of scanned text,5685 isolated labelled sentences,13353 isolated and labelled text lines and 115320 isolated labelled words.It is one of the best benchmarked datasets exists in the literature.

EMNIST is an extended version of MNIST dataset [36], which consists of 52 characters(both upper and lowercase) and 10 digits. Total of 814255 samples.

Total samples contained in ALEXU-WORD dataset is 25,114 Arabic words. 907 writers contributed to this dataset and the lexicon size is 109 [37].

RIMES(Reconnaissance et Indexation de donnes Manuscrites et de fac similS / Recognition and Indexing of handwritten documents and faxes) is a handwritten French Document data set. 1,300 people contributed to the creation of this dataset and it contains 12,723 pages[38].

CASIA is an acronym for Institute of Automation of Chinese Academy of Sciences, consists of both online and offline handwritten dataset[39]. Dataset consists of 52,230 lines of 5,091 pages, which contributed by 1019 writers.

GW a.k.a George Wahington dataset arranged in nine series consists of 65000 documents. The digitized version of some of these documents are available with library of congress[40]. Stauffer et.al [41] propose a graph dataset for the words of 20 letters from GW.

2.3.2 Indic Datasets

ISI Databases of Handwriting Samples consists three datasets of the scripts Devanagari dataset consists of 22,556 numerical samples contributed by

1049 writers. Bangla dataset consists of 12,938 numerical samples contributed by 556 writers. Oriya dataset consists of 5,970 numerical samples contributed by 356 writers[42]. It also consists of Bangla and Devanagari basic characters of 37,858 and 30,000 respectively with class size of 50 and 49.

Bangla Numeral Dataset consists of 23392 samples collected from postal mails and application forms [43].

Hindi Word Dataset consists 39700 Hindi word dataset with a lexicon size of 100. Total number of writers to form this dataset is 436 [44].

Hindi and Marathi Words using in Bank Cheques are created by [45] contains valid words extracted from 240 bank cheques. Lexicon size of Hindi word dataset is 106 and Marathi words it is 114. Total sample size of Hindi dataset is 8,480 and Marathi datataset is 18,240. Experiments with this dataset is reported in Chapter: 4.

Kalanajiyam means repository, which developed in two phases consists of handwritten Tamil images[46]. Phase-1 consists of isolated characters and phase-2 consists of paragraphs contributed by around 1000 individuals.

CMATERdb2.1.2 consists of Bangla Handwritten dataset with a lexicon size of 120 and sample size of 18000 images[47].

Roy Dataset consists of Bangla and Devanagari words with clear split of Test,Train and validation data[48]. The total sample size of Bangla and Devanagari are 17,091 and 16,128 respectively.

NewISIdb Bangla dataset consists of word, two paragraph dataset consists of basic characters and conjuntcs[49]. It consists of 815 unique words with a total sample size of 107,550.

Cursive And Language Adaptive Methodologies abbreviated as CALAM consists of offline handwritten Urdu text images[50]. Dataset is provided with detailed groud truth details and annotations are available in XML format. The total number of words samples extracted from the dataset is 46,664.

UCOM is an Urdu handwritten dataset contributed by 100 writers consits of 100 scanned pages[51]. Total sample size of the words is 62000.

KHTD(Kannada Handwritten Text Database) is a handwritten dataset for Kannada with proper annotation[52]. It consists of 204 scanned images of pages consists of 4298 lines and 26115 words.

2.4 Techniques For Handwriting Recognition

2.4.1 Related Works on Non-Indic Script

Arabic

Arabic, persian(Farsi) languages followed Arabic Script, it is written from right to left is considered in this survey. Khemiri et.al [53] propose a method using Probabilistic Graphical Models classifiers. Experiments are done using the words from IFN/ENIT dataset. Dynamic Bayesian Network provides an accuracy of 85.21 % result with 50 words and 6451 total samples. Vertical and Horizontal HMM(VH-HMM) provides an accuracy of 90.42 % with the same dataset. Structural and Statistical Features, number of pixel transitions and number of PAWs(Parts of Arabic Words) are used as features for classification. In another extension work [54] , with 83 words and 7881 word samples VH-HMM provides an accuracy of 90.02 % accuracy.

AlKhateeb et.al [55] compare the performance of IFN/ENIT dataset with Hidden Markov Model and Dynamic Bayesian Network. All the word images are height normalized to 45 pixels. With statistical features and Hidden Markov Model achieved an accuracy of 80 %.

Broumandnia et.al [56] propose a method for Farsi handwritten word recognition, where they use wavelet energy features in polar transformed image. Lexicon size used was 100 and they got an accuracy of 96 % with Mahalanobis classifier.

Latin

English and French are two popular languages coming under Indo-European family which uses Roman and French alphabets respectively. The major works discussed in this section using IAM and RIMES benchmark datasets produces state-of-the-art architecture.

Poznanski et.al [57] propose a method using Convolutional Neural Network with 1, 2, 3, 4, 5-gram based label encoding viz PHOC achieves 96.10% accuracy in word level and 98.10 % in character level with RIMES Dataset. The same architecture with IAM dataset produce an accuracy of 93.55% in word level and 96.56% in character level.

Bluche et.al [58] propose a Recurrent architecture produce 88.2 % in word level and 96.7 % in character level with a 4-gram language model. The same architecture with IAM dataset produce an accuracy of 88.1 % in word level and 95.1 % in character level with 3-gram language model.

In another work Bluche et.al [59] propose a combination of Convolutional Neural Network and Hidden Markov Model with a unigram language model achieved an accuracy of 90.8 % with ICDAR 2009 Rimes Dataset. The same architecture produced 79.5 % with IAM dataset.

Mensari et.al [38] developed a system for isolated and multiword recognition using RIMES dataset. HMM with Grapheme based and Sliding window approaches provided a result of 75 % and 78 % respectively. Recurrent Neural Network Approach provided 91.1 % accuracy. Finally these three approaches combined and achieved 95.14 % accuracy for Isolated word and 95.01 % accuracy with RIMES 2011 dataset.

Doetsch et.al [60] proposed a method using B(Bernoulli) and G(Gaussian) HMM(Hidden Markov Model) and LSTM. Pixels in the image are veritcally repositioned using Centre of Gravity based on Sliding Window approach provided the best accuracy. Experiments are done in Rimes dataset. The combination of GHMM with LSTM provided better accuracy with 90.3 % compared to BHMM.

Kozielski et.al [61] proved that preprocessing of the images with moment based normalization improves the accuracy of offline handwriting recognition.With HMM the system gives an accuracy of 86.6% using RIMES dataset. With IAM dataset it provides an accuracy of 62.7%.

Chinese

Messina et.al [62] use MultiDimensional Recurrent Neural Network(MDRNN) architecture for the recognition. The architecture consists of four directional LSTM(Long Short Term Memory)-Convolutional-fully connected layers. With language model the system provided an accuracy of 89.4 %.

Wang et.al [63] propose a framework to induce knowledge in hierarchical manner to the CNN(Convolutional Neural Network) for handwritten text recognition. In the case of text recogntion the knowledge required is, class label and information about the boundary of character. To incorporate the knowledge two heterogeneous CNN architectures (cascading CNN and Negative-awareness CNN) are implemented and the correct text is predicted using Language Model(LM). Heterogeneous CNN required both true and false samples. The highest accuracy of 94.02 % was reported with negative-awareness CNN and LM.

Surayni et.al [64] propose a CNN-Recurrent-Hybrid HMM architecture for the recognition of handwritten text. CNN used for feature extraction, Recurrent architecture for sequence labelling and hybrid HMM for label alignment. Accuracy achieved is 84.87%.

Wu et.al [65] present a CNN - Separable Multi Dimensional Recurrent Neural Network Architecture(SMDRNN). SMDRNN implemented using LSTM, horizontally and vertically in both directions. Connectionist Temporal Classification(CTC) is used for the sequence alignment.Language model implemented through weighted finite state transducers.The overall recognition accuracy is 90.72 %.

2.4.2 Related Works on Indic Scripts

The following subsections comprehensively explains the technology or method used across different Indian lanaguages

MQDF based Techniques for Recognizing Handwritten Words

MQDF(Modified Quadratic Discriminant Function) is used to find the probability of a given character from its image or part of the image. Pal et al. [66] proposed a recognition scheme for Bangla handwritten city names .The method was lexicon specific and 84 city names considered for recognition. Characters are segmented from the words using water reservoir concept. Chaincode based features are fed to MQDF for classifiy the characters. Pal et al. [67] use the same approach for the recognition of English-Hindi-Bangla city names from the postal cards. The lexicon size for English is 89, Hindi was 117 and for Bangla it was 84. Thadchanamoorthy et al. [68] use the same approach to lexicon driven Tamil Handwritten city name recognition. Lexicon size was 265 and it contains city names with two-words and three-words.

Recognition of Handwritten Words Using Neural Network/SVM Based Techniques

Here we discuss the works related with MultiLayerPerceptron(MLP), Convolutional Neural Network(CNN) and Recurrent Architectures. Many of the works used MLP also use SVM, so those works are also mentioned here.

Bhowmik et al. [47] propose a combination of tetragonal, elliptical and vertical pixel density histogram based features with MLP and SVM as the classifiers for Bangla Handwritten Word Recognition with CMATERdb2.1.2 dataset.

Basu et al. [69] propose a zone based recognition for handwritten Bangla words. Data collected through specially designed form with a provision to enter upper,middle and lower zone of the word. The features used for recognition are longest-run, modified-shadow and octant-centroid. For the recognition of middle zone, two-stage approach is implemented.

Thadchanamoorthy et.al [70] presented a system for Tamil word recognition. Gabor filters are used to extract features from the images and with SVM it provides an accuracy of 86.36 % . The total sample size used for this experiment is 4270.

Das et al. [71] suggested feature selection using Harmony search based technique. Features used in this work is elliptical and experimented with seven classifiers and MLP provide the best result for the Bangla word recognition with a lexicon size of 20 and total sample size was 1020.

Sagheer et.al. [72] propose offline Urdu Handwritten image recognition using SVM as the classifier and the features used are gradient calculated using Robert's filter and projection profiles achieves an accuracy of 97%.

Malakar et.al [73] propose holistic resognition of Hindi words with a lexicon size of 33 with total sample size of 4620. Geometric and directional features extracted from image and sub part of it with MLP as the classifier provides an accuracy of 90.78 %

Mukhtar et.al [74] have achived an accuracy of 75 % with Urdu handwritten word samples of 1600 handwritten words contributed by two writers. They use gradient features calculated using sobel operator, structural features use the curvature and concavity.

Shaw et.al [75] presented a system for the recognition of Devanagari words with Gradient, Structural and concavity features from the skeleton of the image and contour based features (chain code histogram). SVM with linear kernel provides an accuracy of 81.14%.

Karthik et.al [76] use HOG(Histogram of Oriented Gradient) afer the segmentation module for Kannada handwritten text recognition. Classifier used in this experiment is SVM.

Adak et.al [49] presented convolutional-recurrent neural network architecture for the recognition of Bangla words.

Paneri et.al [77] presented recogntion of Gujarati words with a dataset of 2700 samples and 10 lexicons. HOG features provide an accuracy of 85.87 % with SVM as the classifier.

Dutta et.al [78] use convolutional-recurrent neural network architecture for the recognition of Hindi and Bangla words. Network trained using the synthetic data created using different fonts and the system is finally implemented using RoyDB dicussed in subsection 2.3.2. Lexicon based decoding provides an accuracy of 95.7%, 95.38% for Bangla and Devanagari respectively.

Hidden Markov Model (HMM) based transcription methods

Bhoi et.al [79] explored Odia text recognition with a total of 4000 samples with 500 lexicon. HMM is used for sequential classification, where the sequence can any of the 289 syllables found in the dataset considered in this experiment.

Roy.et.al [80] propose a hierarchical based recognition method by dividing the Bangla and Devanagari words segment to top, middle and bottom parts. Pyramidal Histogram of Oriented Gradient features are used for classification. In the middle zone features are extracted using sliding window and fed to HMM for classification in a sequence manner. Top and bottom zones are resized to a fixed size, then SVM is used for classification. Total word recognition accuracy for Bangla is 85.49 % and Devanagari is 86.14 %.

Vajda. et.al [81] presented a system for postal document recognition, where the contents are written in Bangla and English in a mixed manner. After removing the postal stamp part from the image, address part is detected. To separate the scripts viz. Bangla and English, water reservoir concept[82, 83] is used. In addition to that, these two scripts are differentiated using Matra features, which are present only in Bangla. The recognition of the address part consists of alphabets from Bangla, English and numerals. Two stage classfication method is used for numeral classfication. In the first stage it classified all the English and Bangla numeral together, second stage it classified Bangla and English digits separately. MLP is used for classification. Recognition of the words are performed through Non-Symmetric Half Plane Hidden Markov Model(NSHPHMM), achieved an accuracy of 86.80 % accuracy.

Bhowmik et.al [84] use genetic algorithm to optimize HMM with chain code features provide 79.12 % accuracy for Bangla town names.

Parui et.al [85] propose a holistic approach for the recognition of Devanagari words using the stroke based features. With a lexicon size of 50 and HMM as the calssifier provides an accuracy of 82.89 %. As an extension of this work, Shaw et.al [44] proposed segmentation based method for a lexicon size of 100. HMM provides an accuracy of 81.63 %.

Other Methods

Jayadevan et.al [45] proposed recognition of Hindi and Marathi words extracted from simulated bank cheque forms. Features used are gradient, structural and concavity with Binary Vector Matching provide good results.

Ramachandrula et.al [86] present a method for Hindi word recogntion using Directional Element Features. The similarity beween the word to be recognized and lexicon are performed through dynamic programming.

Major works in Urdu,Marathi,Kannada and Odiya Word recogntion is shown in Table: 2.1 Advancement towards Bangla word recognition is

Table 2.1: Major works in Urdu, Marathi, Kannada and Odiya

<table>
<tr><th colspan="5">Urdu</th></tr>
<tr><td>Lexicon size</td><td>Author</td><td>Features</td><td>classifier</td><td>Accuracy</td></tr>
<tr><td>57</td><td>Sagheer et.al [2010]</td><td>Gradient, Projection Profile</td><td>SVM</td><td>96.02</td></tr>
<tr><td>100</td><td>Mukthar et.al[2009]</td><td>Gradient, Structural Concavity</td><td>SVM</td><td>75</td></tr>
<tr><th colspan="5">Marathi</th></tr>
<tr><td>Lexicon size</td><td>Author</td><td>Features</td><td>classifier</td><td>Accuracy</td></tr>
<tr><td>114</td><td>Jayadevan et.al [2011]</td><td>Gradient Structural Concavity</td><td>Binary Vector Matching</td><td>85.78</td></tr>
<tr><td>114</td><td>Jayadevan et.al [2011]</td><td>Gradient Structural Concavity</td><td>Binary Vector Matching</td><td>84.61</td></tr>
<tr><th colspan="5">Kannada</th></tr>
<tr><td>Lexicon size</td><td>Author</td><td>Features</td><td>classifier</td><td>Accuracy</td></tr>
<tr><td>*</td><td>Karthik et.al[2016]</td><td>HOG</td><td>SVM</td><td>95.02</td></tr>
<tr><th colspan="5">Odiya</th></tr>
<tr><td>Lexicon size</td><td>Author</td><td>Features</td><td>classifier</td><td>Accuracy</td></tr>
<tr><td>500</td><td>Bhoi. et.al[2015]</td><td>Concavity</td><td>HMM</td><td>64.82</td></tr>
</table>

summarized in Table:2.2

Advancement towards Hindi Word recognition is shown in Table: 2.3.

Here the advancement refers to the improvements in feature selection methods,lexicon size, isolated character recognition to word recognition, technology used and accuracy. Recognition in Devanagari script has advanced a lot, especialy Bangla and Hindi languages. In Bangla lexicon size has increased from 119 to 1547, different feature selection algorithms like deep learning based methods are also experimented with and they produced results having good accuracy in recent years. In Hindi lexicon size has increased from 50 to 1957 and accuracy also improves from 82.78% to 90.78%.

Table 2.2: Advancement Towards Bangla Word Recognition

Lexicon Size	Author	Features	Classifier	Accuracy
120	Bhowmik et.al[2018]	Shape based-elliptical, Tetragonal, Vertical pixel density	MLP	79.87
1547	Dutta et.al[2018]	CNN Extracted	Recurrent Architecture	95.7
1547	Adak et.al[2016]	CNN Extrcated-Character level	Recurrent Architecture	85.42
815	Adak et.al[2016]	CNN Extracted-Character level	Recurrent Architecture	86.96
1547	Roy et.al[2016]	PHOG	HMM and SVM	83.39
20	Das et.al[2016]	Elliptical	MLP	90.29
127	Basu et.al[2009]	longest run, shadow, octant-centroid	MLP	80.58
76	Vajda et.al[2009]	Pixels	NSHP-HMM and MRF	86.8
84	Pal et.al[2009]	Chaincode	MQDF	94.08
119	Bhowmik et.al[2008]	Shape based	HMM	79.12

Table 2.3: Advancement Towards Hindi Word Recognition

Lexicon Size	**Author**	**Features**	**Classifier**	**Accuracy**
33	Malakar et.al[2017]	Geometric, Directional	MLP	90.78
1957	Roy et.al[2016]	PHOG	HMM & SVM	84.24
100	Shaw et.al[2014]	Gradient, Structural, Concavity	SVM	81.14
84	Pal et.al[2012]	chaincode	MQDF	90.16
30	Ramachandrula et.al[2012]	Directional Element	Dynamic Programmming algorithm	79.94
106	Jayadevan et.al[2011]	Grdaient, Structural, Concavity	Binary Vector Matching	83.07
100	Shaw et.al[2008]	Stroke based	HMM	81.63
50	Parui et.al[2007]	Stroke based	HMM	82.89

In Malayalam isolated character recognition achieves 99.78 % accuracy.

Major works in Malayalam character recognition are shown in Table: 2.4. To the best of our knowledge there is no major work reported for Malayalam offline handwritten word recognition.

Table 2.4: Advancement Towards Malayalam Character Recognition

Methodology	Features	Classifier	Dataset	Accuracy
Jomy et.al [2014][87]	Gradient	SVM	18000/ 90 classes	97.72
Raju et.al [2014][88]	Gradient RLC Centroid	MLP	19800/ 44 classes	99.78
Binu P et.al [2011][89]	Division Point	SVM	49 classes	95.36
Lajish et.al [2008][90]	State Space Point Distribution	MLP	44 classes	73.03

2.5 Existing Supplementary Tools

Kaldi Toolkit

This toolkit [91] can be used for the decoding from the observations of RNN/HMM with a language model. Both Windows and Linux versions are available.

OpenFst

Weighted Finite State Transducers(WFST) are useful in Word Segmentation, Word Recognition, Language Model especialy in decoding.OPenFst [92] is a library to implement WFST.

SRILM

SRI Language Modeling(SRILM) [93] toolkit is used for creating the language models.

Eesen

Eesen toolkit[94] originally developed for automatic speech recognition in end-to-end manner. It is useful for decoding purpose after the optical model in Handwriting recognition.

RETAS

RETAS is a text alignment scheme originally designed for the word alignment of scanned books [95] [96].

HTK

HTK(HMM toolkit) [97] can use for implementing Hidden Markov Model.

RNNLIB

RNNLIB [98] is designed by AlexGraves for the implementation of LSTM networks for online/offline handwriting recognition.

2.6 Summary

The overall word recognition method found in literature is shown in Figure 2.1

Lot of research works are happened in non-Indic scripts, where in Indic scripts only Bangla and Devanagari produce fruitful results. The major

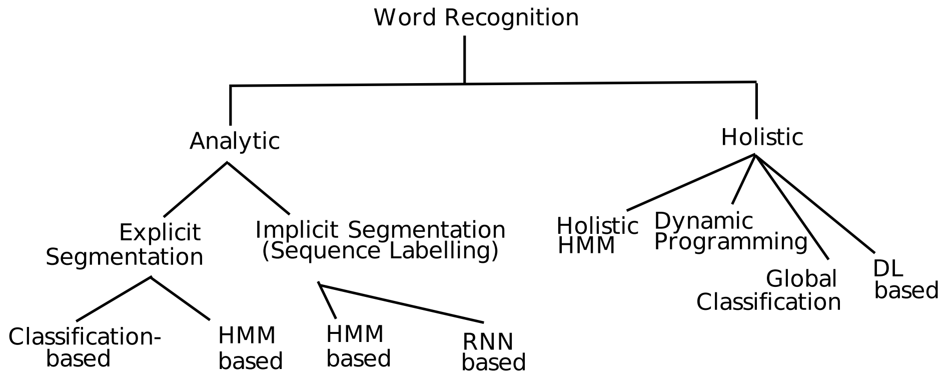


Figure 2.1: Handwriting Recognition Methods

reason may be the lack of dataset. So this survey explains in detail about the useful datasets and toolkit avialable in the web repository.

Chapter 3

The New Dataset-JMHRDB

Amazing things happen when you pull individual pieces of information together into larger linked datasets: meaning emerges, as you produce facts from figures.

Ben Goldacre

3.1 Introduction

Data is the important factor in all machine learning applications. In Malayalam there is no well known or standard dataset available for offline handwritten word/document recognition. This chapter explains about the new handwritten Malayalam database. JMHRDB is an abbreviation of Jino Malayalam Handwriting Recognition Data Base, which consists of a dataset of words(JMHRDB1) and another dataset of birth certificate application forms(JMHRDB2). It is suitable for word/ document level recognition for the purpose of research/ commercial design of the applications.

3.2 Malayalam Script: An Overview

Malayalam is one of the twenty two scheduled languages of India and was declared as a classical language by the Government of India in 2013 [99]. Spoken by around 35 million people, it is the official language of Kerala province, union territories of Lakshadweep and Puducherry. The script in which the language is written is also called Malayalam and it is one among the ten major official Indian scripts. It is derived from Grantha, an inheritor of ancient Brahmi script. Malayalam script is alpha-syllabic and non-cursive in nature. Its basic character set consists of vowels and consonants. Each vowel has an independent form and another dependent form except the first vowel ("അ/a/") which has no corresponding dependent form. Dependent vowel signs do not appear on their own. Such a vowel appears as a diacritic attached to a consonant. These are composed of one or more glyph pieces which appear either to the left, right or both sides of a consonant.

A new reformed Malayalam script was introduced in the year 1971 by an order of Kerala Government on the basis of recommendations made by Committees formed for this purpose. In this reformed script, the number of glyphs had been reduced substantially for the ease of printing and typewriting. The basic reforms include (i) substitution of irregular ligatures by corresponding sequences of basic glyphs and (ii) replacement of severely complex shaped conjucts (combinations of multiple characters) by the corresponding sequences of separated basic characters and diacritic [100]. Although textbooks recommended by Govt. sponsored schools follow this new script, in practice, this is followed only partially in daily writing in Malayalam. In the present scenario, both printed and handwritten texts appear to have a mixture of characters from both old and new scripts. It makes their automatic recognition a more difficult task.

3.3 JMHRDB1 lexicon

Offline handwritten Malayalam word sample dataset developed as a part of the present study is based on a lexicon consisting of 241 city (Panchayath) names and another 89 words selected from Sabdatharavali Malayalam Dictionary. List of lexicons is shown in table 3.1 and table 3.2

Lengths of words of the present lexicon vary widely – its shortest words consist of 2 characters while its longest word has 15 characters. Frequencies of words versus their lengths in this lexicon are shown in Figure 3.1.

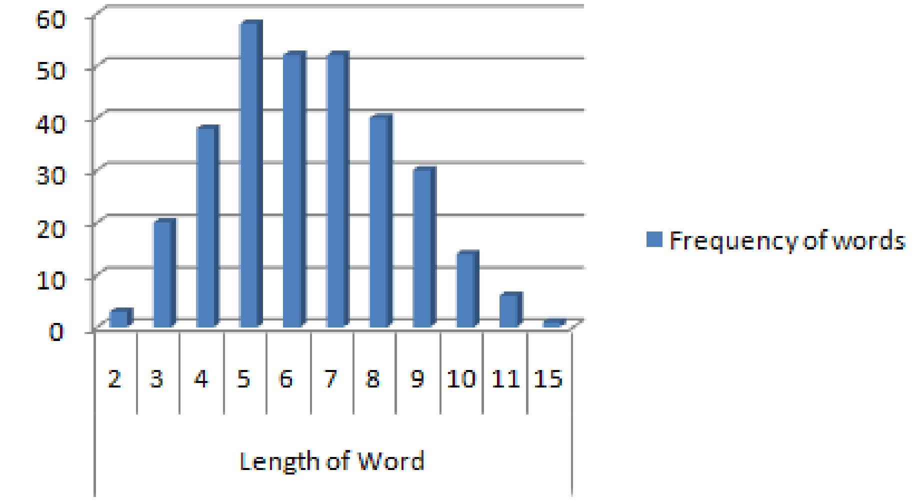


Figure 3.1: Character Length Versus words

According to a treatise of Malayalam grammar [101], the new script has 97 symbols: 36 consonants, 5 pure consonants, 13 vowels, 13 dependent vowels, 4 consonant signs and 26 compound characters. The present lexicon includes all of these symbols. In addition to this, words of the present lexicon set include 13 compound characters of the traditional (old)

Table 3.1: Lexicon with Class ID(1-168)

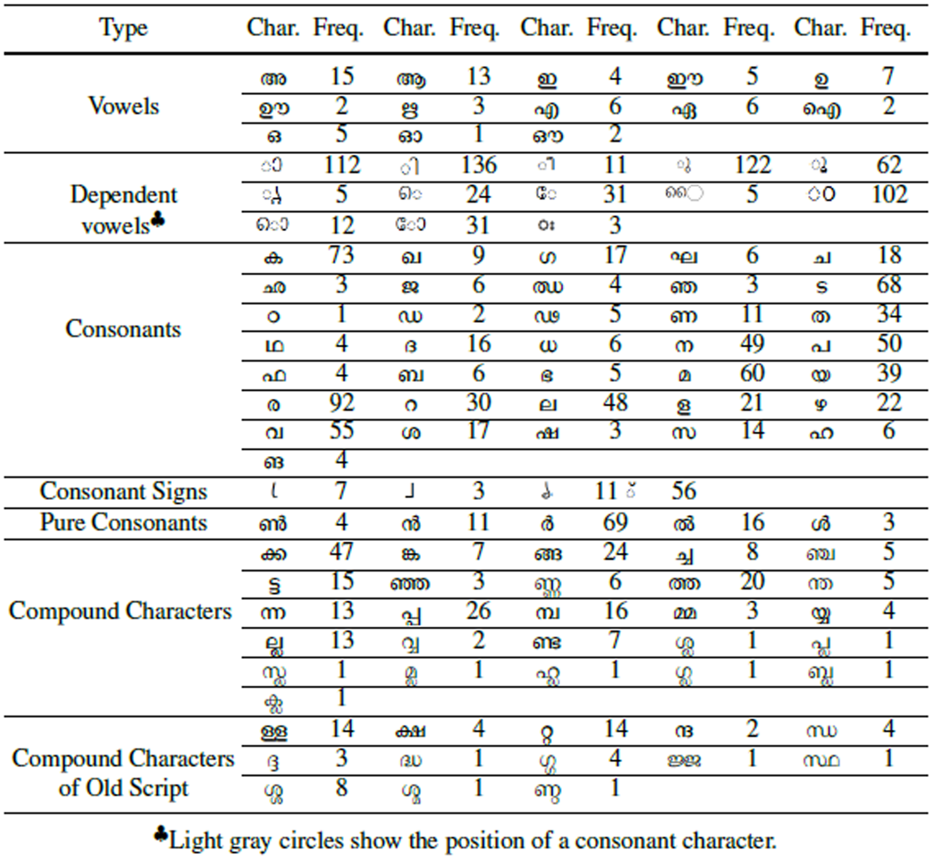
1	അടൂർ	2	ആലങ്ങാട്	3	ആലപ്പാട്	4	അലയമൺ
5	ആലുവ	6	ആനക്കര	7	അഞ്ചുതെങ്ങ്	8	ആനിക്കാട്
9	ആരക്കുഴ	10	അരിക്കുളം	11	അരൂക്കുറ്റി	12	അരുവാപ്പുലം
13	ആര്യങ്കാവ്	14	അതിരപ്പിള്ളി	15	ആറ്റിങ്ങൽ	16	ആവോലി
17	അയിലൂർ	18	അയ്യമ്പുഴ	19	അഴീക്കോട്	20	അഴുത
21	ബാലുശ്ശേരി	22	ഭരണിക്കാവ്	23	ചടയമംഗലം	24	ചാലിശ്ശേരി
25	ചങ്ങരോത്ത്	26	ചാവക്കാട്	27	ചെക്യാട്	28	ചെല്ലാനം
29	ചെമ്പ്	30	ചെങ്ങന്നൂർ	31	ചെറിയമുണ്ടം	32	ചെറുന്നിയൂര്
33	ചെറുവണ്ണൂർ	34	ചിങ്ങോലി	35	ചിറ്റാർ	36	ചിറ്റൂർ
37	ചിറയിൻകീഴ്	38	ചൊവ്വന്നൂർ	39	കൊച്ചി	40	ധർമ്മടം
41	എടക്കാട്	42	ഇടമുളയ്ക്കൽ	43	എടപ്പറ്റ	44	ഇടവ
45	എടവിലങ്ങ്	46	ഇളമാട്	47	എളങ്കുന്നപ്പുഴ	48	എളവള്ളി
49	ഏറത്ത്	50	എരുമപ്പെട്ടി	51	ഇരിമ്പിളിയം	52	കടലുണ്ടി
53	കടമ്പഴിപ്പുറം	54	കതിരൂർ	55	കടുത്തുരുത്തി	56	കല്ല്യാശ്ശേരി
57	കല്ലൂർക്കാട്	58	കാഞ്ചിയാർ	59	കങ്ങഴ	60	കണിയാമ്പറ്റ
61	കണ്ണമ്പ്ര	62	കാരാകുറുശ്ശി	63	കരിങ്കുന്നം	64	കരുംകുളം
65	കാസർകോട്	66	കട്ടപ്പന	67	കവളങ്ങാട്	68	കയ്യൂർ
69	കീഴ്മാട്	70	കിഴക്കമ്പലം	71	കൊടംതുരുത്ത്	72	കൊടുവള്ളി
73	കൊല്ലയിൽ	74	കോങ്ങാട്	75	കൂരാച്ചുണ്ട്	76	കൂട്ടിലങ്ങാടി
77	കോതമംഗലം	78	കോട്ടാങ്ങൽ	79	കോട്ടയം	80	കോട്ടുകാല്
81	കൊയിലാണ്ടി	82	കുലുക്കല്ലൂർ	83	കുമ്പളം	84	കുഞ്ഞിമംഗലം
85	കുറുമാത്തൂർ	86	കുത്തന്നൂർ	87	കുറ്റ്യാടി	88	കുറ്റിക്കോൽ
89	കുഴൽമന്ദം	90	മടവൂർ	91	മടിക്കൈ	92	മലപ്പുറം
93	മമ്പാട്	94	മംഗലം	95	മാണിക്കൽ	96	മഞ്ഞള്ളൂർ
97	മഞ്ചേശ്വരം	98	മാങ്കുളം	99	മാറാക്കര	100	മാരാരിക്കുളം
101	മരിയാപുരം	102	മാട്ടൂൽ	103	മയ്യനാട്	104	മീനച്ചിൽ
105	മേലടി	106	മേലില	107	മേപ്പയൂർ	108	മൊകേരി
109	മുന്നിയൂർ	110	മൊറയൂർ	111	മുഹമ്മ	112	മുളന്തുരുത്തി
113	മുണ്ടേരി	114	മൺറോതുരുത്ത്	115	മുത്തോലി	116	മുട്ടാർ
117	മൂവാറ്റുപുഴ	118	മൈനാഗപ്പള്ളി	119	നടുവിൽ	120	നല്ലേപ്പിള്ളി
121	ഞാറക്കൽ	122	നരിക്കുനി	123	നെടുമങ്ങാട്	124	നെടുമ്പ്രം
125	നേമം	126	നെയ്യാറ്റിൻകര	127	വടക്കൻ പറവൂർ	128	ഒളവണ്ണ
129	ഒഞ്ചിയം	130	ഒറ്റൂർ	131	പടിയൂർ	132	പായിപ്പാട്
133	പാലക്കുഴ	134	പള്ളിച്ചൽ	135	പള്ളിക്കര	136	പള്ളിവാസൽ
137	പാമ്പാടി	138	പനങ്ങാട്	139	പനയം	140	പാങ്ങോട്
141	പന്ന്യന്നൂർ	142	പാപ്പിനിശ്ശേരി	143	പരപ്പ	144	പാറശ്ശാല
145	പരുതൂർ	146	പട്ടാഴി വടക്കേക്കര	147	പവിത്രേശ്വരം	148	പീരുമേട്
149	പേരയം	150	പെരിങ്ങോട്ടുകുറിശ്ശി	151	പെരുമാട്ടി	152	പെരുമ്പളം
153	പിറവന്തൂർ	154	പോരൂർ	155	പോത്തുകല്ല്	156	പുതുക്കാട്
157	പുളിക്കൽ	158	പുൽപ്പറ്റ	159	പുന്നയൂർ	160	പുറമേരി
161	പുത്തൻ വേലിക്കര	162	പുഴയ്ക്കൽ	163	രാമമംഗലം	164	റാന്നി അങ്ങാടി
165	ശാസ്താംകോട്ട	166	ഷൊർണ്ണൂർ	167	ശ്രീമൂലനഗരം	168	താനാളൂർ

Table 3.2: Lexicon with Class ID(169-330)

169	തകഴി	170	തലപ്പലം	171	തലയാഴം	172	തണ്ണീർമുക്കം
173	തെക്കുംകര	174	തെന്നല	175	തിരുമാറാടി	176	തിരുവാലി
177	തിരുവനന്തപുരം	178	തിരുവാർപ്പ്	179	തൊടിയൂർ	180	തൃക്കലങ്ങോട്
181	തുമ്പമൺ	182	തിരൂരങ്ങാടി	183	തൃപ്രങ്ങോട്ടൂർ	184	തുവ്വൂർ
185	വടക്കേക്കാട്	186	വക്കം	187	വള്ളത്തോൾ നഗർ	188	വാമനപുരം
189	വാണിയംകുളം	190	വാരപ്പെട്ടി	191	വാടാനപ്പള്ളി	192	വാഴയൂർ
193	വെച്ചൂച്ചിറ	194	വെളിനല്ലൂർ	195	വെള്ളറട	196	വെള്ളിനേഴി
197	വേളൂക്കര	198	വെങ്ങാനൂർ	199	വേങ്ങൂർ	200	വിജയപുരം
201	വോർക്കാടി	202	വണ്ടൂർ	203	വിതുര	204	ഏരൂർ
205	ഉള്ളൂർ	206	കുളത്തൂർ	207	പെരുമ്പാവൂർ	208	ചേരാനല്ലൂർ
209	കൈനകരി	210	കൈപ്പറമ്പ്	211	കൈപ്പമംഗലം	212	കൊല്ലം
213	കൊടുവായൂർ	214	കൊടുങ്ങല്ലൂർ	215	ചിറക്കാക്കോട്	216	കോഴിക്കോട്
217	പുന്നപ്ര	218	തൃപ്രയാർ	219	തൃശ്ശൂർ	220	പേരാമ്പ്ര
221	ആദിച്ചനല്ലൂർ	222	അജാനൂർ	223	ഐക്കരനാട്	224	ആലപ്പുഴ
225	ആറന്മുള	226	ആര്യങ്കോട്	227	ബഡിയടുക്ക	228	ബളാൽ
229	ബേളൂർ	230	ബുധനൂർ	231	ഏലപ്പാറ	232	ഏലൂർ
233	ഏഴിക്കര	234	കുലശേഖരപുരം	235	മുഖത്തല	236	ഒറ്റശേഖരമംഗലം
237	ഏറ്റുമാനൂർ	238	അഞ്ചൽ	239	ഓച്ചിറ	240	ഒല്ലൂക്കര
241	ഉദയഗിരി	242	ഉദയപുരം	243	ഫാത്തിമ	244	ഝകാരം
245	ഝങ്കാരം	246	ഝടഝട	247	ഘടകകക്ഷി	248	ഘടകം
249	ഘടന	250	ഘടി	251	ഘനം	252	ഘനജലം
253	ഛരഗണം	254	ഛന്ദം	255	ഛർദ്ദി	256	ഢം
257	ഢക്ക	258	ഢക്കരി	259	നാഥൻ	260	ഖദർ
261	ഖജനാവ്	262	ഖരം	263	asthi അസ്ഥി	264	ഫക്കീർ
265	ഫലം	266	ഫലകം	267	ഉദരംഭരി	268	ഉദാഹരണം
269	ഉദ്ദണ്ഡൻ	270	ഉദയസന്ധ്യ	271	ഈട്ടി	272	ഈണം
273	ഈന്ത	274	ഈന്തപ്പന	275	ഈയൽ	276	കഥ
277	ഊഢ	278	ഊൺ	279	ഋണം	280	ഋജുരേഖ
281	ഋശ്യശൃംഗൻ	282	ഐശ്വര്യം	283	ഔദാര്യം	284	ഔജസ്യം
285	ക്ഷണിതാവ്	286	ക്ഷണം	287	ക്ഷാമബത്ത	288	ഞങ്ങൾ
289	ഞാഞ്ഞൂൾ	290	ദുർദ്ദശ	291	ദുർജ്ജനം	292	ദുർഗ്ഗ
293	ദുർഗ്ഗുണം	294	ദുർഗ്ഗന്ധം	295	പാഠം	296	ഡസൻ
297	ധനികൻ	298	ധരാധരം	299	ധർമപഥം	300	ശോഭ
301	ശോഭന	302	ശോഭനം	303	ശോഷ	304	സങ്കീർണം
305	സർഗ്ഗം	306	സംഹാരം	307	സംഹിത	308	സംഹാരമൂർത്തി
309	സുഗന്ധം	310	സുഗന്ധി	311	സുവിശേഷം	312	കഥകളി
313	യോദ്ധാ	314	ഗരുഢൻ	315	വാങ്‌മയം	316	പുനഃപുനഃ
317	ദുഃഖം	318	ശ്ലീഹ	319	പ്ലീനം	320	സ്ലാവിക്
321	മ്ലേച്ഛം	322	ആഹ്ലാദം	323	ഗ്ലാനി	324	ക്ലിപ്‌തം
325	ബ്ലോഗ്	326	ശ്‌മശാനം	327	ശുണ്ഠി	328	വാങ്‌മയ
329	വാങ്‌മുഖം	330	വാങ്‌മൂലം				

Malayalam script. The occurrence frequencies of these 110 symbols (can represent in unicode) in the present lexicon are shown in figure 3.2.

The set of first 20 characters occurring most frequently in our lexicon is in agreement with similar results presented in the study of [102].



Type	Char.	Freq.	Char.	Freq.	Char.	Freq.	Char.	Freq.	Char.	Freq.
Vowels	അ	15	ആ	13	ഇ	4	ഈ	5	ഉ	7
	ഊ	2	ഋ	3	എ	6	ഏ	6	ഐ	2
	ഒ	5	ഓ	1	ഔ	2				
Dependent vowels♣	◌ാ	112	◌ി	136	◌ീ	11	◌ു	122	◌ൂ	62
	◌ൃ	5	◌െ	24	◌േ	31	◌ൈ	5	◌ം	102
	◌ൊ	12	◌ോ	31	◌ഃ	3				
Consonants	ക	73	ഖ	9	ഗ	17	ഘ	6	ച	18
	ഛ	3	ജ	6	ഝ	4	ഞ	3	ട	68
	ഠ	1	ഡ	2	ഢ	5	ണ	11	ത	34
	ഥ	4	ദ	16	ധ	6	ന	49	പ	50
	ഫ	4	ബ	6	ഭ	5	മ	60	യ	39
	ര	92	റ	30	ല	48	ള	21	ഴ	22
	വ	55	ശ	17	ഷ	3	സ	14	ഹ	6
	ങ	4								
Consonant Signs	◌്യ	7	◌്ര	3	◌്വ	11 ◌്	56			
Pure Consonants	ൺ	4	ൻ	11	ർ	69	ൽ	16	ൾ	3
Compound Characters	ക്ക	47	ങ്ക	7	ങ്ങ	24	ച്ച	8	ഞ്ച	5
	ട്ട	15	ഞ്ഞ	3	ണ്ണ	6	ത്ത	20	ന്ത	5
	ന്ന	13	പ്പ	26	മ്പ	16	മ്മ	3	യ്യ	4
	ല്ല	13	വ്വ	2	ണ്ട	7	ശ്ല	1	പ്ല	1
	സ്ല	1	മ്ല	1	ഫ്ല	1	ഗ്ല	1	ബ്ല	1
	ക്ല	1								
Compound Characters of Old Script	ള്ള	14	ക്ഷ	4	റ്റ	14	ന്ദ	2	ന്ധ	4
	ദ്ദ	3	ദ്ധ	1	ഗ്ഗ	4	ജ്ജ	1	ന്ഥ	1
	ശ്ശ	8	ശ്മ	1	ഞ്ഞ	1				

♣Light gray circles show the position of a consonant character.

Figure 3.2: Malayalam Characters & Frequency in One form

3.4 Data Collection

Collection of offline handwriting data is relatively easy because it will get from scanning some existing forms. Some of the problems with this type of data collection is the difficulties to include all the characters and the dataset should balanced with frequency of the characters that occurs in common use. So we collected data in three ways from 1) Special Designed Forms 2) Birth Certificate Forms and 3) Redesigned Birth Certificate Forms. Specialy designed forms are shown in Figure 3.3. All the writers are asked to write the words given in the left column of the form to the right side. Enough space provided in each box to write the words in unconstrained manner. To identify variations in the handwriting the same writer was asked to fill two forms in different time period. Numbering Pattern Followed for the forms are $Userid$ and $Userid\_1$ as shown in the top left of figure: 3.3. Since we are distributing specially designed forms the ground truth of the handwritten words can be achieved from the form itself.

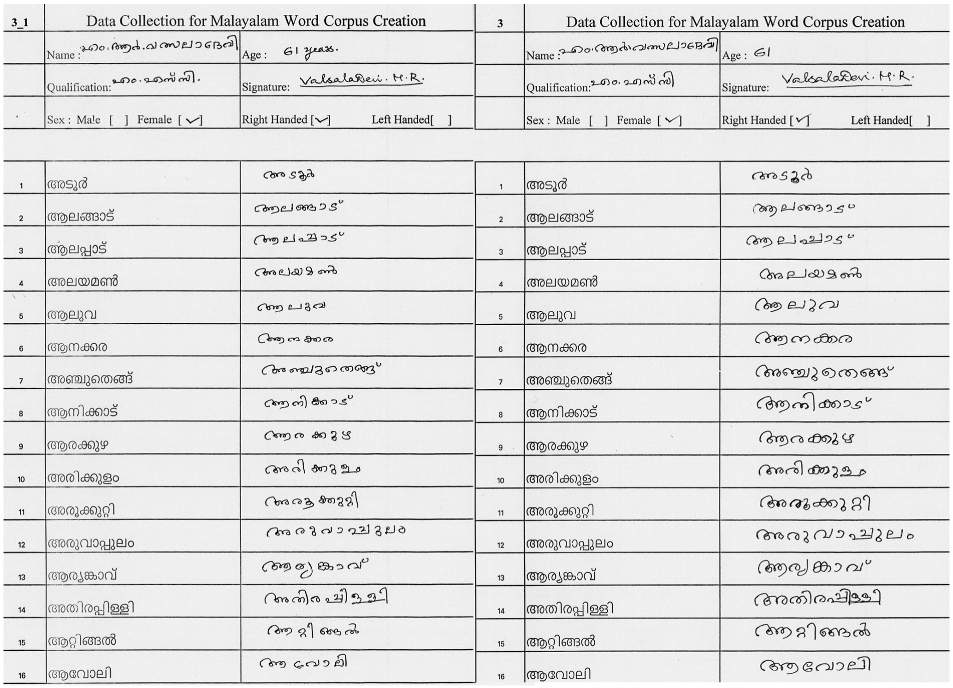
Data collected through distributing forms to people belongs to different age groups.

3.4.1 Form Processing

One of the most essential preprocessing steps is skew correction, if it is presented in the scanned document/ form images.

Skew Correction

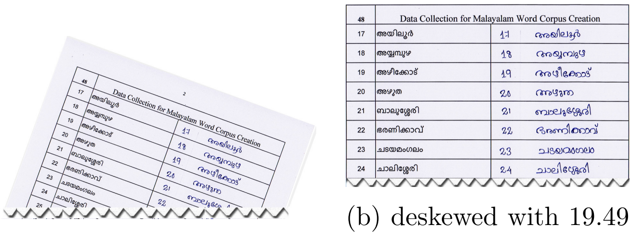
This is an important preprocessing step. Skewed forms while scanning are deskewed as shown in figure 3.4a and 3.4b.



3_1	Data Collection for Malayalam Word Corpus Creation	
	Name : എം.ആർ.വത്സലാദേവി	Age : 61 years.
	Qualification: എം.എസ്സി.	Signature: Valsaladevi. M.R.
	Sex : Male [ ] Female [✓]	Right Handed [✓] Left Handed[ ]
1	അടൂർ	അടൂർ
2	ആലങ്ങാട്	ആലങ്ങാട്
3	ആലപ്പാട്	ആലപ്പാട്
4	അലയമൺ	അലയമൺ
5	ആലുവ	ആലുവ
6	ആനക്കര	ആനക്കര
7	അഞ്ചുതെങ്ങ്	അഞ്ചുതെങ്ങ്
8	ആനിക്കാട്	ആനിക്കാട്
9	ആരക്കുഴ	ആരക്കുഴ
10	അരിക്കുളം	അരിക്കുളം
11	അരൂക്കുറ്റി	അരൂക്കുറ്റി
12	അരുവാപ്പുലം	അരുവാപ്പുലം
13	ആര്യങ്കാവ്	ആര്യങ്കാവ്
14	അതിരപ്പിള്ളി	അതിരപ്പിള്ളി
15	ആറ്റിങ്ങൽ	ആറ്റിങ്ങൽ
16	ആവോലി	ആവോലി

3	Data Collection for Malayalam Word Corpus Creation	
	Name : എം.ആർ.വത്സലാദേവി	Age : 61
	Qualification: എം.എസ്സി	Signature: Valsaladevi. M.R.
	Sex : Male [ ] Female [✓]	Right Handed [✓] Left Handed[ ]
1	അടൂർ	അടൂർ
2	ആലങ്ങാട്	ആലങ്ങാട്
3	ആലപ്പാട്	ആലപ്പാട്
4	അലയമൺ	അലയമൺ
5	ആലുവ	ആലുവ
6	ആനക്കര	ആനക്കര
7	അഞ്ചുതെങ്ങ്	അഞ്ചുതെങ്ങ്
8	ആനിക്കാട്	ആനിക്കാട്
9	ആരക്കുഴ	ആരക്കുഴ
10	അരിക്കുളം	അരിക്കുളം
11	അരൂക്കുറ്റി	അരൂക്കുറ്റി
12	അരുവാപ്പുലം	അരുവാപ്പുലം
13	ആര്യങ്കാവ്	ആര്യങ്കാവ്
14	അതിരപ്പിള്ളി	അതിരപ്പിള്ളി
15	ആറ്റിങ്ങൽ	ആറ്റിങ്ങൽ
16	ആവോലി	ആവോലി

Figure 3.3: Sample Form



48	Data Collection for Malayalam Word Corpus Creation		
17	അയിലൂർ	17	അയിലൂർ
18	അയ്യമ്പുഴ	18	അയ്യമ്പുഴ
19	അഴീക്കോട്	19	അഴീക്കോട്
20	അഴുത	20	അഴുത
21	ബാലുശ്ശേരി	21	ബാലുശ്ശേരി
22	ഭരണിക്കാവ്	22	ഭരണിക്കാവ്
23	ചടയമംഗലം	23	ചടയമംഗലം
24	ചാലിശ്ശേരി	24	ചാലിശ്ശേരി

(a) Skewness with -19.49 degree

(b) deskewed with 19.49 degree

Figure 3.4: Example for Skewed and Deskewed Documents

Word Extraction

Manual extraction of the line/ word/ characters from form is a time consuming process. So we automated the process of word extraction through program. All the extracted words are verified and corrected some of the samples manualy.

The schematic daigram of word extraction and labeling is shown in Figure 3.5

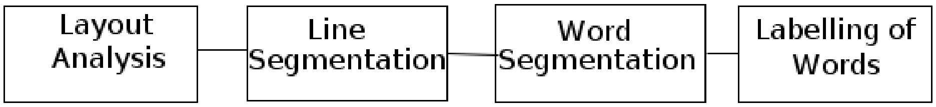


Figure 3.5: Schematic Daigram of Word Extraction and Labelling

3.4.2 Samples of JMHRDB1

Samples of JMHRDB-1 were collected from a group of 99 natives belonging to different sections of the population with respect to age, sex, education, profession and income. Writers of its samples had age in the range 10 to 60. The set of writers consists of both left handed and right handed ones. They were asked to write the words of a given lexicon set on a specific form printed on A4 size paper. Header part of this form was used to collect information about the writer such as name, age, qualification, signature etc. So the present database can be used for several other applications of handwriting analysis. There were 45 writers who wrote the samples from lexicon $1-314$ at two different points of time and the remaining 4 writers wrote it only once. Lexicon $314-330$ was written by another set of 50 writers, 44 writers wrote twice and 6 writers wrote once. The form was so designed that automatic extraction of individual word samples should be easy. Writers used their own pens. Filled-in forms were scanned using a flatbed scanner at 300 dpi. Automatically extracted samples were manually checked for necessary corrections. A few (20) word samples from the present database are shown in Figure 3.6. Each word sample of this figure belongs to a distinct word class. It consists of pairs of classes of similar shapes. As for example, the pair of words belonging to (1st row, 1st col) and (2nd row,

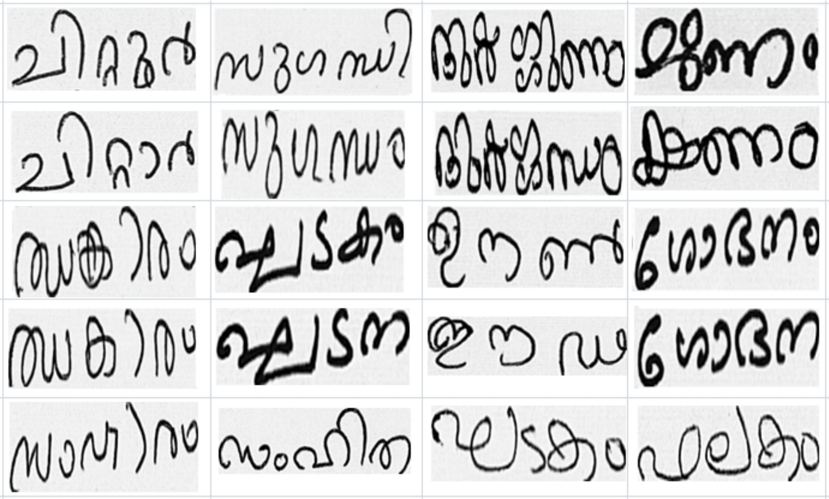


Figure 3.6: A few samples (each belonging to a distinct word class) from the present database – these provide a broad idea of the interclass similarity present in our database.

1st col) have similar shapes. Similarly, the pair of words belonging to (3rd row, 1st col) and (4th row, 1st col) looks similar.

This database consists of 31,020 handwritten Malayalam word samples. According to the study [103], the frequency of occurence of various characters and symbols in Malayalam text ranges from 9.19(ാ) to 0.00078(ഝ). According to the frequency of occurence 90^{th} percentile is the character and symbol is "മ്മ/mma" with frequency,$v = 0.18\%$

sample for proportions

$$n = \frac{Z^2 pq}{e^2} \tag{3.1}$$

Estimated proportion of an attribute, $p = 96.24\%$ $q = (1 - p) = 3.76\%$

with a 95% confidence level and ±5%

$n = \frac{1.96^2 * 0.9624 * 0.0376}{0.05^2} = 55.60522$

Total minimum sample size, $N = \frac{n}{v} = \frac{55.60522}{0.0018} = 30892$

so the minimum sample size required to cover atleast 90% symbols is 30892, so we select 31020 samples. Also we ensured that all the symbols from new reformed script is present.

The entire database is divided into training and test sets. Samples provided by 60 writers forms its training set while the samples of remaining 39 writers forms the test set. The training and test sets consist of 19,800 and 11,220 samples respectively.

Total number of writers contributed to the entire dataset is 199, Number of writers contributed to JMHRDB1 is 99. The percentage of writes of samples is shown in Table 3.3 and Table 3.4.

Table 3.3: Percentage of writes in JMHRDB1

Number of Writers	Samples	Percentage of writes (%)
45	28260	91.10
4	1256	4.04
44	1408	4.53
6	96	0.309

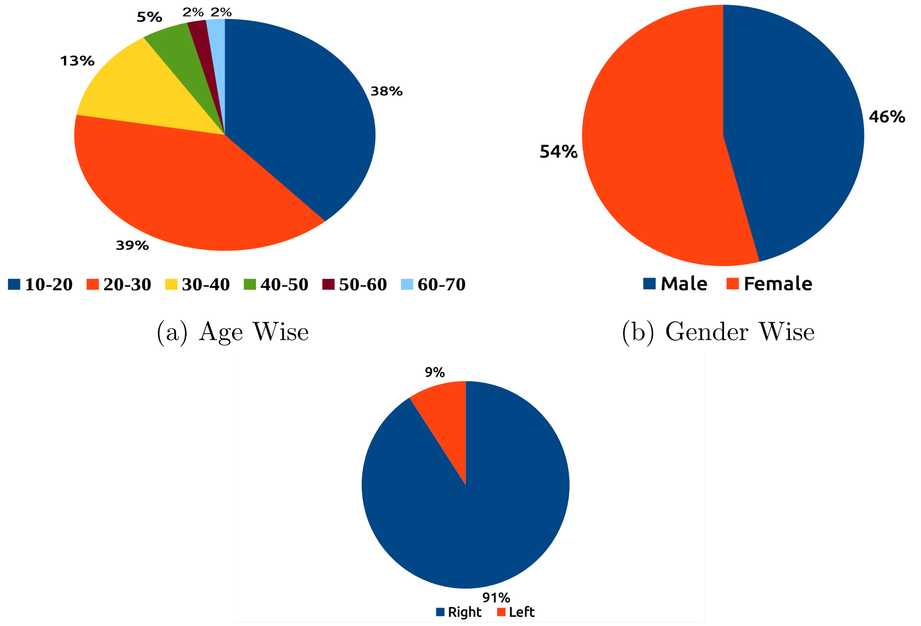
Table 3.4: Percentage of writes in JMHRDB2

Number of Writers	Number of Forms	Type of Forms	Percentage of writes (%)
50	50	Original	50
50	50	Redesigned	50

3.4.3 Details of JMHRDB1

Analysis of the Dataset

The distribution of data in Age,Gender,Handedness is shown in figure:3.7a,3.7b and 3.7c



(a) Age Wise

(b) Gender Wise

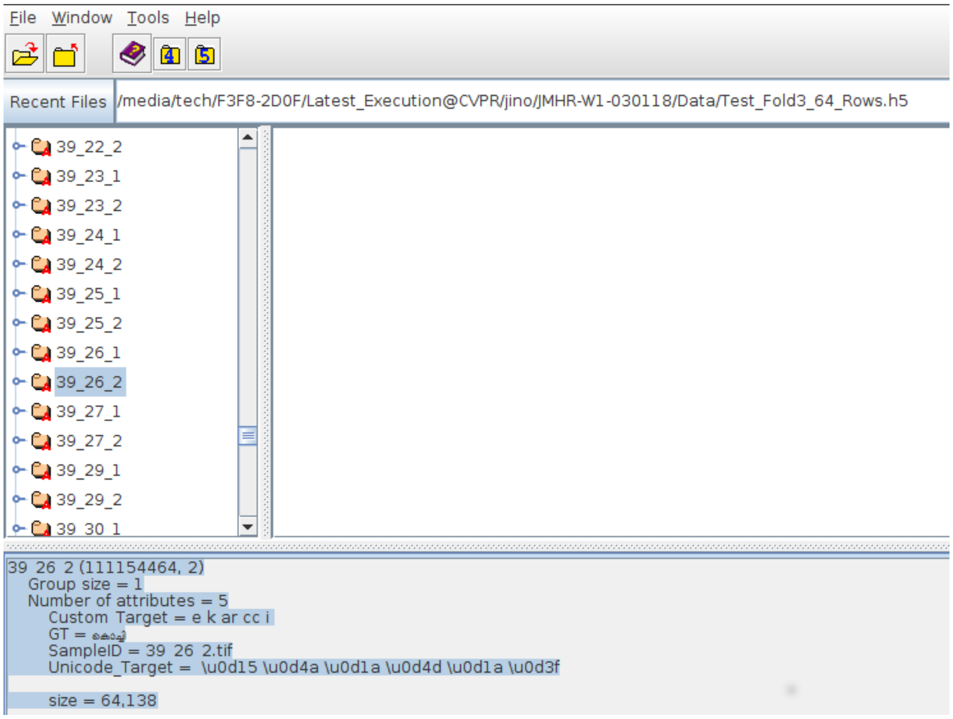
(c) handedness

Figure 3.7: Analysis on Dataset

Ground Truth

Gound truth is essential for the dataset, all the filenames are annotated with its unicode, malayalam, grapheme level and filename as shown in Figure 3.8. Individual handwritten word image in the database comes with the follwing Ground Truth Information

- Malayalam Word in UTF-Encoding



```
/media/tech/F3F8-2D0F/Latest_Execution@CVPR/jino/JMHR-W1-030118/Data/Test_Fold3_64_Rows.h5
```

```
39_22_2
39_23_1
39_23_2
39_24_1
39_24_2
39_25_1
39_25_2
39_26_1
39_26_2
39_27_1
39_27_2
39_29_1
39_29_2
39_30_1
```

```
39 26 2 (111154464, 2)
   Group size = 1
   Number of attributes = 5
      Custom Target = e k ar cc i
      GT = കൊച്ചി
      SampleID = 39 26 2.tif
      Unicode_Target =  \u0d15 \u0d4a \u0d1a \u0d4d \u0d1a \u0d3f

      size = 64,138
```

Figure 3.8: Ground Truth of One Image in the Dataset

- Malayalam word as character sequence in two ways

  Grapheme Level

  Unicode level

- Number of words (handwritten images)

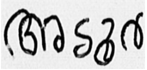
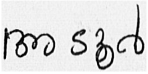
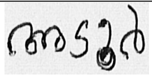
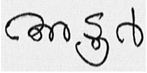
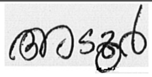
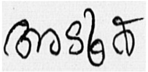
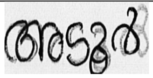
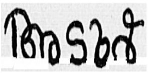
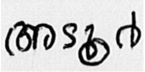
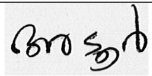
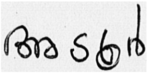
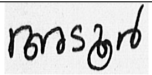
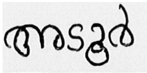
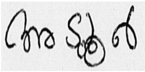
  classwise

  Train/Test statistics

- Author identifier, age, gender, profession (annotated manually)

Word sample “adoor/അടൂര്” written by various people with their personal details are shown in Table: 3.5

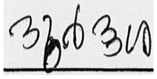
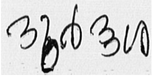
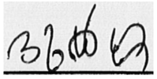
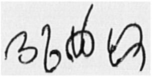
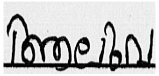
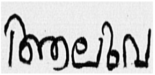
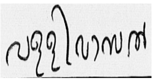
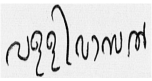
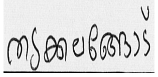
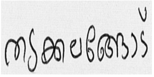
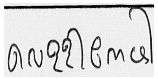
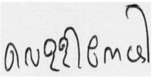
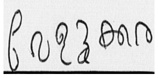
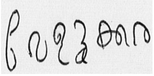
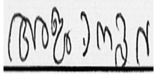
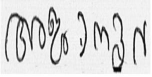
Table 3.5: Some Samples of "അടൂർ" From The Dataset with Personal Details

	Sahil C S,Age:17 SSLC,Male		Anupama, 31 MBA, Female University Assistant
	Valsala Devi,61 MSc, Female Retd.Govt.Employee		M K abdul Sathar,57 BA,Male LIC Agent
	Sanam K S,17 Plus Two,Female		Nahaz M Z,18 Plus Two, Male
	Varghese P,25 MBA,MVoc,Male Asst. Manager		Musammil,16 SSLC,Male
	Rose Merin Rens,15 SSLC,Female		Deepa,41 MSc,BEd HSA
	George Kutty M,57 MA, BEd,Male HSA		Arshad M H,10 Male
	Shyam Sundhar,28 MSc, Male Research Scholar		Shamitha K,48 SSLC,Female House Wife

Post Processing

Some of the samples required post processing are shown in Table 3.6 with their corrected version. For those samples the touching lines are extracted manualy.

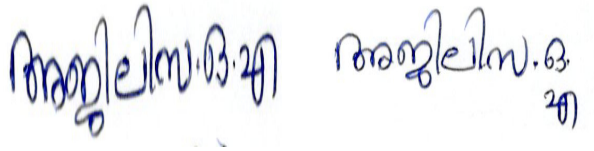
Table 3.6: Corrected Samples

Segmentation Required	Corrected	Segmentation Required	Corrected
			
			
			
			

3.5 Document Image Dataset-JMHRDB2

This dataset developed for the experimentation of application form recogntion. We selected birth certificate application form because its wide use in hospitals. Geometric structure of the application forms are redesigned for the purpose of automation. This dataset consists 50 form samples of actual and 50 form samples of redesigned forms.Thus a total of 100 writers contributed to this dataset. In the actual form, segmentation to the lines and words are a complex process. The presence of two scripts in the same form side by side make this process more difficult. Space to fill the details is the major constraint. So in the present scenario manual corrections are required in filled up forms. Samples of a handwritten word sample (name of the person) from the same document is shown in Figure 3.9a and Figure 3.9b.

These words are extracted from the form shown in Figure 3.10. The segmentation and correction of the data in these type of forms are done manually. So for the automation, we proposed a new redesigned form and collected data in it. With the redesigned form the automation was made easier and the segmentation process is also not complex. The processing

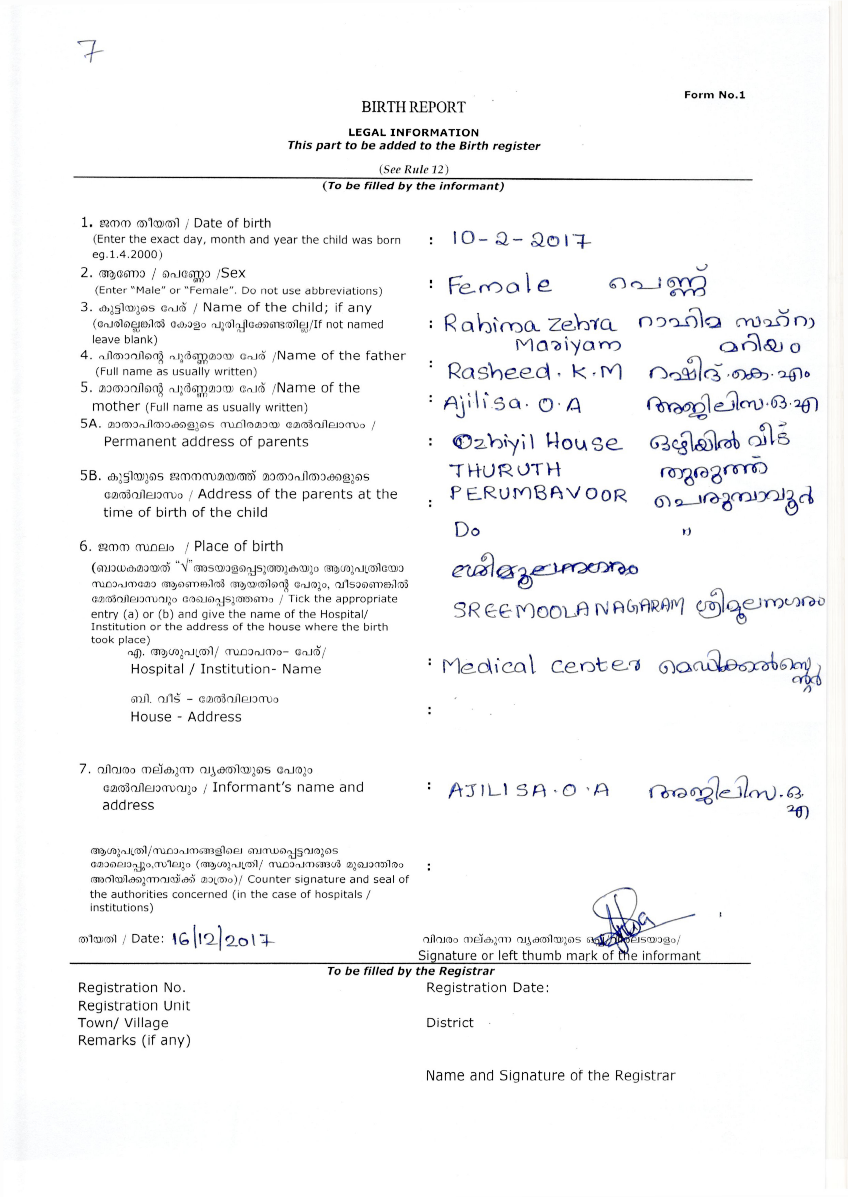


(a) Name of a Person in single line

(b) Name of person in two lines

Figure 3.9: Challenges- Segmentation and Recognition of Form Data

of the documents is explained in Chapter 7. Redesigned form is shown in Figure 3.11.



7

Form No.1

BIRTH REPORT

LEGAL INFORMATION
This part to be added to the Birth register

(*See Rule 12*)

(To be filled by the informant)

1. ജനന തീയതി / Date of birth (Enter the exact day, month and year the child was born eg.1.4.2000)	:	10-2-2017
2. ആണോ / പെണ്ണോ /Sex (Enter "Male" or "Female". Do not use abbreviations)	:	Female പെണ്ണ്
3. കുട്ടിയുടെ പേര് / Name of the child; if any (പേരില്ലെങ്കിൽ കോളം പൂരിപ്പിക്കേണ്ടതില്ല/If not named leave blank)	:	Rahima Zehra Mariyam റാഹിമ സഹ്റ മറിയം
4. പിതാവിന്റെ പൂർണ്ണമായ പേര് /Name of the father (Full name as usually written)	:	Rasheed. K.M റഷീദ്.കെ.എം
5. മാതാവിന്റെ പൂർണ്ണമായ പേര് /Name of the mother (Full name as usually written)	:	Ajilisa. O.A അജിലിസ.ഒ.എ
5A. മാതാപിതാക്കളുടെ സ്ഥിരമായ മേൽവിലാസം / Permanent address of parents	:	Ozhiyil House ഒഴിയിൽ വീട് THURUTH തുരുത്ത് PERUMBAVOOR പെരുമ്പാവൂർ
5B. കുട്ടിയുടെ ജനനസമയത്ത് മാതാപിതാക്കളുടെ മേൽവിലാസം / Address of the parents at the time of birth of the child	:	Do "
6. ജനന സ്ഥലം / Place of birth (ബാധകമായത് "√" അടയാളപ്പെടുത്തുകയും ആശുപത്രിയോ സ്ഥാപനമോ ആണെങ്കിൽ ആയതിന്റെ പേരും, വീടാണെങ്കിൽ മേൽവിലാസവും രേഖപ്പെടുത്തണം / Tick the appropriate entry (a) or (b) and give the name of the Hospital/ Institution or the address of the house where the birth took place)		ശ്രീമൂലനഗരം SREEMOOLANAGARAM ശ്രീമൂലനഗരം
എ. ആശുപത്രി/ സ്ഥാപനം- പേര്/ Hospital / Institution- Name	:	Medical center മെഡിക്കൽ സെന്റർ
ബി. വീട് - മേൽവിലാസം House - Address	:	
7. വിവരം നല്കുന്ന വ്യക്തിയുടെ പേരും മേൽവിലാസവും / Informant's name and address	:	AJILISA.O.A അജിലിസ.ഒ.എ
ആശുപത്രി/സ്ഥാപനങ്ങളിലെ ബന്ധപ്പെട്ടവരുടെ മോലൊപ്പും,സീലും (ആശുപത്രി/ സ്ഥാപനങ്ങൾ മുഖാന്തിരം അറിയിക്കുന്നവയ്ക്ക് മാത്രം)/ Counter signature and seal of the authorities concerned (in the case of hospitals / institutions)	:	

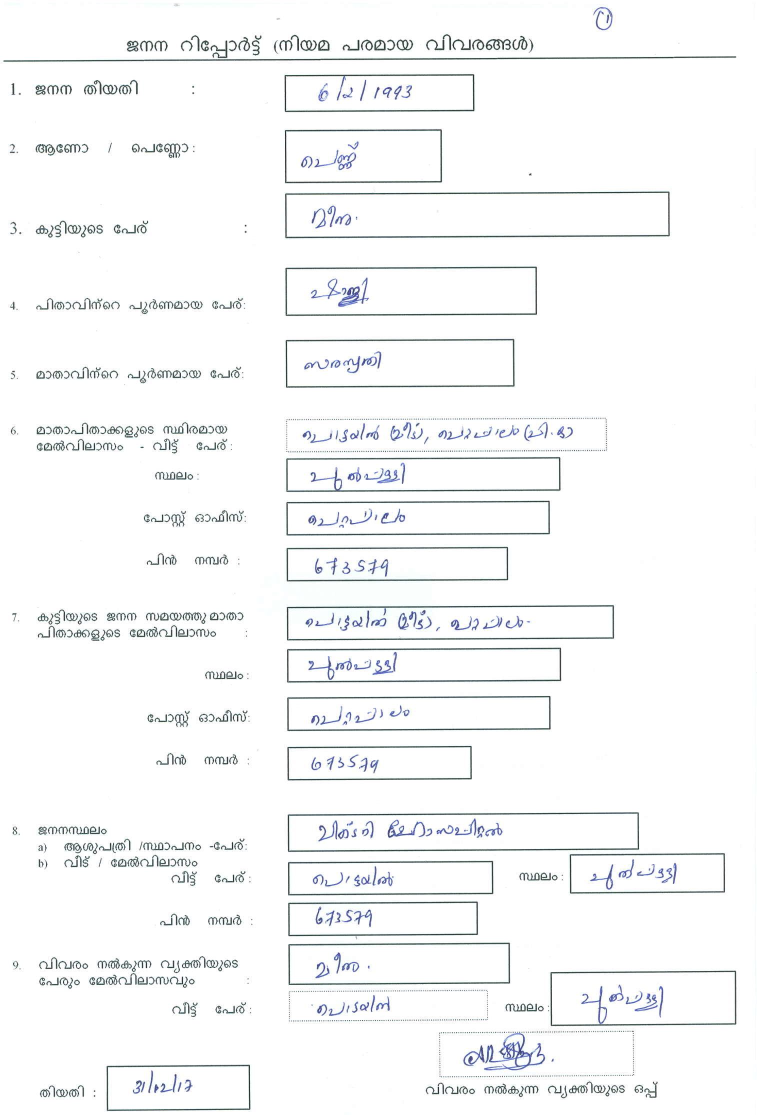
തീയതി / Date: 16|12|2017

വിവരം നല്കുന്ന വ്യക്തിയുടെ ഒപ്പ്/വിരലടയാളം/
Signature or left thumb mark of the informant

To be filled by the Registrar

Registration No.	Registration Date:
Registration Unit	
Town/ Village	District
Remarks (if any)	
	Name and Signature of the Registrar

Figure 3.10: Actual Form



1

ജനന റിപ്പോർട്ട് (നിയമ പരമായ വിവരങ്ങൾ)

1. ജനന തീയതി : 6/2/1993

2. ആണോ / പെണ്ണോ : പെണ്ണ്

3. കുട്ടിയുടെ പേര് : മീന.

4. പിതാവിന്റെ പൂർണമായ പേര്: ഉണ്ണി

5. മാതാവിന്റെ പൂർണമായ പേര്: സരസ്വതി

6. മാതാപിതാക്കളുടെ സ്ഥിരമായ മേൽവിലാസം - വീട്ട് പേര് : പൊട്ടയിൽ വീട്, ചെറ്റപ്പാലം (പി.ഒ)
സ്ഥലം : പുൽപ്പള്ളി
പോസ്റ്റ് ഓഫീസ്: ചെറ്റപ്പാലം
പിൻ നമ്പർ : 673579

7. കുട്ടിയുടെ ജനന സമയത്തു മാതാ പിതാക്കളുടെ മേൽവിലാസം : പൊട്ടയിൽ വീട്, ചെറ്റപ്പാലം.
സ്ഥലം : പുൽപ്പള്ളി
പോസ്റ്റ് ഓഫീസ്: ചെറ്റപ്പാലം
പിൻ നമ്പർ : 673579

8. ജനനസ്ഥലം
a) ആശുപത്രി /സ്ഥാപനം -പേര്: വിൻസന്റ് ഹോസ്പിറ്റൽ
b) വീട് / മേൽവിലാസം
വീട്ട് പേര് : പൊട്ടയിൽ സ്ഥലം : പുൽപ്പള്ളി
പിൻ നമ്പർ : 673579

9. വിവരം നൽകുന്ന വ്യക്തിയുടെ പേരും മേൽവിലാസവും : മീന.
വീട്ട് പേര് : പൊട്ടയിൽ സ്ഥലം : പുൽപ്പള്ളി

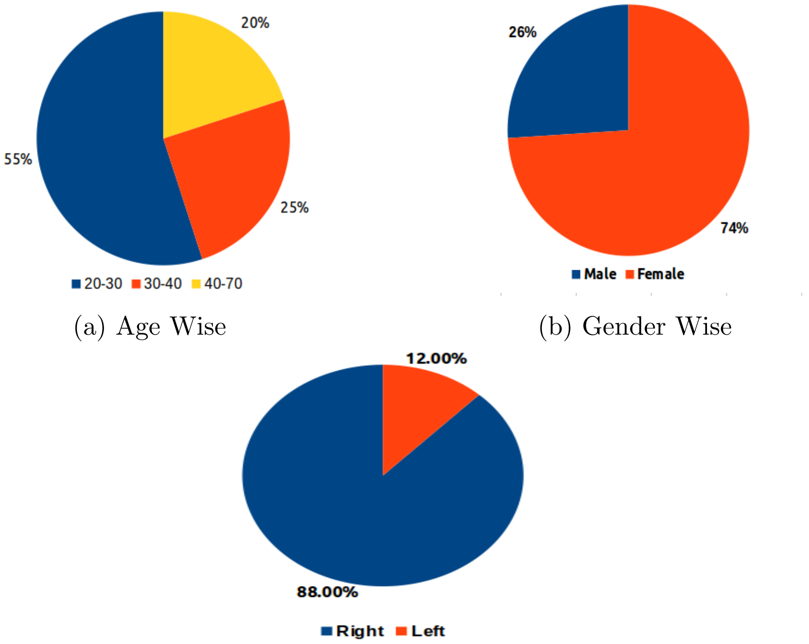
തിയതി : 3/02/17

വിവരം നൽകുന്ന വ്യക്തിയുടെ ഒപ്പ്

Figure 3.11: Redesigned Form

3.6 Analysis of JMHRDB2

Age,gender, handedness wise analysis of JMHRDB2 is shown in Figure: 3.12a,Figure: 3.12b and Figure: 3.12c



(a) Age Wise

(b) Gender Wise

(c) handedness

Figure 3.12: Analysis on JMHRDB2

3.7 Summary

In this chapter we discussed about the newly developed Handwritten Dataset of Malayalam Words and Birth Certificate Forms. Some of the unique characteristic of the dataset are 1) Ground truth is provided for all the extracetd word images 2)Specially designed Birth certificate form suitable for automatic data extraction is provided 3)Data available in “tif” image format and “hdf5” file format. Some of the data collected from the people who are staying outside Kerala for more than 10 years. 199 individuals contributed to this database. The dataset is available for further academic/non profitable research on a request basis.

Chapter 4

Lexicon Specific Recognition Using Deep Architecure

"The speed of change makes you wonder what will become of architecture."

Tadao Ando

4.1 Introduction

Deep learning composed of several architectures and the proliferation of the number of architectures viz. the commercial availability of Graphical Processing Units(GPU) in much affordable rates.These architectures consists of several layers with specific function and can be added in a hierarchical manner. If the depth of the layers increases, the architecture can be labelled as "Deep Architecture". Deep architectures are apt for computer vision and Advanced Natural Language Processing Problems beacause its ability to extract suitable features which in turn increase the accuracy and reduce the processing time in future perspective.

4.2 Deep Architectures

It is hard to decide on the most suitable feature vector for sufficiently successful recognition by handcrafted feature selection. As an alternative, raw image of handwritten samples may also be fed at the input layer of deep architecures and the network may be allowed to learn to discriminate samples of different classes. It has now been well established that CNNs are capable of learning efficient discriminative features needed for a classification task. The deep architectures are good in feature selection and classification. In the following sections we are discussing the two deep architectures that give better results.

4.2.1 Convolutional Neural Network

Convolutional Neural Networks are designed in several ways, the foremost architecture was LeNet designed by Yann Lecun [104] to recognize handwritten digits. Later several architectures comes like Alexnet [105], ZFNet [106] and GoogLeNet [107] for the classification of images.

The architecture of Convolutional Neural Network includes one or more convolutional layers precedes maxpooling layer. Maxpooling operation subsample the output of the previous layer by taking the maximum value in a sliding window(rectangular neighbourhood). The movement of the sliding window is determined by a hyper parameter called stride. If the stride value is 2 and sliding window size 2×2(maxpooling kernal size), the maxpooling layer convert the two dimensional array to half of the size, which is invariance to translation [108].

The input to the architecture can be one dimensional or two dimensional vectors. Generally in documents, it will be gray scale image with size $h \times w$ where h is the height and w is the width of the image. The number of filters used in the convolution layers can be n, of size $h_k \times w_k$ where $h_k < h$ and $w_k < w$. The kernels are convolved with the image and produce n feature maps of size $(h - (h_k - 1), w - (w_k - 1))$. To retain the size of the image, usually padded with $h_k - 1$ zero's in height wize and $w_k - 1$ zero's in width wise.

The last layer is normally a fully connected layer with softmax as the activation. Usually, a deep CNN architecture involves a large number of connection weights and thus its proper training requires a significantly high volume of training samples which often stands as a bottleneck of using similar learning strategies in handwriting recognition tasks. A solution to this problem is the use of transfer learning strategy [109], where a moderately trained CNN is used for feature extraction and the values computed at its last convolution or sub-sampling layer are fed as features to another classifier such as a support vector machine (SVM) which has the capacity of being sufficiently trained based on a relatively smaller number of samples. Recently, similar transfer learning strategy has been used [110] to recognize handwritten numerals of several Indian scripts such as Devanagari, Bangla, Telugu and Oriya.

A fully connected Multi Layer Perceptron (MLP) can successfully recognize handwritten samples when these are fed with efficient discriminative feature vectors. Such feature vectors are usually handcrafted ones and it is obviously hard to decide on the feature vector which should lead to sufficiently successful recognition. As an alternative, raw image of handwritten samples may also be fed at the input layer of an MLP and the network may be allowed to learn to discriminate samples of different classes. However, the number of connection weights of such a network must be huge, particularly when the number of classes of the underlying recognition problem is large. Thus, training of such a heavy network requires a large number of labelled samples which is difficult to arrange.

An alternative to the use of MLP for handwriting recognition disregarding selection of feature vector is the use of a convolutional neural network. It has now been well established that CNNs are capable of learning efficient discriminative features needed for a classification task. Since the lower part of a CNN architecture is not fully connected, these involve smaller number of connection weights than its MLP counterpart. Moreover, if a CNN is used only as a feature vector extractor leaving the classification task to another suitable tool such as the Support Vector Machine (SVM), then the CNN need not to be sufficiently trained and it helps to get the job done even with the availability of a limited number of training samples. Such a strategy is known as "Transfer Learning" [111]. On the other hand, it has now been established that the larger depth of a CNN architecture translates into its performance [112]. However, there is a limit of the depth of a CNN beyond which the network fails to be trained successfully due to the well known problem of vanishing / exploding gradients [113].

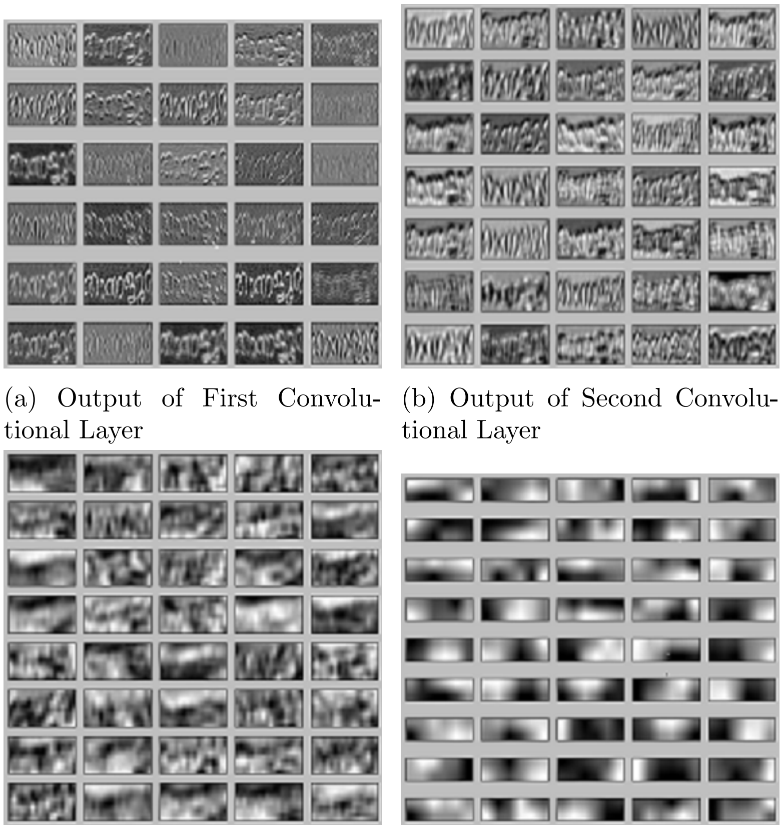
The different hyperparameters used and their values are shown in Table:4.1 Output of the first, second, third and fourth Convolutional Layers is shown in Figure 4.1. First convolution layer produces 30 feature maps and in some of them produces foregorund and back ground separation.Second layer convloution produces 35 feature maps and capture edges and structural features of the image. Third and fourth convolution layers capture more fine details.

Table 4.1: Hyper parameters

parameter	**value**
Depth	30,35,40,45
Stride	1
zero-padding	None
Batch Size	32

Proposed Approach

The work flow of the proposed recognition approach includes a brief preprocessing stage followed by feature extraction using a CNN and finally



(a) Output of First Convolutional Layer

(b) Output of Second Convolutional Layer

(c) Output of Third Convolutional Layer

(d) Output of Fourth Convolutional Layer

Figure 4.1: Output of Convolutional Layers

classification with the help of a SVM. Details of the approach are presented below. Further details of CNN and SVM can be found in [104] and [114] respectively.

Preprocessing

In traditional approaches of offline handwriting recognition, several preprocessing operations are performed on input samples to reduce the variations in their images. However, similar preprocessing modules do not have much role in CNN based recognition approaches because the design of a convolutional neural network architecture has the inherent capacity to handle various sources of variations in input samples. The only preprocessing operation required to be performed before feeding the samples to a CNN architecture is size normalization because the input layer of such a neural network has certain fixed size. In the present study, we experimented with several choices of the size of input layer and observed 64×100 input layer as the optimal size. Thus, in the present approach, the preprocessing stage involves (i) cropping the minimum bounding rectangle of the input word image and (ii) size normalization of the cropped image to the size 64×100 using bicubic interpolation[115].

CNN architecture

The convolutions operation, $S(i, j)$ of handwritten image, I and a kernal, K of size(m, n), can be defined as Equation (4.1) [108]

$$S(i,j) = (I * K)(i,j) = \sum_{p=0}^{m} \sum_{q=0}^{n} I(i+p, j+q) K(p,q) \tag{4.1}$$

The deep architecture of the CNN used in the present task of offline handwritten Malayalam word recognition consists of 10 layers: 4 convolution layers, 4 subsampling layers and a fully connected part which includes 1 hidden layer and an output layer. A diagram of this architecture is provided in Figure 4.2.

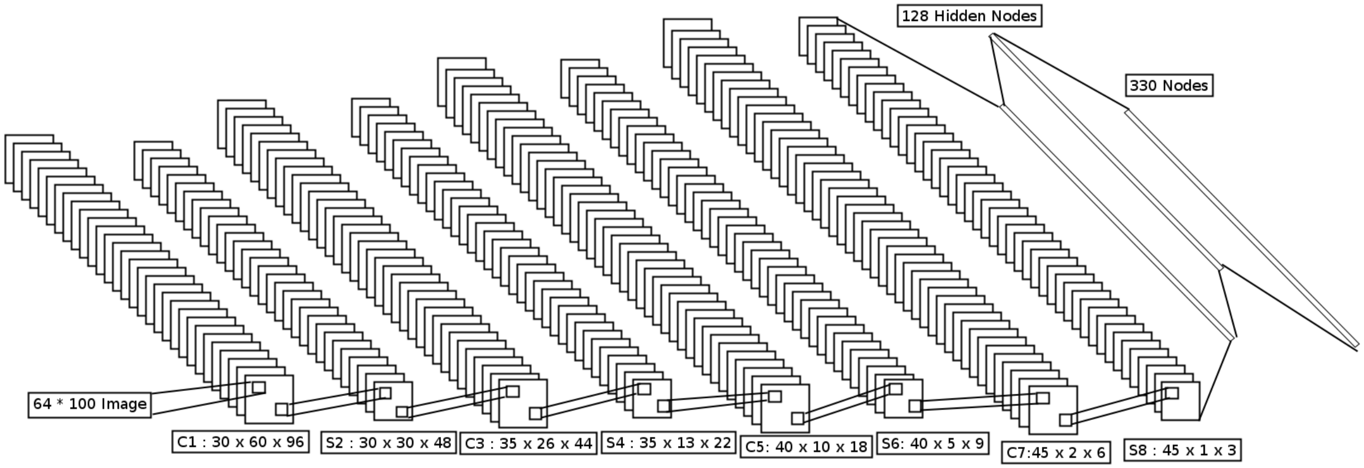


Figure 4.2: Proposed architecture of the convolutional neural network

Size of the input image to this network is 64×100 on which 5×5 kernel with stride 1 is used to generate 30 feature maps each of size 60×96 at the first convolutional layer C1. Maxpooling based on non-overlapping 2×2 kernel on these feature maps produces 30×48 feature maps at the subsampling layer S2. Next, we apply convolution operation using 5×5 kernel with stride 1 followed by maxpooling with non-overlapping 2×2 kernel to obtain 35 feature maps of size 26×44 at second convolutional layer C3 and the same number of feature maps of size 13×22 at the second subsampling layer S4 successively. Output of the third convolutional layer C5 consists of 40 feature maps of size 10×18 and these are obtained by using 4×5 kernel at stride 1. These are next reduced to the size 13×22 at the subsampling layer S6 with the help of maxpooling based on the same non-overlapping 2×2 kernel. Finally, 4×4 kernel at stride 1 is used to obtain 45 feature maps of size 2×6 at the convolution layer C7 and maxpooling as before is used to obtain 1×3 feature maps at the last subsampling layer S8. There are 135 values at S8 which are fed as input to the fully connected part of the network. This part has a hidden layer of 128 nodes and an output layer 330 nodes which is the number of underlying word classes.

We arrived at the above architecture of the CNN based on simulations of a large number of various architectures and the present architecture provided an optimal performance on the dataset described in Chapter 3. Activation used in all the convolutional layers and fully connected layer is ReLU (Rectified Linear Unit) [105]. ReLU is more suitable for deep convolutional neural networks because it is much faster than other activation units like tanh [105].

Activations used in the output of convolutional and first fully connected layer is ReLU. The equation 4.2 is a typical ReLU acivation function, where a is the input to the activation.

$$f(a) = max(0, a) \tag{4.2}$$

Final layer use softmax as the activation function[116] as shown in equation 4.3, where $f_i(y)$ is the output of i^{th} node in the final layer. It ensures the sum of all the outputs is 1 and the predicted value in between 0 and 1.

$$f_i(y) = \frac{e^z_i}{\sum\limits_{j=1}^{330} e^z_j} \tag{4.3}$$

Categorical cross entropy is used as the loss function viz. if t_j is the target value and y_j is the predicted value, it can be calculated defined in equation 4.4.

$$f(t_j, y_j) = -\sum_j t_j \log y_j \tag{4.4}$$

Training of the CNN

Intialization of the weights to neural network is important because if it is too small, then the signal shrinks as it passes through each layer and if the weights in a network starts too large then the signal grows. Weights of the CNN were initialized with glorot or xavier uniform initializer [117]. Gradient descent is the most popularly used method for training of neural networks. In the literature, there exists different algorithms for optimization of gradient descent training of neural networks. These include (i) batch gradient descent which updates the connection weights after each epoch, i.e., after presentation of the whole set of training samples to the network, (ii) mini-batch gradient descent which update the network weights after each presentation of a batch of n training samples and (iii) stochastic gradient descent, popularly known as SGD [118], which randomly selects a training sample at each iteration for presentation to the network and each time the weights are updated. In the present implementation, we used mini-batch gradient descent variant with batch size equals to 32.

Our experiments show that adadelta [119], one of the gradient descent optimization algorithms giving better performance or learning compared to SGD (Stochastic Gradient Descent). In adadelta it is taking only the running average of gradient [119]. Let us consider the gradient as g_t, Accumulate Gradient using equation 4.5

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1-\gamma)g_t^2 \tag{4.5}$$

Compute update using equation 4.6

$$\Delta\alpha_t = -\frac{RMS[\Delta\alpha]_{t-1}}{RMS[g]_t} g_t \tag{4.6}$$

Where $RMS[g]_t] = \sqrt{E[g]_t^2 + \epsilon}$

calculate Accumulate update through equation 4.7

$$E[\Delta\alpha_t^2] = \gamma E[\Delta\alpha^2]_{t-1} + (1-\gamma)\Delta\alpha_t^2 \tag{4.7}$$

Apply update using equation 4.8

$$\alpha_{t+1} = \alpha_t + \Delta\alpha_t \tag{4.8}$$

In our experiment, we use $\gamma = 0.95$ and $\epsilon = 1e - 08$, since mini-batch gradient descent algorithm cannot guarantee convergence of the training at a good local minimum, different additional strategies are adopted by the practitioners for the required effective training of the network.

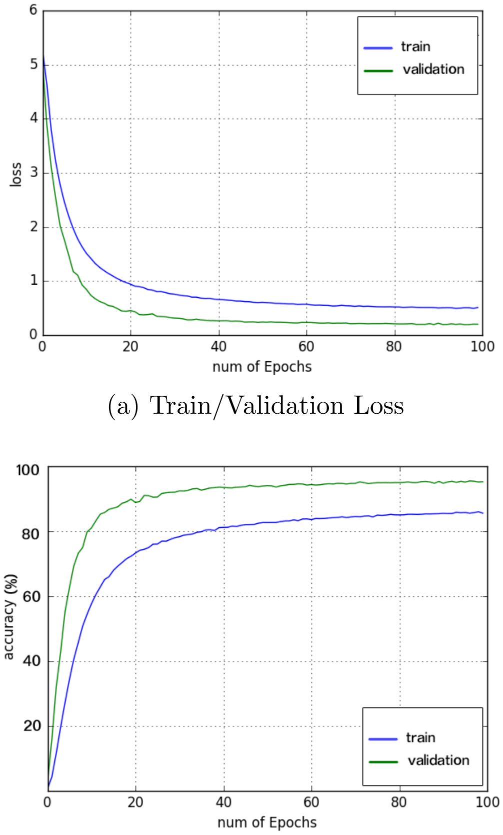
A deep convolutional neural network involves a large number of parameters. Thus, the problem of overfitting occurs in situations when there are only a limited number of training samples. Although the deep learning strategy had shown its promising performance in various applications, the two major issues of this strategy are (i) overfitting and (ii) computational burden of its training algorithm. Effect of computational burden has, in the meantime, becomes manageable due to the availability of high speed GPUs. On the other hand, various regularization techniques have been experimented in the literature to avoid the overfitting problem. Dropout is a comparatively new regularization technique that has been more recently employed in deep learning [120] to get rid of this problem. The term ‘dropout’ refers to dropping out units in a neural network during training. By dropping a unit out, we mean temporarily ignoring it from the learning task along with all its incoming and outgoing connections. Units of the network for dropout are chosen randomly. In the present implementation, each unit of the network is dropped out with the probability 0.2.

Data augmentation [120] is another strategy which is used to prevent overfitting of deep neural network architecture. Here, we randomly apply one of the transformations from (i) rotation (-5° to +5°) and (ii) Gaussian noise with variance 0.2, before feeding a training sample to the network.

Stopping of training epochs of a neural network is another crucial issue for its successful application. We used a validation set of samples for this purpose. In fact, 20% of the training samples of each class is selected randomly to form the validation set and the remaining training samples were actually used for adjustment of connection weights. Initially, the network error on validation set remains high and this gets decreased as the training progresses but after a certain number of iterations, the validation error starts increasing and this is the instant when we stopped training and obtained the network performance on the test set.The training/ validattion loss/ accuracy is graphically plotted in Figure 4.3.

Support Vector Machine

In the literature, support vector machines (SVM) have been established as an efficient classification model even in the presence of a limited number of training samples. It maps input samples into a higher dimensional space where an optimal separating hyperplane is constructed. Since its computation involves solving a quadratic programming problem, SVM does not have the difficulty of the existence of multiple local minima unlike the gradient descent based learning method of CNN architecture.



(a) Train/Validation Loss

(b) Train/Validation Accuracy

Figure 4.3: Error-Accuracy Curves

Traditionally, SVM is a two-class classifier. It solves multiclass classification problem using one of the two strategies: "one-versus-all" or "one-versus-one". In the present task, we used the latter strategy. In this strategy, $n(n-1)/2$ classifiers are constructed for an n-class classification problem and each one is trained by using samples from the two classes. Finally, results of all these 2-class classifiers are combined to reach at the decision.

SVM uses the well known kernel trick. A kernel provides the similarity of two inputs to it as the output. Usually, an implementation of SVM provides various options for the kernel function such as (i) linear, (ii) polynomial, (iii) RBF etc. In the present task, we used RBF kernel [121]. This kernel involves two parameters which are often denoted by C and γ. We obtained the suitable values of these two parameters by a grid search strategy [122].

An SVM as described above has been trained using the 135 feature values computed at the last sub-sampling layer S8 of our CNN architecture. Since the training of SVM does not need any validation sample set, we have used the entire set of 19,800 training samples for training of the SVM. In the recognition phase, feature vector of an unknown (test) sample is first generated by the CNN at S8 layer of the CNN which is next fed to the SVM to get its classification output.

Evaluation Metric

Let TP is the number of True Positives, TN is the number True Negatives, FP is the number of False Positives and FN is the number of False Negative. Accuracy is calculated using the Equation (4.9)

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{4.9}$$

Result-CNN

We have simulated the proposed holistic approach of handwritten word recognition on (i) our database of 330 class handwritten Malayalam words, (ii) two databases of 114 class handwritten Marathi legal amount words and (iii) another database of 106 class Hindi legal amount words. Results with Malayalam Datasets are explained in Table 4.2.

Here, we obtained recognition accuracy percentage of the proposed approach on the test set of 11,220 handwritten samples corresponding to 330

Table 4.2: Result Analysis-CNN with JMHRDB1

Dataset (D)	Accuracy (CNN)(%)	SVM(%)	Dataset (D)	Accuracy (CNN)(%)	SVM(%)
D1(1-100)	96	97.1	$D_1 \cup D_2$	95.7	96.9
D2(101-200)	96.6	97.9	$D_1 \cup D_3$	96	97
D3(201-330)	95.4	96	$D_2 \cup D_3$	96.1	97
			$D_1 \cup D_2 \cup D_3$	**95.74**	**96.90**

classes of Malayalam words. The hybrid architecture consisting of CNN and SVM provided 96.90 % recogni- tion accuracy on this test set while the CNN alone provided 95.74 % accuracy.

The result analysis with different values of depth is shown in 4.3

Table 4.3: Results with different hyper parameter-depth combinations

Depth	$D_1 \cup D_2 \cup D_3$
30,35,40,45	96.90
25,30,35,40	95.4
15,20,25	89

An analysis of misclassified words shows that when a word is misclassified as another word in the lexicon, these two words have necessarily a common character string. Examples of a few such word pairs are shown in Table 4.4.

The model implemented for the Malayalam Word recognition is also simulated on a database of handwritten Hindi words used in [45]. The lexicon size of this database is 106 and it consists of 8480 samples provided by 80 writers. It contains all the possible words that one needs to use to write a valid amount in Hindi language. The training set consists of 6360 word samples written by 60 writers and the remaining 2120 samples written by 20 writers form the test set. A hybrid architecture similar to the one used for Malayalam word database is trained for the present 106 class word recognition problem and the recognition performance of the same is verified

Table 4.4: Misclassfication scenarios of Malayalam words

Actual word		Recognized Word	
Malayalam	Transliteration	Malayalam	Transliteration
ആലപ്പാട്	Alappad	ആലപ്പുഴ	Alappuzha
അരിക്കുളം	Arikkulam	കരുംകുളം	Karumkulam
ചെങ്ങന്നൂർ	Chengannur	പന്ന്യന്നൂർ	Pannyannur
ചിറ്റൂർ	Chittoor	ഒറ്റൂർ	Ottoor
ഇടമുളയ്ക്കൽ	Idamulakkal	പുഴയ്ക്കൽ	Puzhakkal
കൊടംതുരുത്ത്	Kodamthuruth	കടുതുരുത്തി	kaduthuruthi
കൊടംതുരുത്ത്	Kodamthuruth	മുളന്തുരുത്തി	Mulamthuruthi
ഝകാരം	Dhakaram	ഝങ്കാരം	Dhamkaram
ഋണം	Wranam	ക്ഷണം	Kshanam

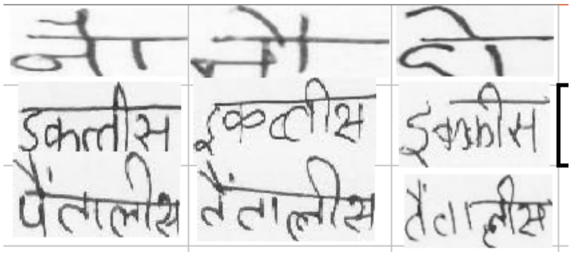


Figure 4.4: Samples from Hindi Dataset

on its test samples. We obtained 94 % accuracy on the test set of Hindi valid amount word database which improves the existing state-of-the-art accuracy value of 83.07% published in [45]. Results with Hindi datasets are explained in Table 4.5.

Samples from the Hindi Dataset is shown in Figure 4.4.

The hybrid deep neural network model presented in this thesis has been simulated on two handwritten word databases DB1 and DB2 [45] of Marathi valid amount word used in bank cheques. The lexicon size of both of these two Marathi word Databases DB1 and DB2 is 114. Samples of DB1 were written by 90 writers while the same of DB2 were written by another 70

Table 4.5: Result Analysis-CNN-Hindi

Author	Correctness (%)	Error (%)	Rejection (%)	Reliability (%)
Malakar[2017]	90.78	9.22	0	90.78
Jaydev[2011]	83.07	16.70	0.22	83.25
Proposed	94.0	6.0	0	94

writers. Compared to DB2, samples of DB1 are neat and legible. Word databases DB1 and DB2 consist of 10260 and 7980 image samples respectively. The proposed approach provided 93% and 92% recognition accuracies on DB1 and DB2 respectively, which improved the existing respective state-of-the-art accuracy values of 85.78% and 78.79% [45]. Results with Marathi datasets are shown in Table: 4.6

Samples from Marathi dataset is shown in Figure: 4.5

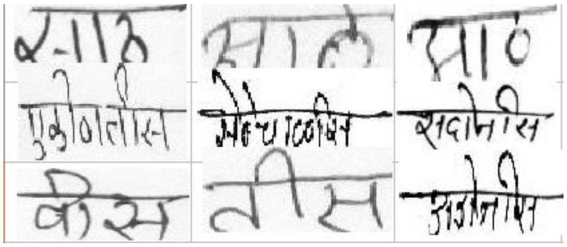


Figure 4.5: Samples from Marathi Dataset

Table 4.6: Result Analysis-CNN-Marathi

Author	Database	Correctness (%)	Error (%)	Rejection (%)	Reliability (%)
Jaydev	DB1	85.78	10.48	3.72	89.10
	DB2	78.79	18.62	2.57	80.88
Proposed	**DB1**	**93**	**7.0**	**0.0**	**93.0**
	DB2	**92**	**8.0**	**0.0**	**92**

4.2.2 Residual Network

The Deep Architectures suffer a problem with vanishing gradient or explosion. Also in deep architecture means more layers and obviously more parameters, some times it may not help in training and produce errors. Residual Network(Resnet) gives a solution with residual block. In the original implementation of resnet it successfully implemented 18 to 152 layers[123][124]. Residual mapping accomplished through shortcut connections. These shortcut connections helps in backpropagation of gradients and ensure faster training. Sample residual block is shown in Figure 4.6.

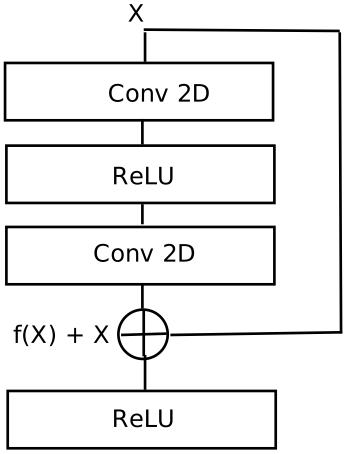


Figure 4.6: Residual Block

Consider the iput of the network is x and output is $R(x)$. The residual function $F(x) = R(x) - x$. So the actual output is $R(x) = F(x) + x$. Resnet introduces identity connection between layers.

Resnet Architecture

Input image, I with size 64 × 100 is fed to the first convolution layer C_1 on which 7 × 7 kernel with stride 2 is used to generate 64 feature maps with size 32×50. All the convolution layers are followed by Batch Normalization (BN) [125]. In BN, normalize each mini batch by both mean and variance. Various style in handwriting inputs of the same class may vary or the distribution of the features may vary. It is known as covariate shift. BN converts unnormalized values to normalized values and also enable us to train very deep networks.

$$Mean, \mu = \frac{1}{m}\sum_{i=1}^{m} X_i$$

$$Variance, \sigma^2 = \frac{1}{m}\sum_{i}(X_i - \mu)^2$$

$X_{inorm} = \frac{X_i - \mu}{\sqrt{\sigma^2 + \epsilon}}$, Where ϵ is included for numerical stablity and defined in our experiments us 0.001
$X_{norm} = \gamma X_{inorm} + \beta$, where γ and β are learnable parameters of model

The advantage of batch normalization are

1. Generalization
2. Network can trained with higher learning rate
3. Improved results
4. Initialization of the weights much easier

Maxpooling operation with 3×3 kernel on these feature maps produce 16×25 feature maps at subsampling layer $S2$. Layer 3 and 4 together known as residual block-1 consists of two convolutional layers with 3×3 kernel and 64 feature maps. Layers 5 and 6 together is known as residual block-2 and consists of two convolutional layers with the same configuration as residual block-1. Next Convolutional Layer $C6_1$ with 1×1 convolutions with stride 2 is used to match the size for add next residual blocks and it produce feature maps with size of 8×13. Layer 7 and 8 together is known as residual block-3 and consists of two convolutional layers with 3×3 kernel and 128 feature maps. Layer 9 and 10 together is known as residual block-4 and it follows the same configuration of residual block-3. Next Layer $C10_1$ is used for size matching purpose,with a convolution of 1×1 and with a stride of 2 produce a feature map of size 4×7. Layers 11 and 12 together is known as residual block-5 and consists of two convolutional layers with 3×3 kernel size and 256 feature maps. Layer 13 is used for average pooling and in the final layer softmax is used as the activation function to predict the output. Architecture of the proposed method with residual blocks are shown in Figure 4.7.

The detailed schematic daigram of residual block is shown in Figure: 4.8.

Training of Resnet

The architecture, the parameters of network and proper training will give a optimal generalization performance. The weights of CNN layers are initialized using He_normal method [126]. For the optimization of the model Stochastic Gradient Descent is used. Batch size selected is 32 and for improving optimization, batch normalization is used after each convolution layer. Batch normalization [125] provide regularizing effect and thus dropout is avoided in the implementation. Validation accuracy determines the quality of generalization, so the 20% of the training samples(T) are used for validation. 15840 samples are used for training, 3960 samples are used for validation and 11220 samples are used for testing(T'). The Train/ Test/ Validation split is matching with the section 4.2.1. During training if the validation accuracy doesn't improves the learing rate is reduced. We use a factor of 0.3162 and a patience of 5 epochs to wait and if there no improvement, learning rate is reduced to $learningrate \times 0.3162$. Minimum learning rate is fixed to $0.5e^{-6}$. To avoid overfitting the early stopping method is implemented and if there is no improvement of accuracy by 0.001 for the last 10 epochs, the training will stop. This network took 55 epochs for

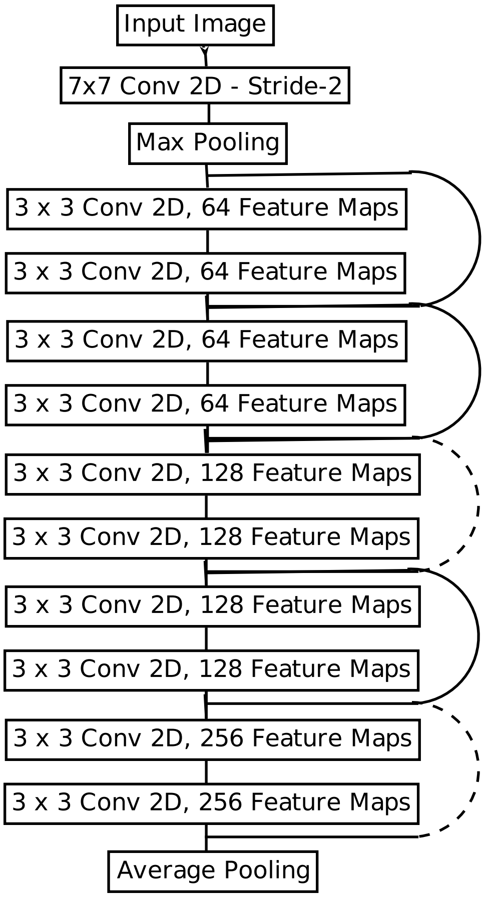

Input Image
7x7 Conv 2D - Stride-2
Max Pooling
3 x 3 Conv 2D, 64 Feature Maps
3 x 3 Conv 2D, 64 Feature Maps
3 x 3 Conv 2D, 64 Feature Maps
3 x 3 Conv 2D, 64 Feature Maps
3 x 3 Conv 2D, 128 Feature Maps
3 x 3 Conv 2D, 128 Feature Maps
3 x 3 Conv 2D, 128 Feature Maps
3 x 3 Conv 2D, 128 Feature Maps
3 x 3 Conv 2D, 256 Feature Maps
3 x 3 Conv 2D, 256 Feature Maps
Average Pooling

Figure 4.7: Resnet Architecture

convergence. Error Curves and Accuracy curves of training and validation are shown in Figure 4.9.

Result Analysis-Resnet

Experiments are done in Malayalam dataset. The split of Training/ Validation/ Testing Data follows Section: 4.2.1. The result analysis of resnet with Malayalam dataset is shown in Table 4.7.

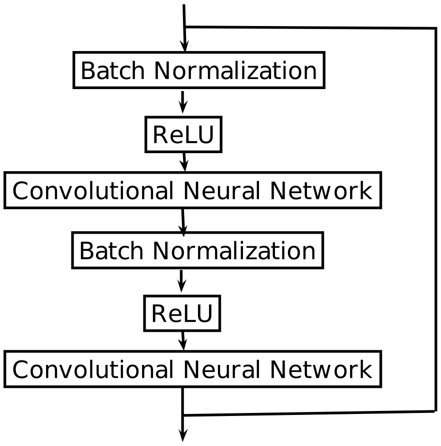


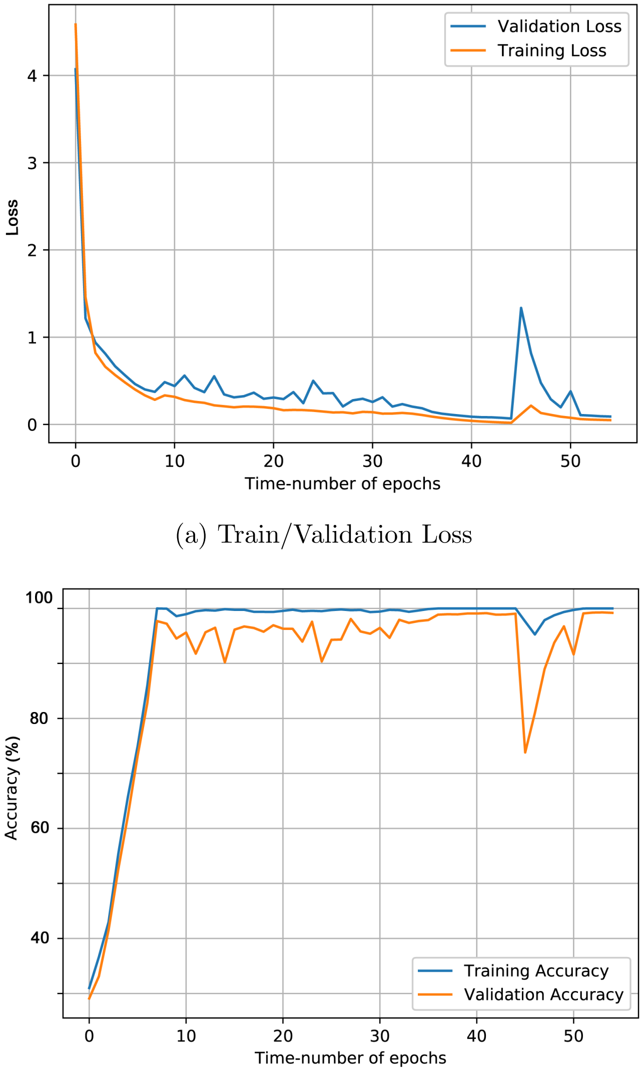
Figure 4.8: Residual Block Expanded

Table 4.7: ResnetResult

Dataset (D)	Accuracy	Dataset(D)	Accuracy
D1(1-100)	97.32	$D_1 \cup D_2$	97.4
D2(101-200)	97.7	$D_1 \cup D_3$	96.6
D3(201-330)	96.1	$D_2 \cup D_3$	96.3
		$D_1 \cup D_2 \cup D_3$	97.53

4.2.3 Two Stage Classification

The analysis of the results shows that one of the factor determines the accuracy is the samples with similar shapes. The confusion matrix obtained after the first stage using the architecture discussed in Section 4.2.2 for find the mutually mis classified samples. In the second stage the features are computed from the group of classes using the convolutional neural network discussed in the section 4.2.1 and the classifier used is SVM. In a nutshell groups can be identified using stage I and classes within the group can be identified in stage II. In the first stage explained in section 4.2.2, we found 277 samples are misclassified out of 11220 samples. After the analysis of



(a) Train/Validation Loss

(b) Train/Validation Accuracy

Figure 4.9: Error-Accuracy Curves-Resnet

confusion matrix of first stage, we identified 28 partly or fully similar word groups of 110 samples. So the 39.71 % of misclassified samples belongs to the set of groups. Groups with class ID is shown in table: 4.8. The group formation and merging of the group is explained in algorithm 1. It consists of two levels, First level group formed with mutually misclassified samples. For example in first level “{ചിറ്റാർ/Chittar,ചിറ്റൂർ/Chittoor}”, and “{ചിറ്റൂർ/Chittoor},ഒറ്റൂർ/Ottoor” are two distinct groups. In the seconds level it merged to a single group “ചിറ്റാർ/Chittar,ചിറ്റൂർ/Chittoor,ഒറ്റൂർ/Ottoor”.

Algorithm 1 Algorithm for Group Formation

```
procedure GROUP FORMATION
    Input=$CM$                                         ▷ Confusion Matrix
    $G = \emptyset$, $k = 0$
    for $i \leq j$ do
        if $CM_{i,j} \neq 0$ & $CM_{j,i} \neq 0$ then
            if $CM_{i,j} + CM_{j,i} \geq threshold$ then            ▷ threshold=2
                $G_k$ =Merge($i, j$)                ▷ Merge $i^{th}$ and $j^{th}$ classes
                $G = G \cup \{G_k\}$
                $k = k + 1$
    while all sets are not mutually exclusive in $G$ do
        for $\forall G_i, G_j \in G$ do
            if $G_i \cap G_j \neq \emptyset$ then
                $G_i = G_i \cup G_j$
    return $\{G\}$
```

Two-Stage Classification gives an improvement of 0.55 % accuracy in over all result as shown in Table:4.9.

Table 4.8: Groups With Class ID

S. No {class ID's}	Word Pairs	No. Of Misclassified Samples
1 (1,20)	അടൂർ/Adoor,അഴുത /Azhutha	4
2(15,78)	ആറ്റിങ്ങൽ/Aattingal,കോട്ടാങ്ങൽ/Kottangal	4
3(35,36,130)	ചിറ്റാർ/Chittar,ചിറ്റൂർ/Chittoor,ഒറ്റൂർ/Ottoor	10
4 (43,158)	എടപ്പറ്റ/Edappatta,പുൽപ്പറ്റ/Pulppatta	2
5(55,112,71)	കടുത്തുരുത്തി/Kaduthuruthi,മുളന്തുരുത്തി /Mulanthuruthy,കൊടംതുരുത്ത് /Kodamthuruthu	8
6 (64,10)	കരുംകുളം /Karumkulam,അരിക്കുളം/Arikkulam	4
7(74,2)	കോങ്ങാട്/Kongad,ആലങ്ങാട്/Alangad	2
8(101,200)	മരിയാപുരം/Mariyapuram,വിജയപുരം/Vijayapuram	2
9(109,107)	മുന്നിയൂർ/Munniyoor,മേപ്പയൂർ/Meppayoor	2
10(110,192)	മൊറയൂർ/Morayoor,വാഴയൂർ/Vazhayoor	2
11(134,135)	പള്ളിച്ചൽ/Pallichal,പള്ളിക്കര/Pallikkara	2
12(138,140)	പനങ്ങാട്/Panangad,പാങ്ങോട്/Pangod	2
13(162,157)	പുഴയ്ക്കൽ/Puzhaykkal,പുളിക്കൽ/Pulikkal	2
14(272,273)	ഈണം/Eenam,ഈന്ത/Eentha	4
15(278,277)	ഊൺ/Uun,ഊഢ/Uuda	4
16(183,180)	തൃപ്രങ്ങോട്ടൂർ/Thriprangottoor, തൃക്കലങ്ങോട്/Thrikkalangodu	2
17(213,214)	കൊടുവായൂർ /Koduvayoor,കൊടുങ്ങല്ലൂർ/Kodungalloor	2
18(241,242, 270))	ഉദയഗിരി /Udayagiri,ഉദയപുരം/Udayapuram, ഉദയസന്ധ്യ/udayasandhya	8
19(244,245)	ഝകാരം/Dhakaram,ഝങ്കാരം/Dhankaram	6
20(248,249)	ഘടകം/Ghadakam,ഘടന/Ghadana	2
21(265,266)	ഫലം /Phalam,ഫലകം/Phalakam	2
22(279,286)	ഋണം/Wranam,ക്ഷണം/Kshanam	4
23(283,284)	ഔദാര്യം/Audaryam,ഔജസ്യം/Aujasyam	4
24(293,294)	ദുർഗ്ഗുണം /Durgunam,ദുർഗ്ഗന്ധം/Durgandham	4
25(301,302)	ശോഭന/sobhana,ശോഭനം/sobhanam	4
26(306,307)	സംഹാരം/Samharam,സംഹിത/Samhitha	4
27(309,310)	സുഗന്ധം/Sugandham,സുഗന്ധി/sugandhi	4
28(330,315, 329,328)	വാങ്മൂലം/Vangmoolam,വാങ്മയം/vangmayam, വാങ്മുഖം/vangmugham,വാങ്മയ/vangmaya	10
	Total Samples	**110**

Table 4.9: Improvement of Two Stage Classification

Dataset (D)	Accuracy	Percentage(%) of Improvement
$D1 \cup D2 \cup D3$	98.08	**0.55**

4.3 Summary

This chapter explores a deep hybrid neural network architecture for offline handwritten Malayalam word recognition. The same architecture has also been experimented on existing handwritten word datasets of Hindi and Marathi legal amounts. The later experiment results show that the proposed approach improves the result on these two datasets. Implementation of Resnet also provides good results and two-stage classification also tried and achieved good results.

Chapter 5

Performance Analysis: Lexicon Specific

"It amazes me sometimes that even intelligent people will analyze a situation or make a judgement after only recognizing the standard or traditional structure of a piece."

David Bowie

5.1 Introduction

This chapter compares the performance of traditional methods with deep learning based methods. Feature Extraction and classification are integral part for any pattern recognition tasks. Feature Extraction methods can be classified to either handcrafted or Deep Learning based [127]. In handcrafted feature extraction methods human expert decides the relevant features that required to classify the pattern effectively and efficiently. Architecture like CNN is used for the extraction of relevant features from the raw data.

Generally template based and feature based methods are used for pattern recognition tasks[128]. Handcrafted features was popular in earlier days but now handcrafted or machine extracted features along with machine learning techniques are used for the recognition. Spatial domain and transform domain are the two approaches where in the former case it extracts features directly from the image and in later case it transforms image to another representation, features are extracted from these representations. Most commonly used features in spatial domain methods are topological, statistical, directional and curvature[129]. According to No Free Lunch Theorem [130] states the importance of separate machine learning models for different types of datasets.

5.2 Handcrafted Feature Extraction Methods

Extraction of the relevant features is always a cumbersome procedure to remove needless variabilty from the handwritten word images. In computer vision one can find a lot of features suitable for image recognition, the selection of exact and efficient features is a major bottleneck. In this section we explore some of the state-of-the-art features for the script independent recognition of handwritten images.

5.2.1 Histogram of Oriented Gradient(HOG)

HOG Feature Descriptor calculates the histograms of directions of oriented gradients [131]. It is highly successful to find out the object shape because magnitude of gradients is high around edges and corners. It is the reason why we select this for our holistic recognition of words.

Steps for HOG Feature Extraction is shown in Algorithm: 2 [132]

First step towards the calculation of HOG descriptor is to find out the vertical and horizontal gradient. The Image is filtered with the kernels to find out the gradients g_x and g_y.

Algorithm 2 Algorithm for HOG Feature Extraction for nbin = 5,6,7 and blockstride of size 8, 16

```
procedure HOG Feature Extraction
    Resize image to $64 \times 128$
    Perform Thinning operation
    Compute Gradient
    for each nbin and blockstride do
        Compute the Histogram of Gradient of $8 \times 8$ cell
    return Feature Vector
```

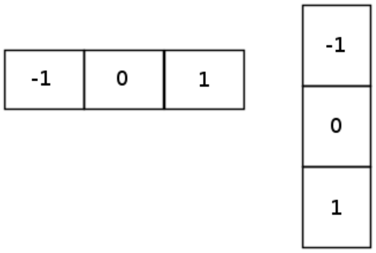


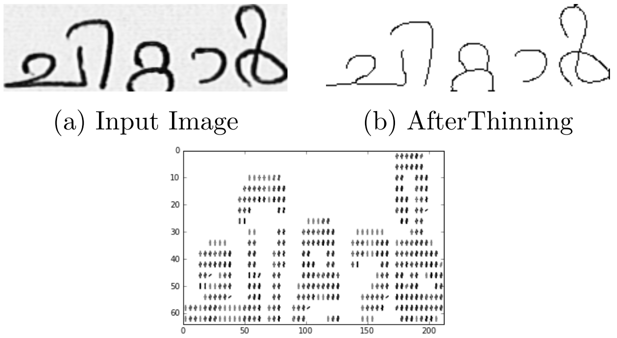
Figure 5.1: kernel to find horizontal and vertical gradient

Magnitude and direction of the gradient can be calculated using the Equation (5.1) and Equation (5.2)

$$g = \sqrt{{g_x}^2 + {g_y}^2} \tag{5.1}$$

$$\theta = \arctan \frac{g_y}{g_x} \tag{5.2}$$

As we mentioned in the Algorithm: 2, image is converted to 64 rows and 128 columns. Suppose Block size, block stride, cell size, number of bins are (16,16), (8,8), (8,8), 7 respectively. With this parameters we will get 2940 features. Thus we require a proper dimensionality reduction technique.Original Image, Thinned Image and its HOG visualization is shown in Figure 5.2a,Figure 5.2b and Figure 5.2c.



(a) Input Image (b) AfterThinning

(c) Visulaisation of HOG

Figure 5.2: HOG Visualization with Thinned Image

5.2.2 Pyramid Histogram of Oriented Gradient(PHOG)

PHOG feature descriptor uses to find the spatial distribution of shape [133] and it consists of two steps 1) Find the Image Pyramids 2) Calculate HOG as explained in Section 5.2.1 [134]. Image pyramids are stack of images with mulitple resolutions in different frequency band. The range of frequency band will be from low to high viz. smooth detail to finest detail. Sample image pyramid is shown in Figure 5.3 and algorithm is explained in Algorithm 3

5.2.3 Wavelet Based

Wavelet transform is an effective tool to represent images at different levels of resolution. It extracts temporal and spacial information from the image. Mother wavelet, prototype from all other types of wavelets are scaled or shifted

Wavelet families include Haar, Daubechies, Symlets, Coiflets, Biorthogonal, Reverse biorthogonal, Meyer, Gaussian, Mexican hat, Morlet, Complex Gaussian, Shannon, Frequency B-Spline and Complex Morlet. Orthogonal and biorthoganal are the two categories of wavelet family. The

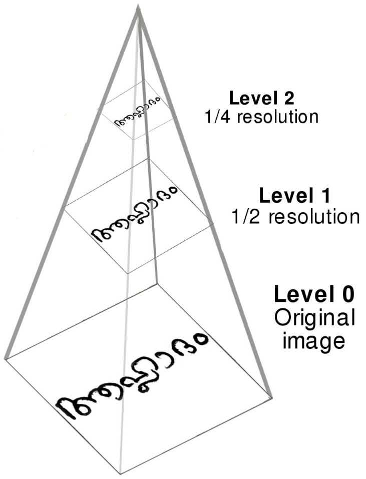


Figure 5.3: Pyramid Representation of Handwritten Word Image of "ആഹ്ലാദം"

Algorithm 3 Algorithm for PHOG Feature Extraction for nbin = 5,6,7, blockstride of size 8, 16 and pyramid level, n=1,2

```
procedure PHOG Feature Extraction
    Resize image to $64 \times 128$
    Apply Thinning operation
    Represent in $n$ pyramidal form
    for each $n$ do
        for each image in the pyramid stack do
            Compute Gradient
            for each nbin and blockstride do
                Compute the Histogram of Gradient of $8 \times 8$ cell
    Concatenate HOG Feature Vectors to form the n-level PHOG feature vector
    return PHOG feature vector.
```

coefficients of orthoganal filters are real numbers and in case of biorthoganal filters it is real numbers or integers. Selection of a proper mother wavelet is required according to the application to achieve the proper result. In our experiments, we consider Haar and Daubechies wavelet.

Haar Wavelet

Haar scaling function

$$\phi(x) = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

$$\psi(x) = \begin{cases} 1 & 0 \leq x < \frac{1}{2} \\ -1 & \frac{1}{2} \leq x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.4}$$

The scaling function $\phi(x)$ and the wavelet funtion $\psi(x)$ associated with the scaling filter h_ϕ and the wavelet filter h_ψ are:

$$\phi(x) = \sum_n h_\phi(n)\sqrt{2}\phi(2x - n) \tag{5.5}$$

$$\psi(x) = \sum_n h_\psi(n)\sqrt{2}\phi(2x - n) \tag{5.6}$$

According to [135] the sequences of vector spaces $(V_{2^j})_{j\in\mathbb{Z}}$ form a multi resolution approximation of $L^2(R^2)$ if and only if $(V_{2^j})_{j\in\mathbb{Z}}$ is a multi resolution approximation of $L^2(R)$. One can then easily show that the scaling function $\phi(x, y)$ can be written as

$$\phi(x, y) = \phi(x)\phi(y)$$

where $\phi(x)$ is the one dimensional scaling function of the multiresolution approximation $(V_{2^j})_{j\in\mathbb{Z}}$. Relevance is given to the horizontal and vertical directions in the image with a separable multi resolution approximation. This emphasis is apt for many types of images, such as handwritten documents. Let $u_n = x - 2^{-j}n$ and $v_m = y - 2^{-j}m$, the orthoganal basis of V_{2^j}

is then given by, for all $(n, m) \in \mathbb{Z}^2$,

$$\left(2^{-j}\phi_{2^j}(u_n, v_m)\right)_{n,m} = \left(2^{-j}\phi_{2^j}(u_n)\phi_{2^j}(v_m)\right)_{n,m} \tag{5.7}$$

The approximation of a signal $f(x, y)$ at a resolution 2^j is therefore characterized by the set of inner products

$$A^d_{2^j} = \langle f(x, y), \phi_{2^j}(u_n)\phi_{2^j}(v_m)\rangle_{(n,m)} \tag{5.8}$$

Let $(V_{2^j})_{j\in\mathbb{Z}}$be a separable multi resolution aproximation of $L^2(\mathbb{R}^2)$, Let $\phi(x)\phi(y)$ be the associated two dimensional scaling function. Let $\psi(x)$ be the one dimensional wavelet associated with the scaling function $\phi(x)$, then the three "wavelets"

$$\psi^1(x, y) = \phi(x)\psi(y), \psi^2(x, y) = \psi(x)\phi(y), \psi^3(x, y) = \psi(x)\psi(y) \tag{5.9}$$

are such that

$$\left(2^{-j}\psi^1_{2^j}(u_n, v_m),\ 2^{-j}\psi^2_{2^j}(u_n, v_m),\ 2^{-j}\psi^3_{2^j}(u_n, v_m)\right)_{(n,m)}$$

is an orthonormal basis of O^j_2 and

$$\left(2^{-j}\psi^1_{2^j}(u_n, v_m), 2^{-j}\psi^2_{2^j}(u_n, v_m), 2^{-j}\psi^3_{2^j}(u_n, v_m)\right)_{(n,m)}$$

is an orthonormal basis of $L^2(\mathbb{R}^2)$

As a conclusion we can define the the decomposition of image $A^d_{2^{j+1}}f$ into $A^d_{2^j}f$ and $D^k_{2^j}f$, where $k \in \{1, 2, 3\}$

$A^d_{2^j}f$ reperesents low horizontal and vertical frequencies and defined as

$$A^d_{2^j}f = ((f(x, y) * \phi_{2j}(-y))(2^{-j}n, 2^{-j}m))_{(n,m)} \tag{5.10}$$

$D^1_{2^j}f$ represents vertical high frequencies and horizontal low frequencies as defined as

$$D^1_{2^j} f = ((f(x, y) * \phi^j(-x)\psi^j(-y))(2^{-j}n, 2^{-j}m))_{(n,m)} \tag{5.11}$$

$D^2_{2^j} f$ represents vertical low frequencies and horizontal high frequencies as defined as:

$$D^2_{2^j} f = ((f(x, y) * \psi_{2^j}(-x)\phi_{2^j}(-y))(2^{-j}n, 2^{-j}m))_{(n,m)} \tag{5.12}$$

$D^3_{2^j} f$ represents vertical high frequencies and horizontal high frequencies as defined as:

$$D^3_{2^j} f = ((f(x, y) * \psi_{2^j}(-x)\psi_{2^j}(-y))(2^{-j}n, 2^{-j}m))_{(n,m)} \tag{5.13}$$

Daubechies Wavelets

According to [136] Daubechies wavelet is a function

$$\psi = {}_N\psi \in L^2(\mathbb{R})$$

, where $N \in \mathbb{N}$ defined by

$$\psi(x) := \sqrt{2} \sum_{k=0}^{2N-1} (-1)^k h_{2N-1-k} \phi(2x - k) \tag{5.14}$$

where $h_0, h_1, h_{2N-1} \in \mathbb{R}$ are constant filter co-efficients satisfying the conditions

$$\sum_{k=0}^{N-1} h_{2k} = \frac{1}{\sqrt{2}} = \sum_{k=0}^{N-1} h_{2k+1}$$

as well as, for $l = 0, 1, 2, ...N - 1$

$$\sum_{k=2l}^{2N-1+2l} h_k h_{k-2l} = \begin{cases} 1 & \text{if l } = 0 \\ 0 & \text{if l } \neq 0 \end{cases}$$

Scaling Filter coefficients for decomposition of image is shown in Table 5.1

Table 5.1: Scaling Filter Coefficients

DB4	DB8	DB12	DB16
0.48296291314453	0.23037781330890	0.11154074335011	0.05441584224310
0.83651630373781	0.71484657055292	0.49462389039845	0.31287159091430
0.22414386804201	0.63088076792986	0.75113390802110	0.67563073629729
-0.12940952255126	-0.02798376941686	0.31525035170920	0.58535468365421
	-0.18703481171909	-0.22626469396544	-0.01582910525635
	0.03084138183556	-0.12976686756726	-0.28401554296155
	0.03288301166689	0.09750160558732	0.00047248457391
	-0.01059740178507	0.02752286553031	0.12874742662048
		-0.03158203931749	-0.01736930100181
		0.00055384220116	-0.04408825393079
		0.00477725751095	0.01398102791740
		-0.00107730108531	0.00874609404741
			-0.00487035299345
			-0.00039174037338
			0.00067544940645
			-0.00011747678412

Wavelet Filter Coefficients for the decomposition of image is shown in Table 5.2 Wavelet co-efficients are used as the features. Algorithm for wavelet based feature extraction is given in Algorithm 4.

Algorithm 4 Algorithm for Wavelet based Feature Extraction for type=haar and daubechies, level of decomposition, $l = 1, 2$

```
procedure WAVELET BASED FEATURE EXTRACTION
    Resize image to $64 \times 128$
    for each level, $l$ do
        Apply Haar Wavelet Transform
        Extract $LL_l$ and append to $HL_lA$
    Apply daubechies wavelet tranform for $l = 2$
    Extract $LL_2$ and append to the $DL_lA$
    return $HL_lA$,$DL_2A$
```

Table 5.2: Wavelet Filter Coefficients

DB4	DB8	DB12	DB16
-0.12940952255126	-0.01059740178507	-0.00107730108531	-0.00011747678412
-0.22414386804201	-0.03288301166689	-0.00477725751095	-0.00067544940645
0.83651630373781	0.03084138183556	0.00055384220116	-0.00039174037338
-0.48296291314453	0.18703481171909	0.03158203931749	0.00487035299345
	-0.02798376941686	0.02752286553031	0.00874609404741
	-0.63088076792986	-0.09750160558732	-0.01398102791740
	0.71484657055292	-0.12976686756726	-0.04408825393079
	-0.23037781330890	0.22626469396544	0.01736930100181
		0.31525035170920	0.12874742662048
		-0.75113390802110	-0.00047248457391
		0.49462389039845	-0.28401554296155
		-0.11154074335011	0.01582910525635
			0.58535468365421
			-0.67563073629729
			0.31287159091430
			-0.05441584224310

All the images in the dataset are converted to gray scale. Images are trimmed to avoid white spaces in the boundary. Finally images are resized to 64 rows and 128 columns using bicubic interpolation method. The resized images are fed for multi resolution analysis. Wavelet transform converts the image in to different resolutions and the approximation level contains higher co-efficients. These low frequency components can be the feature for the recognition of handwritten documents. In the multi resolution analysis image is decomposed into four subbands LL, HL, LH and HH are approximate, vertical, horizontal and daigonal features respectively. Changes in both horizontal and vertical directions in terms low frequency is represented by approximate features(LL), LH represents low frequency in horizontal and high frequency in vertical directions, HL represents high frequency in horizontal and low frequency in vertical and HH represents high frequencies in both horizontal and vertical directions. One of the observations after wavelet transform is that wavelet co-efficients are higher in approximation(LL) band. Large wavelet co-efficients are more important than smaller wavelet coefficients. Haar and Daubechis wavelet co-efficients are used as features for our study. In the case of Haar h_ϕ =[0.70710678118655, 0.70710678118655] and h_ψ =[0.70710678118655, -0.70710678118655] , In the case of Daubechis-2 it is h_{ϕ} =[0.48296291314453, 0.83651630373781, 0.22414386804201, -0.12940952255126] and h_{ψ} = [-0.12940952255126, -0.2241, 0.83651630373781, -0.48296291314453] as defined in Equation (5.5) and Equation (5.6) applied to image. In the level 1 decomposition the approximation features will be 32*64 = 2048 and in level 2, it will be 512. Sample decomposition level-2 of word image of കൊല്ലം “kollam”/ is shown in Figure 5.4

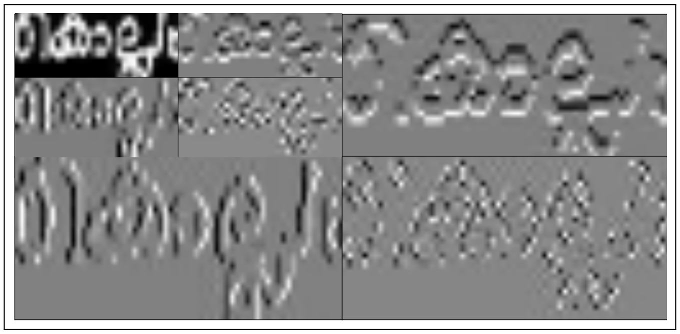


Figure 5.4: Wavelet decomposition using haar wavelet

5.3 Feature Reduction using PCA

Efficient and effective classification can accomplish through minimum features. Several Feature or data reduction methods are available in the literature. Principal Component Analysis(PCA) is a data reduction technique, it transforms original feature space by preserving the orthogonality of the components. It was invented in 1901 by Karl Pearson. It was also studied, developed and discussed later in the 1930s by Harold Hotelling, it is also known as the Hotelling Transform in the quality control jargon. The demand for PCA in most of the applications, some of which are to be commented on below, stems from the problem of high dimensionality which evidently calls for a reformulation of the available variables to select an informative enough subset of linear combinations. For eg. In a risk management targeted financial set-up, a huge (say, in thousands) number of covariate information may be available for a smaller (say, a few hundreds) number of individuals in the sample. Then multiple challenges have to be faced while working with large number of variables. Firstly, an Multiple Linear Regression(MLR) model cannot be directly applied with the usual least squares method, as the design matrix will not generate an invertible scaling matrix. Secondly, even if one uses some more sophisticated mathematical tools (say, a generalized inverse) to avoid the first problem and find a possible solution to the normal equations, the model will most probably be overfit due to the huge number of specifications applied by the full set of variables. Hence, it will underperform when applied on any new (test) set of data other than the original (training) set. Thus, it becomes a desideratum to find out a much smaller set of derived variables that can explain the variability in the response variable as efficiently as possible. Suppose that the variables are expressed as $X_i, i \in 1, \ldots, p$. Let us also denote the vector containing these variables as $\textsc{x}$, with a population dispersion matrix as Σ. Then, a first step of any PCA Algorithm tries to find out a p-dimensional unit normed vector $\textsc{l}$ such that $\textsc{l}^T\textsc{x}$ has the largest possible variance. This is mathematically equivalent to demanding a p-dimensional vector L such that $\frac{L^T \Sigma L}{L^T L}$ has the largest possible value, which is obtained as the largest eigenvalue of Σ, attained when L is the corresponding eigenvector. Iteratively, a PCA Algorithm then attempts to find these eigenvalues in the decreasing order. For a real data setup, we won't have access to Σ in general, hence having to use its sample version S.

A huge class of algorithms are available for executing PCA on various online and offline computing platforms. Most of these procedures eigenvalue analysis techniques including but not limited to the Spectral Decomposition, the Singular Value Decomposition and so on. As mentioned earlier, PCA finds usefulness in diverse situations where dimensionality reduction or some related issues are important. Just as the example mentioned earlier, PCA finds huge scope in quantitative finance[137], including portfolio analysis[138], risk management, predictive modeling, stock subset selection etc. Besides, PCA plays an important role in computer networks[138], neural network procedures and pattern recognition methods.

The limitation of PCA is to find correct direction corresponding to the first principal component, it demands a mean centering in the first step. This however, may create parametrization problems in situation where the variables are necessarily non-negative, say in Neurosciences or in Astronomy. Also, as we are directly using the sample covariance matrix for the calculations, the results are not scale-invariant. PCA is suitable for dimensionality reduction in linearly separable data, where it fails with non-linear data.

The symbolic transformation of test data in our work is shown in Equation: 5.15

$$X_{3400 \times 3780} \Longrightarrow X_{3400 \times 100} \tag{5.15}$$

5.4 Classifiers

5.4.1 Multi Layer Perceptron(MLP)

Evolution of neural network starts in 1943 by Warren McCulloch and Walter Pitts. Later in 1950 Frank Rosenblatt added an extra input called bias and it was capable to learn[139].

Network diagram of simple MLP (Multi Layer Pereptron) shown in Figure 5.5

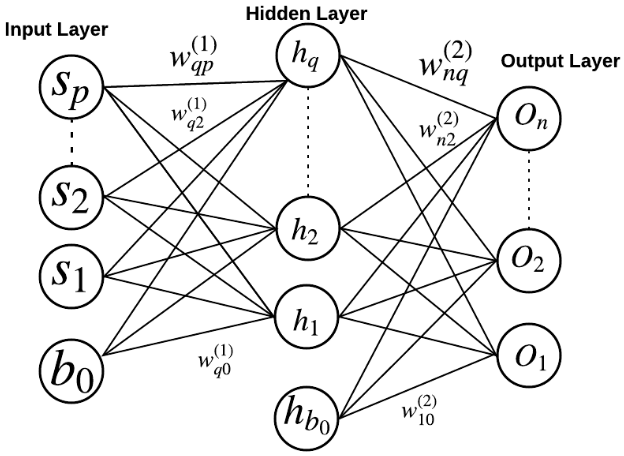


Figure 5.5: Multi Layer Perceptron

Neural network model can consider as a nonlinear function from a set of input variables s_p to a set of output variables o_n controlled by a vector w of adjustable parameters, where w consists of all weight and bias parameters [140].

Input to the hidden layer can be represented by equation 5.16

$$h_q(s, w_p)_{net} = \sum_{i=1}^{p} w_{qi}^{(1)} s_i + w_{q0}^{(1)} \tag{5.16}$$

Hidden Layer uses ReLU as the activation function as shown in equation 5.17.

$$h_q(s, w_p)_{out} = \max(0, h_q(s, w_p)_{net}) \tag{5.17}$$

Input to the hidden layer can be represented by equation 5.18.

$$o_n(s, w_n)_{net} = \sum_{j=1}^{q} w_{nj}^{(2)} \max\left(0, \sum_{i=1}^{p} w_{qi}^{(1)} s_i + w_{q0}^{(1)}\right) + w_{n0}^{(2)} \tag{5.18}$$

Final layer use softmax as the activation as shown in equation 5.19. Let the total number of classes is n

$$o_n(s, w_n)_{out_j} = \frac{e^{o_n(s,w_n)_{net_j}}}{\sum\limits_{i=1}^{n} e^{o_n(s,w_n)_{net_i}}}, \text{ for } j = 1, 2 \ldots n. \tag{5.19}$$

To get the better result all the inputs or features to the Multi Layer Perceptron should be scaled. The hyperparameters that required to mention explicitly are number of neurons in the hidden layer, number of iterations. The number of features will be the number of neurons in the input layer. Scaling can be done through equation 5.20.

$$s_i' = \frac{(s_i - \mu(s))}{\sigma} \tag{5.20}$$

where μ stands for mean, σ stands for standard deviation and s is the input vector

5.4.2 Support Vector Machine (SVM)

SVM's are easy and efficient method for handwriting recognition[141]. It is versatile, depends upon the classification problem different kernels can use. The basic kernels are linear, Polynomial, RBF(Radial Bais Function) and sigmoid. In addition to this one can define custom kernels also. These kernels have different parameters and in our experiment RBF kernel[122] with Gridsearch method for parameter selection. Detailed theoretical explanation is given in Chapter 4.

5.4.3 Random Forest(RF)

RF is knows as an ensemble method for classification with lot of decision trees. RF widely used in remote sensing [142], image classification [143] and document analysis [144]. In Random Forest, Random means randomness in the selection of data and randomness in the split and forest means a bunch of trees. Let these trees be denoted as T_t, where $t \in \{1, 2..., n\}$, $\{T_t\}_1^n$ is the ensemble of n trees. suppose an input x, to classify a given sample, x calculate the tree average as shown in equation 5.21 [145].

$$P(y_i|x) = \frac{1}{n}\sum_{t=1}^{n} P_t(y_i|x) \tag{5.21}$$

$$class(x) = \arg\max p(y_i|x) \tag{5.22}$$

For ensemble of 500 trees to predict 330 classes the probability for the prediction to certain classes is the average of the predictions of the leafnodes of 500 trees shown in equation 5.23.

$$p(y_i|x) = \frac{1}{500}\sum_{t=1}^{500} P_t(y_i|x), \text{where } i \in 1, ...330 \tag{5.23}$$

where x is the features of the word image given for testing and y_i is the class label.

5.4.4 Voting Classifier

The concept of the voting classifier is to associate or combine various machine learning classifiers. Equation 5.24 shows the process of voting [139].

$$y_i = \sum_{n=1}^{N} w_n d_{ni} \text{ where } w_n \geq 0, \sum_{n=1}^{N} w_n = 1 \tag{5.24}$$

where w_n stands for the weight associated with each classifier,d_{ni} is the vote of classifier n for class i and N is the total number of classifiers. In our implementation, we use Multi Layer Perceptron, Support Vector Machine (SVM) and Random Forest(RF). Voting scheme is plurality with uniform weights. This kind of approach balance the weakness of individual classifier. Schematic daigram of voting classifier is shown in figure 5.6.

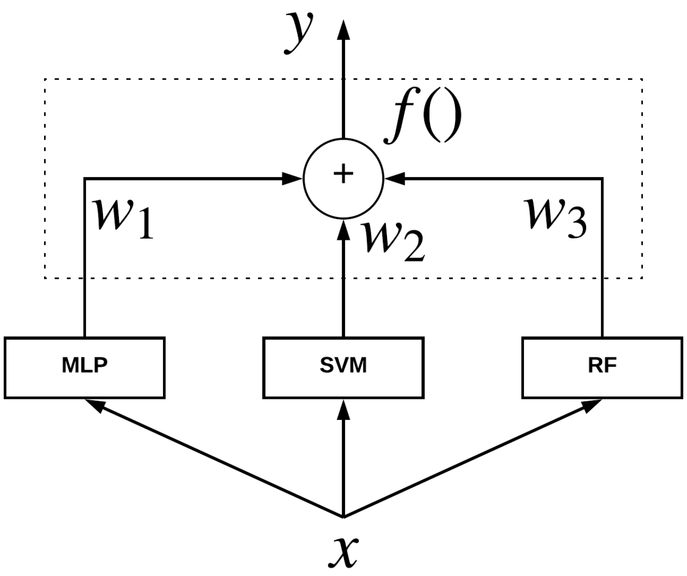


Figure 5.6: Base classifiers are MLP, SVM, RF and their outputs are combined using $f(.)$

5.5 Result Analysis and Performance Evaluation

In this section, we consider HOG, PHOG features descriptor and Wavelet co-efficients as the features. Classification are performed through SVM, MLP and RF.

5.5.1 Performance Evaluation - Traditional Methods

Detailed Analysis of the traditional methods with JMHRDB1, Hindi and Marathi dataset[45] are following.

Evaluation using HOG with JMHRDB1

HOG features are evaluated with various parameters like block stride and number of bins. Here the number bins will decide the number of directions, For example if the number of bins, nbin=6, then the directions it will consider is $0°, 30°, 60°, 90°, 120° and 150°$. If the block stride is (8,8), block size=(16,16) and nbin=5 , then for a 64 × 128 image, Number of features extracted is 2520. In our experiment HOG with $nbin = 6$ and $nbin = 7$ provide almost same result with block stride (8,8) as shown in Table 5.3. HOG feature vector with nbins=7,block stride(8,8) gives the best result. Number of features are 2940, with SVM it achieves 89.4% accuracy.

Table 5.3: Result Analysis Of HOG with JMHRDB1

No.Of Bins or Direction	Block Stride	No.Of Features	SVM	MLP	Random Forest	Voting Classifier
7	**(8,8)**	**2940**	**89.4**	**85**	**79**	**87.7**
6	**(8,8)**	**2520**	**89.2**	**84.11**	**80.7**	**87.1**
5	(8,8)	2100	87.3	82.2	78.4	85.1
7	(16,16)	896	85.6	80.3	77.4	84.4
6	(16,16)	768	85.2	79.5	76.9	82.5
5	(16,16)	640	83	77.4	74.7	81

The performance with SVM, MLP, RF and voting classifier are shown in Figure 5.7.

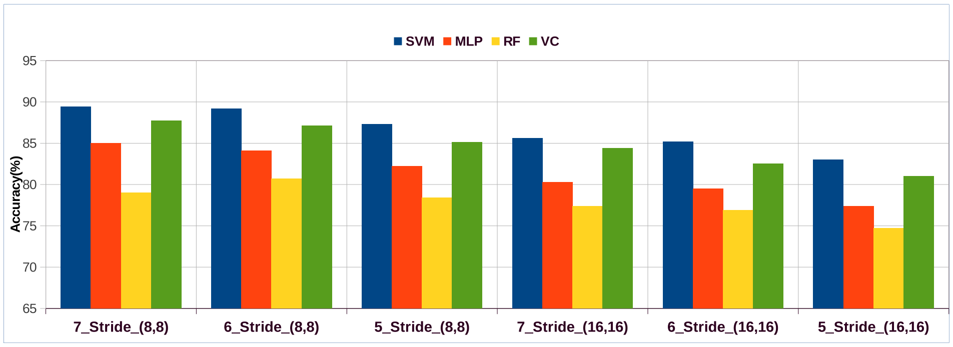


Figure 5.7: Performance Evaluation of HOG with JMHRDB1

PCA implemented for dimensionality reduction. Results with reduced HoG features extracted through the following parameters, Number of Bins:7 and stride:$(8, 8)$ using PCA are shown in Table 5.4. Comparison of the results furnished in Table 5.4, with reduced HOG Features of 1000, 500, 200 and 100 features are shown in Figure 5.8.

Table 5.4: Result Analysis−HOG With PCA In JMHRDB1

No.Of Bins or Direction	Block Stride	Reduced Features (PCA)	SVM	MLP	Random Forest	Voting Classifier
7	**(8,8)**	**1000**	**89.2**	**84.4**	**78.4**	**87.5**
7	**(8,8)**	**500**	**89.1**	**83.24**	**79.7**	**86.9**
7	(8,8)	200	88.4	81.8	77.4	84.5
7	(8,8)	100	85	77.2	76.2	82

Figure 5.8 shows that the performance of the classifier are steady upto 200 reduced features with PCA after that it degrades.

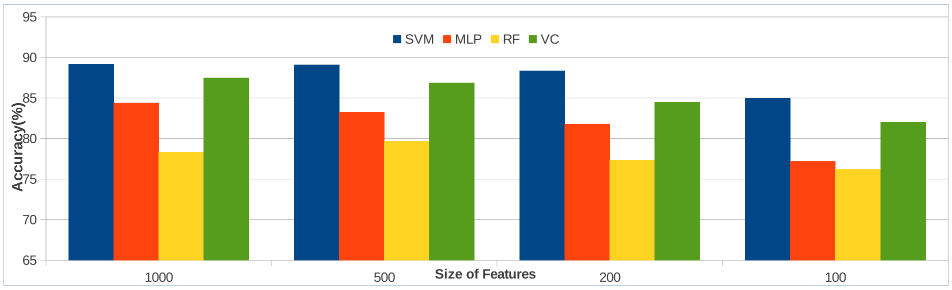


Figure 5.8: Performance Evaluation of HOG with PCA In JMHRDB1

Evaluation using Wavelets with JMHRDB1

The results obtained using Haar wavelets with decomposition levels-1&2 and Daubechis wavelet with level-2 as explained in Section: 5.2.3 are given in Table 5.5.

Table 5.5: Result Analysis−Wavelet with JMHRDB1

Wavelet Type	Level	Number of Features	SVM	MLP	RF	Voting Classifier
Haar	1	2048	85	83.2	81	84
Haar	**2**	**512**	**86.2**	**84**	**82**	**85.3**
Daubechis	2	512	83	80	76	81.8

Haar Wavelet provides better result with SVM. The performance of Haar and Daubechis Wavelet with JMHRDB1 are shown in Figure: 5.9.

Classification through reduced Haar wavelet features using PCA is shown in Table 5.6.

Table 5.6: Result Analysis− Wavelet with PCA

Wavelet Type	Level	Reduced Features	SVM	MLP	RF	Voting Classifier
Haar	2	200	85.1	81.3	79	83.9
Haar	2	100	83.1	79.3	76.5	80.5

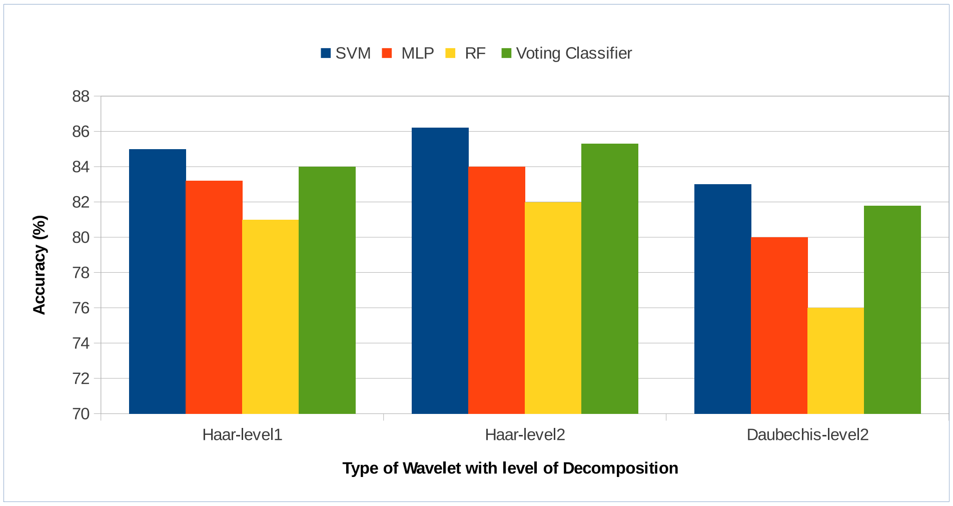


Figure 5.9: Performance Evaluation of Wavelet with JMHRDB1

Performance with reduced wavlet features from 512 to 200 and then 512 to 100 is shown in Figure 5.10.

Evaluation using PHOG with JMHRDB1

PHOG features are calulated in pyramid level -1 and pyramid level-2 as explained in Section 5.2.2. Results are shown in Table 5.7.

Table 5.7: Result Analysis−PHOG

No. Of Levels	Features	SVM	MLP	RF	Voting Classifier
2	3024	90.1	85.6	82	87.9
3	3096	90.6	85.9	82.4	88

Figure 5.11 shows the performance between level 1 and 2 pyramid level representaion with PHOG feature descriptor.

Results with the reduced features of PHOG descriptors are shown in Table 5.8.

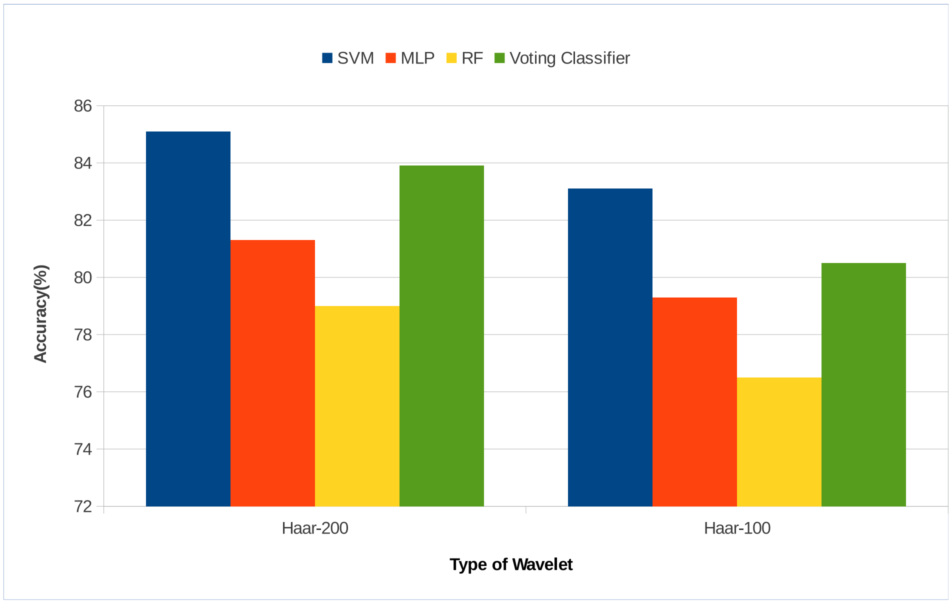


Figure 5.10: Comparison of Wavelet with PCA

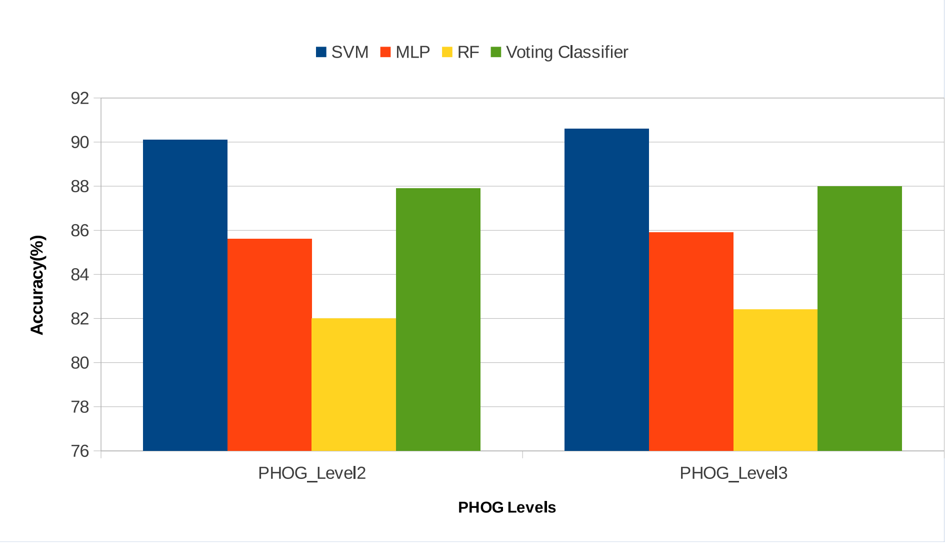


Figure 5.11: Performance Evaluation of PHOG with JMHRDB1

Table 5.8: Result Analysis–PHOG with PCA in JMHRDB1

No. Of Levels	Reduced Features	SVM	MLP	RF	Voting Classifier
3	1000	90.1	85	82	87.5
3	500	90	84.4	81.2	87
3	200	89.5	83.8	80.4	85.9
3	100	87.8	82	78	83.8

Performance with PCA of PHOG features are shown in Figure: 5.12. PHOG features performed well with SVM. PCA with 1000 features gives 90.1 % accuracy, which is only 0.5 % less than without PCA. The best performed combination in all the experiments discussed in previous sections are simulated with Hindi and Marathi Dataset. Result analysis and performance evaluation are explained in the subsequent sections.

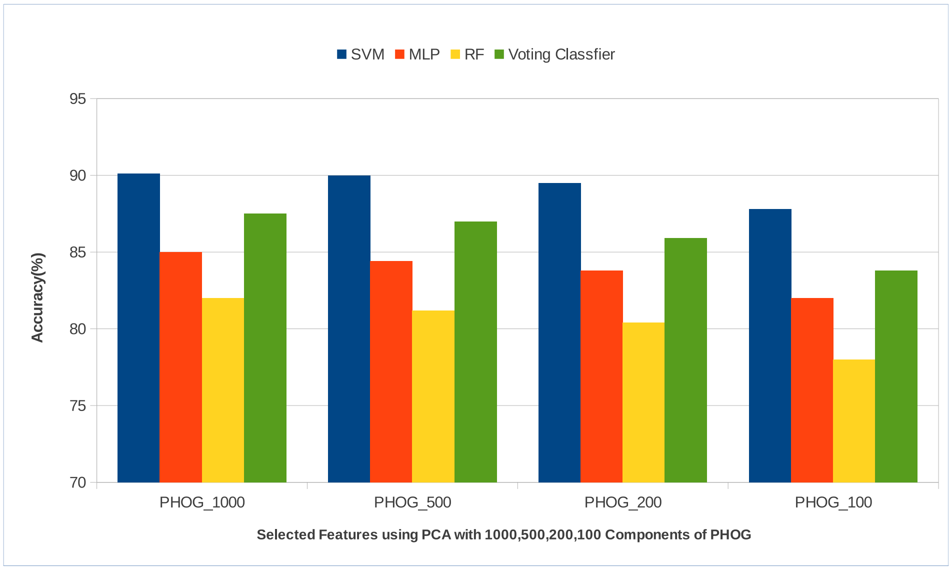


Figure 5.12: Performance Evaluation–PHOG with PCA in JMHRDB1

Performance Evaluation - HOG with Hindi and Marathi Datset

Results with Hindi and Marathi Dataset using HoG (Number of bins=7 and block stride=(8,8)) as features are shown in Table:5.9.

Table 5.9: Result Analysis−HoG with Hindi and Marathi Dataset

DataSet	SVM	MLP	RF	Voting Classifier
Hindi	82	80.4	78	81.2
Marathi-DB1	80	79	74.3	79
Marathi-DB2	78	76	73.4	77.3

Performance of the Hindi-Marathi Dataset with HoG shown in Figure: 5.14.

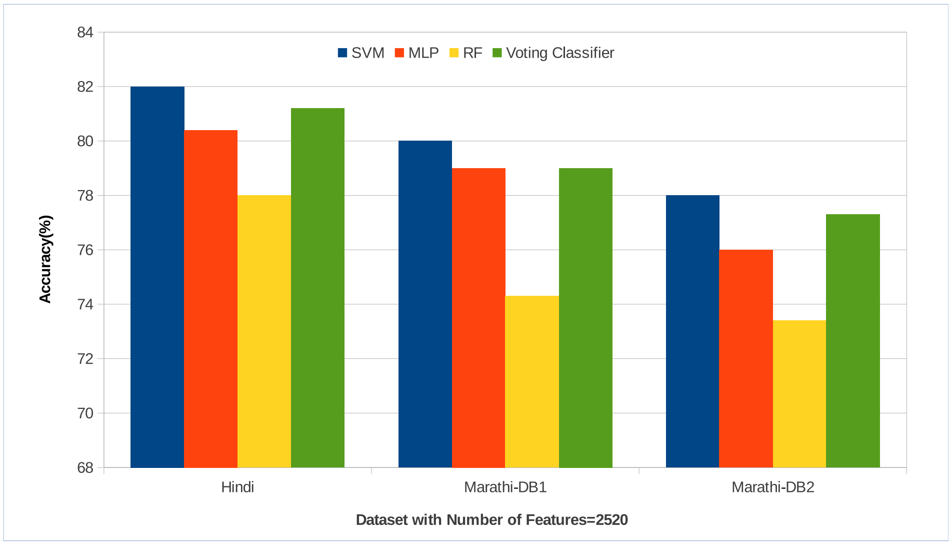


Figure 5.13: Performance Evaluation of HOG Features with Hindi and Marathi Dataset

Features are reduced to 200 and the results with SVM, Multi Layer Perceptron and Random Forest are shown in Table:5.10

Performance with the above mentioned classifiers after applying PCA

Table 5.10: Result Analaysis−HoG with PCA in Hindi and Marathi dataset

DataSet	SVM	MLP	RF	Voting Classifier
Hindi	81.5	79	76.5	81
Marathi-DB1	79	77	73.1	78
Marathi-DB2	77	74.5	72.2	76.4

is shown in Figure 5.14.

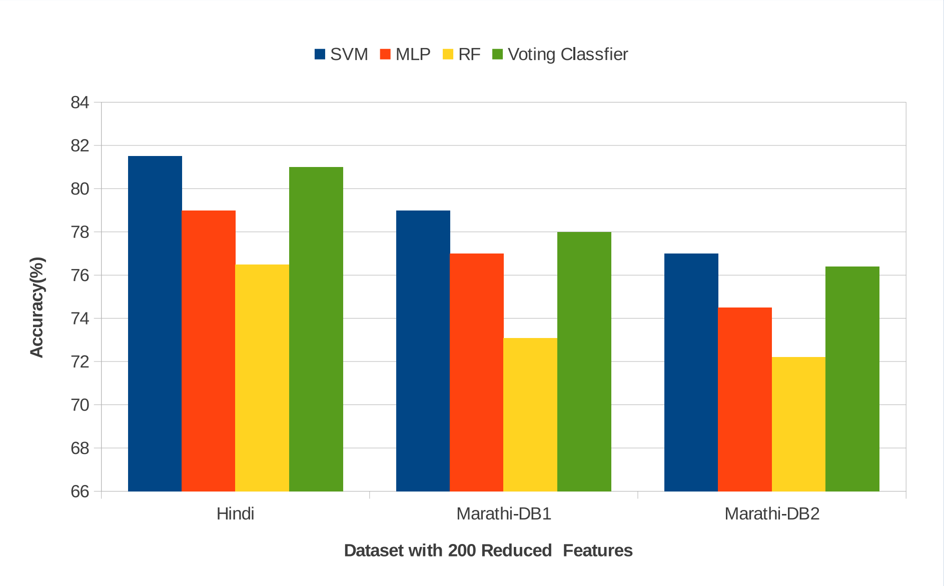


Figure 5.14: Performance Evaluation of HOG Features with PCA in Hindi and Marathi Dataset

Performance Evaluation -Wavelet with Hindi and Marathi Datset

Result Analysis of Hindi and Marathi dataset with Haar and Daubechis wavelet- level2 decomposition is shown in Table 5.11.

Performance of the classifiers with Haar and Daubechis Wavelet is shown in Figure 5.15.

Table 5.11: Result Analysis−Wavelet with Hindi and Marathi dataset

DataSet	Wavelet Type	SVM	MLP	RF	Voting Classifier
Hindi	Haar	80.5	78.5	77	80
	Daubechis	78	76	74	77.5
Marathi-DB1	Haar	77	75	73.1	76.5
	Daubechis	74	70	71.2	73.8
Marathi-DB2	Haar	74.5	73.9	71.5	74.2
	Daubechis	73.1	71.2	70.3	73

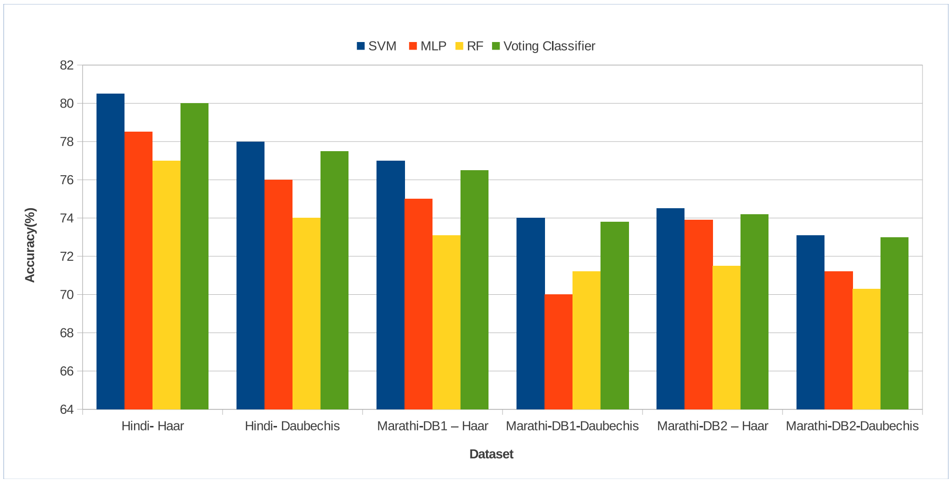


Figure 5.15: Performance Evaluation− Wavelet Features with Hindi and Marathi Dataset

Result Analysis After Appying PCA is shown in Table 5.12.

Table 5.12: Result Analysis−Wavelet with PCA in Hindi and Marathi dataset

DataSet	Wavelet Type	SVM	MLP	RF	Voting Classifier
Hindi	Haar	80	78.5	77.4	79
Marathi-DB1	Haar	76.5	74.3	73.2	76
Marathi-DB2	Haar	74	72	71.1	72.9

The performance of the SVM, MLP and RF with reduced 200 features of wavelet with Hindi-Marathi dataset is shown in figure:5.16.

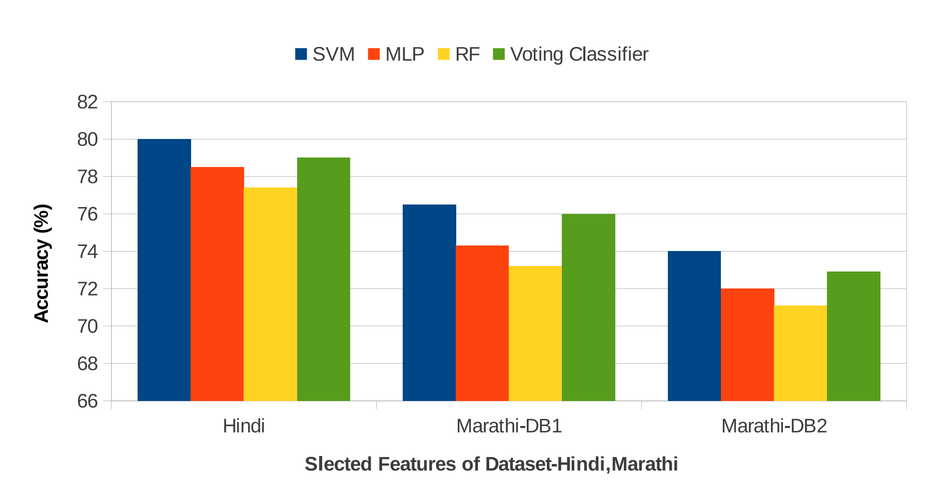


Figure 5.16: Performance Evaluation– Wavelet with PCA in Hindi and Marathi Dataset

Performance Evaluation -PHOG with Hindi and Marathi Datset

Result Analysis With PHOG Features of Hindi and Marathi Dataset are shown in Table 5.13.

Table 5.13: Result Analysis–PHOG in Hindi and Marathi Dataset

DataSet	Pyramid Level	No. Of Features	SVM	MLP	RF	Voting Classifier
Hindi	Level 2	3024	84.1	81	79	82.5
	Level 3	3096	84.7	81.4	79.3	83.4
Marathi-DB1	Level 2	3024	80.4	80	75	81
	Level 3	3096	80.9	80.2	75.6	81
Marathi-DB2	Level 2	3024	79	77	74.4	77.7
	Level 3	3096	79.7	77.5	74.9	79.1

Figure 5.17 shows the performance analysis of PHOG Features with

Table 5.14: Result Analysis–PHOG with PCA in Hindi and Marathi Dataset

DataSet	Pyramid Level	SVM	MLP	RF	Voting Classifier
Hindi	Level 3	84	80.9	78.6	83.1
Marathi-DB1	Level 3	80.2	79.4	75	80.5
Marathi-DB2	Level 3	79	76.5	74	77.3

Hindi and Marathi Dataset.

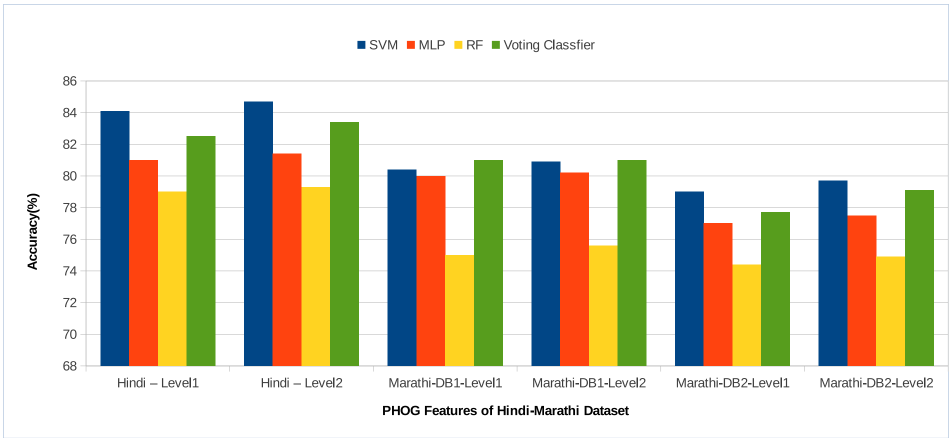


Figure 5.17: Performance Evaluation– PHOG in Hindi and Marathi Dataset

PHOG Features are reduced using PCA. The results with 200 Features of Hindi-Marathi dataset are shown in Table 5.14.

Performance of the PHOG in Hindi and Marathi dataset with Reduced dimensions using PCA is shown in Figure 5.18.

PHOG features extracted from level–3 pyramid representation of the image gives the best result along with SVM. The results achieved for Hindi, Marathi-DB1, Marathi-DB2 are 84.7%,80.9% and 79.7% respectively.

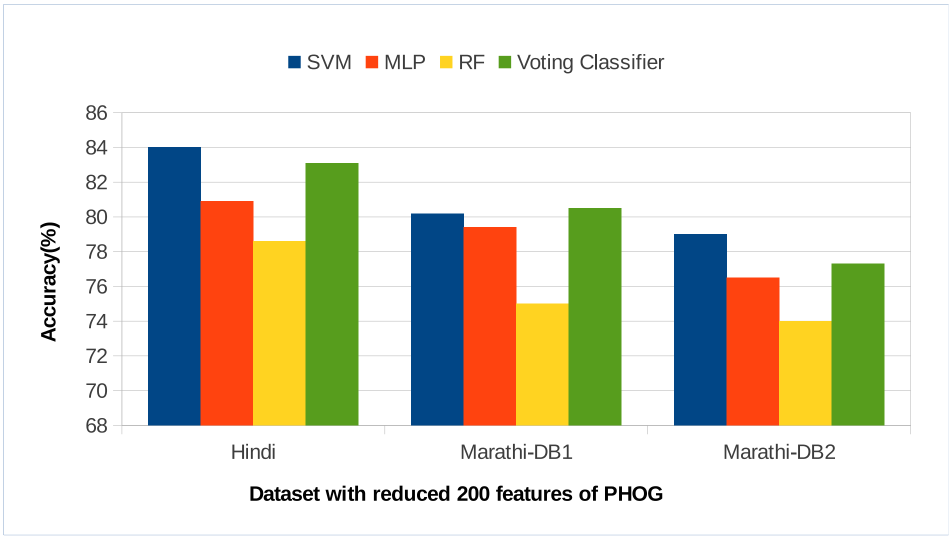


Figure 5.18: Comparison of PHOG Features with Hindi and Marathi Dataset

5.5.2 Performance Comparison-DL/Traditional

Performance comparison of the best performed traditional methods discussed in this chapter and methods discussed in Chapter 4 are shown in Figure 5.19.

In traditional methods PHOG feature descriptor with SVM as the classfier provides an accuracy of 90.6% with JMHR DB1, 84.7% accuracy with Hindi dataset. For Marathi it gives 80.9% and 79.7% respectively for DB1 and DB2. Architectures based on deep learning methods provide better result compared to classical/ traditional machine learning approaches. The hybrid architecture consisting of CNN and SVM provided 96.90% accuracy with JMHRDB1, 97.53% accuracy with a 14 layer resnet architecture, which is 0.63% improvement than CNN hybrid model. Two stage approach gives further improvement and provide an accuracy of 98.08% with JMHRDB1.

The CNN-SVM hybrid architecture provided 94% accuracy on the test set of Hindi legal amount word databse, which improves the existing state-

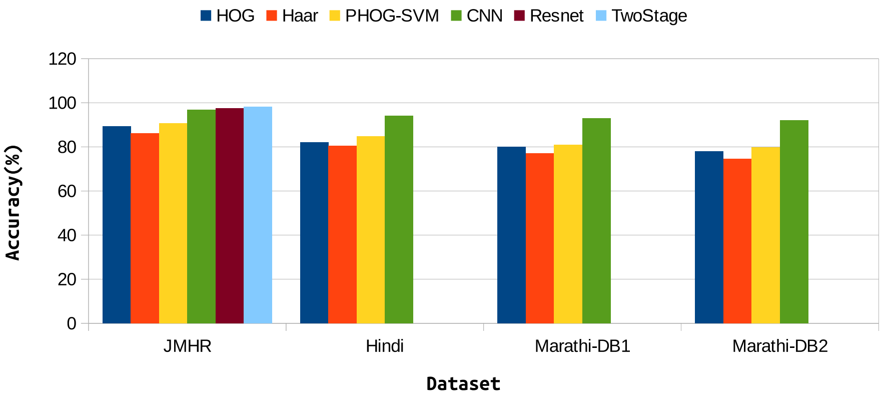


Figure 5.19: Performance Evaluation Traditional Vs. DL based

of-the-art accuracy value published in [45]. With the same architecture we could obtain 93% and 92% recognition accuracies on Marathi DB1 and DB2 respectively, which improved the existing respective state-of-the-art accuracy values of 85.78% and 78.79%[45].

Experiments on Resnet and two-stage classification are only performed with JMHRDB1 because our focus of research is on Malayalam handwriting recognition. Still the other experiments show that script independent recogntion is possible by the selection of a proper architecture or script independent features.

Comparison of the classical and deep learning based methods with the recent literature is shown in Table 5.15. In traditional machine learning method final classfier is SVM with PHOG as features. In deep learning method CNN with two stage classifiers provided better result.

Table 5.15: Comaparison of Accuracy with related works

Author	**Features**	**Classifier**	**Language/script**	**Accuracy**
Bhowmik et.al[2018]	Shape Based	MLP	Bangla	79.87
Present Work	HoG	MLP	Malayalam	85
Dutta et.al[2018]	CNN Extracted	Recurrent Architecture	Bangla	95.7
Present Work	**CNN Extracted**	**Two stage Classfier**	**Malayalam**	**98.08**
Present Work	**PHOG**	**SVM**	**Malayalam**	**90.6**
Roy et.al[2016]	PHOG	HMM & SVM	Hindi	84.24

5.6 Summary

Traditional methods of machine learning performs better with less number of samples and clearly defined features. Handwriting recogntion is a complex task because of the large inter and intra class variance of the samples or features. If a large number of samples are not there, synthetic data is an obvious choice for training the model/architecture. In this chapter,we discussed traditional methods of feature extraction- HOG, PHOG and Wavelet also classifiers - MLP, SVM and Random Forest. Deep methods discussed in Chapter:4 are compared with traditional methods. we observe that deep methods provide better result. The main reason behind it is the abilty of deep architectures to generalize the model for a comparatively higher number of classes. All the experiments discussed in this section are implemented using python script.

Chapter 6

Lexicon Free Recognition

The atoms may be compared to the letters of the alphabet, which can be put together into innumerable ways to form words.

William Henry Bragg

6.1 Introduction

Offline handwriting recognition converts an image of handwritten text as editable unicode representation. Proper segmentation to words or characters from a text is difficult task. The recognition accuracy heavily depends upon the segmentation module. Sayre's paradox [146] states that segmentation based recognition methods creates deadlock because both are dependent. This chapter discusses the segmentation free method for the recognition of words from the documents. Another advantage is the proposed method is lexicon free. The system will try to identify the character by character from a sequence and this process is known as sequence labelling. Recurrent Neural Network is suitable for sequence labelling but it is not capable of remember long term dependencies [20]. Long Short

Term Memory (LSTM) is capable of remembering long term dependencies and the same along with a Connectionist Temporal Classification (CTC) architecture had recently provided good recognition performance in an online Bangla handwriting recognition task[147]. Also, a hybrid architecture consisting of CNN, BLSTM and CTC has been successfully used in [148] for online handwriting recognition of Devanagari and Bangla, the two most popular Indian scripts.

6.2 CNN-LSTM-CTC (CLC) Hybrid Architecture

Here, we present our study of lexicon free recognition of offline handwritten Malayalam words. The neural network architecture used for this study consists of two convolution layers, followed by a BLSTM layer and finally a CTC transcription layer. The implementation of a similar architecture available at[149] is used in this study. A block diagram of this hybrid architecture is shown in Figure 6.1.Convolutional layers are used for feature extraction. This features are used for sequence learning from the raw handwritten image data. In our experiments we use more convloutional layers for feature extraction, but we obtianed best performance with two convolutional layers. Some further details of various components of this hybrid architecture are described below.

6.2.1 CNN

CNN is used to extract the features. Final output of CNN(2^nd Convolutional Layer) give as a one dimensional sequence to BLSTM. Theoretical explanation about CNN is given in Chapter 4.

6.2.2 Recurrent Neural Network(RNN)

RNN is suitable for applications with sequential data. It can process images with variable size like image of handwritten text. The architecture of the

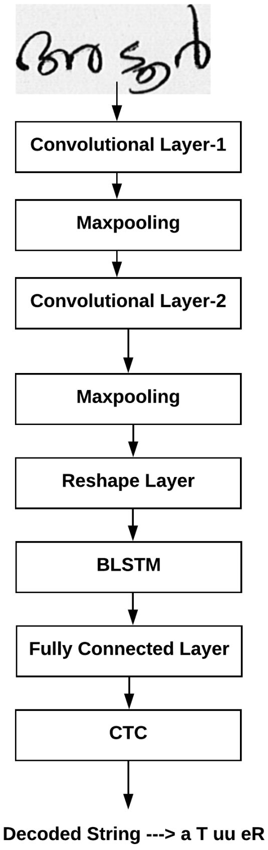


Figure 6.1: Best Scenario – CLC Hybrid Architecture decoded the image to “a T uu eR” / “അടൂർ”

RNN with unfolding over time step is shown in Figure 6.2.

suppose the input sequence $(s_0, s_1, ...s_{T-1})$, produces the hidden states of the recurrent layer $(h_0, h_1, ...h_{T-1})$ and the output of a single hidden layer in RNN $(O_0, O_1, ...O_{T-1})$ can be derived as follows [150][151]

$$h_t = tanh(W_{sh}s_t + W_{hh}h_{t-1} + b_h) \tag{6.1}$$

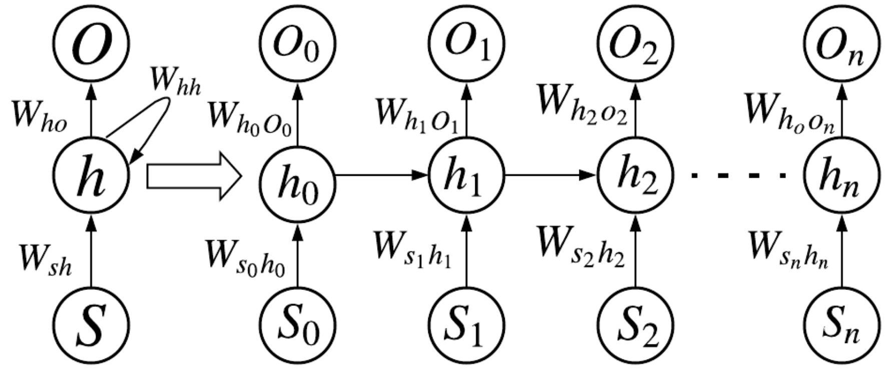


Figure 6.2: RNN unfolding over time

$$O_t = (W_{ho}h_t + b_o) \tag{6.2}$$

Where W_{sh}, W_{hh}, W_{ho} denotes the connection weights from the input layer s to the hidden layer h, the hidden layer h to itself and the hidden layer to output layer. b_h and b_o are the two bias vectors. The drawback of RNN is the vanishing gradient problem [152]. so in our experiments, we use LSTM - One of the variants of RNN.

6.2.3 LSTM

Long Short Term Memory called LSTM is suitable for machine learning applications that require gradient flow for longer durations[153]. LSTM's are a particular type of Recurrent Neural Network. LSTM can solve the drawback of RNN by the introduction of three gates called input gate,forget gate and output gate. The mathematical representation for basic LSTM is shown below.

$$InputGate : i_t = \sigma(W_{si}s_t + W_{hi}h_{t-1} + b_i) \tag{6.3}$$

$$ForgetGate : f_t = \sigma(W_{sf}s_t + W_{hf}h_{t-1} + b_f) \tag{6.4}$$

$$OutputGate : o_t = \sigma(W_{so}s_t + W_{ho}h_{t-1} + b_o) \tag{6.5}$$

$$InputTransform : c_i^t = tanh(W_{sc}s_t + W_{hc}h_{t-1} + b_{c_i}) \tag{6.6}$$

$$stateupdate : c_t = f_t \times c_{t-1} + i_t \times c_i^t \tag{6.7}$$

$$and \quad h_t = o_t \times tanh(c_t) \tag{6.8}$$

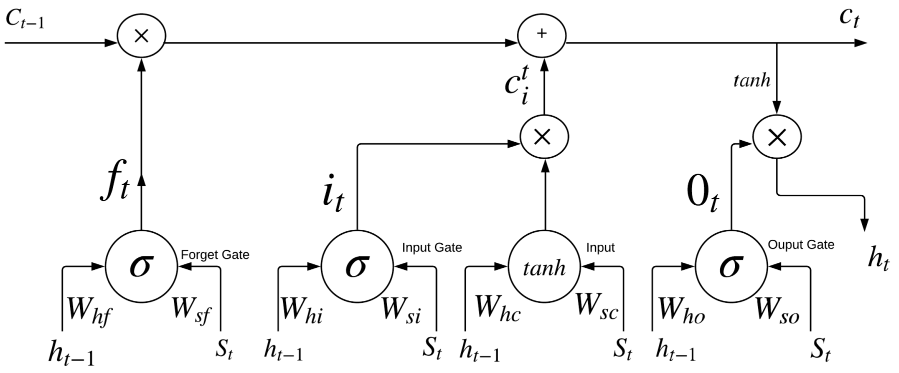


Figure 6.3: Block Diagram of LSTM cell - '+', '×' are pointwise addition and mulitplication operators

Unidirectional LSTM and BLSTM are differ in respect of treating the input sequence data, in the former case it consider sequences from beginning to end, later case it consider from end to beginning of the sequence.

6.2.4 CTC

CTC is used for the purpose of transcription of the output of LSTM to character labels. Without any post processing module it can directly decode the input sequence to the output symbol viz. CTC do post processing after recognition at each time step. Here the symbol can be a character,word or line in the handwriting context. The method that CTC follows is that it simply selects the most probable symbol at each time then it merges the adjacent repeated symbols in the final output. In this case it cannot distinguish extended symbols and repeated symbols. To avoid this softmax layer in CTC consists of one additional symbol other than total alphabets. If the alphabet, α and it's size, $|\alpha|$, then output layer will consist of $|\alpha|+1$ units. Extra unit is a blank token to handle the repeated graphemes in the script. For a given input sequence of feature vectors, the CTC layer will predict the probability of output label sequence. Suppose $(i_1, i_2, i_3, i_4, ..., i_n)$ is the input sequence and $(o_1, o_2, o_3, ..., o_m)$ is the output sequence, where $m \leq n$. The target sequence $\mathbb{Z} = \alpha *$ is the set of all possible sequences over the alphabet α. For an input sequence i of length L, probability of a given output sequence or path can be defined as

$$p(\delta|i) = \prod_{t=1}^{T} y^t_{\delta_t}, \forall \delta \alpha'^T \tag{6.9}$$

where $\alpha' = \alpha \cup \{blank\}$and T is the length of sequence. δ is the number of elements in α'^T as paths. Next step is to find out the exact path from the many possible paths.$\mu : \alpha'^T \mapsto \alpha^{(\leq T)}$, where the set of possible labelling denoted by $\alpha^{(\leq T)}$. eg: $\mu(aaaa\_TaTa\_UUU\_rr) = aTaUr$. We can represent the probabilities of a given label $l \in \alpha^{(\leq T)}$ as

$$p(l|i) = \sum_{\delta \in \mu^{-1}(l)} P(\delta|i) \tag{6.10}$$

output of the classifier $h(x) = \arg\max_{l \in \alpha^{(\leq T)}} = p(l|i)$

Objective function of CTC is defined as the negative log probability of the network follows during training

$$\mathbb{O} = - \sum_{(i,\tau) \in T} \log p(\tau|i) \tag{6.11}$$

where T is the total training set, i is the input sequence and τ is the target label assigned to the sequence. Network weights are updated using Back Propogation Through time in CLC network. CTC performs implicitis language modeliing using best path decoding method.

$$\delta^* = \arg\max_{\delta} p(\delta|i) \tag{6.12}$$

The number of paths will exponentialy increase and it is directly proporional to the sequence length. So the CTC employs forward-backward algorithm for perfect output decoding to a most likely sequence of characters.

6.2.5 Grapheme Level Representation

Grapheme level representation based on the relative position of individual characters or syllables in the word. CTC designed with a proper unicode mapping, still semi ortho-syllable representation [154] provides better results with handwriting recognition in Bangla, where the represntation is based on syllables. The order of the appearance of unicode is not in the same way as they appear in Indian Languages. For example the word: കൊച്ചി/kochi can be represented in the unicode label as $u'\backslash u0D15' + u'\backslash u0D4A' + u'\backslash u0D1A' + u'\backslash u0D4D' + u'\backslash u0D1A' + u'\backslash u0D3F'$. Here കൊ = ക+ ഒ $= u'\backslash u0D15' + u'\backslash u0D4A'$ with two unicodes where in grapheme level it is three. "ച്ച/Compound Character 'cc' " represents $u'\backslash u0D1A' + u'\backslash u0D4D' + u'\backslash u0D1A'$ with three unicodes where in grapheme level it is one. In other words grapheme level labelling makes decoding simpler and it will be suitable for datsets with relatively small number of samples. Mapping of the Malayalam characters to the grapheme level representation is shown in Table 6.1.

Table 6.1: Grapheme level Mapping♣

Si. No	Grapheme	Representation	Si. No	Grapheme	Representation	Si. No	Grapheme	Represention
1	അ	a	2	ആ	aa	3	ഇ	ie
4	ഉ	u	5	ഋ	eru	6	എ	ea
7	ഏ	eaa	8	ഒ	o	9	ക	k
10	ഖ	kha	11	ഗ	ga	12	ഘ	ekka
13	ച	c	14	ഛ	cha	15	ജ	ja
16	ഝ	jha	17	ഞ	nja	18	ങ	Ga
19	ട	T	20	ഠ	TDa	21	ഡ	Da
22	ഢ	DA	23	ണ	N	24	ത	t
25	ഥ	idha	26	ദ	da	27	ധ	dha
28	ന	na	29	പ	p	30	ഫ	ph
31	ബ	b	32	ഭ	bh	33	മ	ma
34	യ	ya	35	ര	ra	36	ല	la
37	വ	va	38	ശ	sh	39	ഷ	sha
40	സ	sa	41	ഹ	ha	42	ള	La
43	ഴ	zha	44	റ	R	45	ൺ	eN
46	ൻ	en	47	ൽ	l	48	ർ	eR
49	ൾ	eL	50	ക്ക	kk	51	ങ്ക	nka
52	ങ്ങ	nga	53	ച്ച	cc	54	ഞ്ച	nch
55	ഞ്ഞ	nna	56	ട്ട	TT	57	ണ്ട	NDa
58	ണ്ണ	NN	59	ത്ത	tt	60	ന്ത	nta
61	ന്ന	nn	62	പ്പ	pp	63	മ്പ	npa
64	മ്മ	mm	65	യ്യ	yy	66	ല്ല	ll
67	വ്വ	vv	68	ള്ള	LL	69	ക്ഷ	ksha
70	റ്റ	RR	71	ന്ദ	nda	72	ന്ധ	ndha
73	ദ്ദ	dada	74	ദ്ധ	dadha	75	◌ാ	ar
76	◌ി	i	77	◌ീ	ii	78	◌ു	u
79	◌ൂ	uu	80	◌ൃ	er	81	െ◌	e
82	േ◌	le	83		iee	84	ം	m
85	ഃ	two	86	്ര	lR	87	്വ	uva
88	്യ	eya	89	്	eu	90	ഗ്ഗ	gaga
91	ജ്ജ	jaja	92	സ്ഥ	saidha	93	ശ്ശ	shsh
94	ച്ഛ	ccha	95	ശ്ല	shLa	96	പ്ല	pLa
97	സ്ല	SaLa	98	മ്ല	maLa	99	ഹ്ല	haLa
100	ഗ്ല	gaLa	101	ക്ല	kLa	102	ബ്ല	bLa
103	ശ്മ	shma	104	ൺഠ	NTDa			

♣Light gray circles show the position of a consonant character

6.2.6 Experimental Setup

All the images in the dataset is resized with a fixed height of 64 without changing the aspect ratio. The maximum width of the image in the dataset is calculated and pad with zero's on the right side of the image for make all the samples in equal size before feeding to CNN. For the experiments the dataset is divided in to 5−folds and perform cross validation. The input to the CLC network is batch normalized[125] for training. Convolutional Layer 1 uses 128 filters with stride 2, where filter height is 64 and width is 5. Covolutional Layer 2 uses 64 filters with stride1, where filter width is 5 and height is 1. Both Convolutional Layers are followed by Maxpooling with stride 2 and 1 respectively. BLSTM layer uses 64 hidden nodes. Final layer is CTC with softmax operation. Greedy decoding is used to implement the best path approach. The network is trained using CTC loss and Adam optimizer with a learning rate of 0.0001. Batch size used for training is 128. All the experiments are implemented in a machine with Intel Core i7-4770 CPU@3.40GHz x 8cores with 16GB RAM using python and tensorflow.

6.2.7 Post Processing

The predicted words or character sequences are corrected using dictionary. Levenshtein Edit Distance method [155]is used to find the closest word to the predicted word and find the error rates. It calculates the number of edits in terms of insertion, substitution and deletion required to get the actual word from the predicted word. Here the actual words are included in the dictionary. Along with BK tree [156] representaion of the dictionary it finds the words that matches.

6.3 Result Analysis

The evaluation can be based on two aspects either character level (Character Error Rate a.k.a CER) or Word level(Word Error Rate a.k.a WER).

CER of the proposed approach is shown in Table: 6.2. CER can be calulated using the equation given below

$$\text{Let Number of Substitutions} = ns$$

$$\text{Number of Insertions} = ni$$

$$\text{Number of Deletions} = nd$$

$$CER = \frac{ns + nd + ni}{\text{Total number of characters in the reference}} \times 100 \tag{6.13}$$

$CER(\text{കൊച്ചി}, \text{കൊച്ച്}) = \frac{1+0+0}{5} \times 100 = 20\%$

$CharacterLevelAccuracy(Char_{Acc}) = 100 - CER = 80\%$

To increase the reliability and generalize the model, evalution method used is k-Fold cross validation, where we experimented with $k \in 1, 2, 3$. The training and validation pairs can be denoted as : $\{Train_i, Test_i\}_{i=1}^{k}$ Dataset $\mathbb{D}$ is divided into 5 folds $\mathbb{D}_1, \mathbb{D}_2, \mathbb{D}_3, \mathbb{D}_4 and \mathbb{D}_5$. 20 % of each training folds are used as validation set. The distribution of Training/ Test dataset for 5 fold cross validation is

$$Train_1 = \mathbb{D}_2 \cup \mathbb{D}_3 \cup \mathbb{D}_4 \cup \mathbb{D}_5 \quad Test_1 = \mathbb{D}_1$$

$$Train_2 = \mathbb{D}_1 \cup \mathbb{D}_3 \cup \mathbb{D}_4 \cup \mathbb{D}_5 \quad Test_2 = \mathbb{D}_2$$

$$Train_3 = \mathbb{D}_1 \cup \mathbb{D}_2 \cup \mathbb{D}_4 \cup \mathbb{D}_5 \quad Test_3 = \mathbb{D}_3$$

$$Train_4 = \mathbb{D}_1 \cup \mathbb{D}_2 \cup \mathbb{D}_3 \cup \mathbb{D}_5 \quad Test_4 = \mathbb{D}_4$$

$$Train_5 = \mathbb{D}_1 \cup \mathbb{D}_2 \cup \mathbb{D}_3 \cup \mathbb{D}_4 \quad Test_5 = \mathbb{D}_5$$

The Percentage of results with $k-$ fold cross validation is shown in Table: 6.2

Loss-Accuracy curive of train/ validation data of fold-3 in 5-fold cross validation during training is shown in Figure 6.4 and Figure 6.5.

Table 6.2: Lexicon Free-Result Analysis

Fold	k=2	k=3	k=5
Accuracy (%)	75.3	74.9	74.68

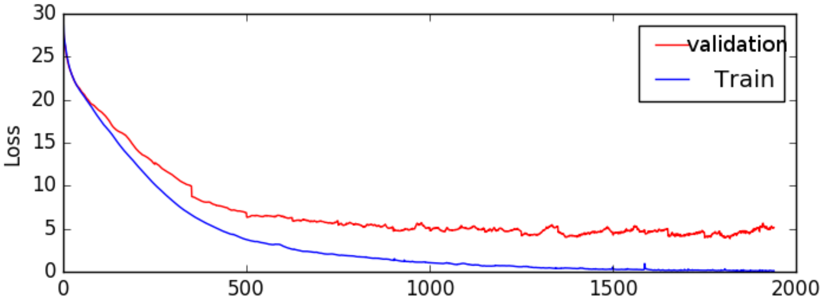


Figure 6.4: Loss Curve with Fold-3 data

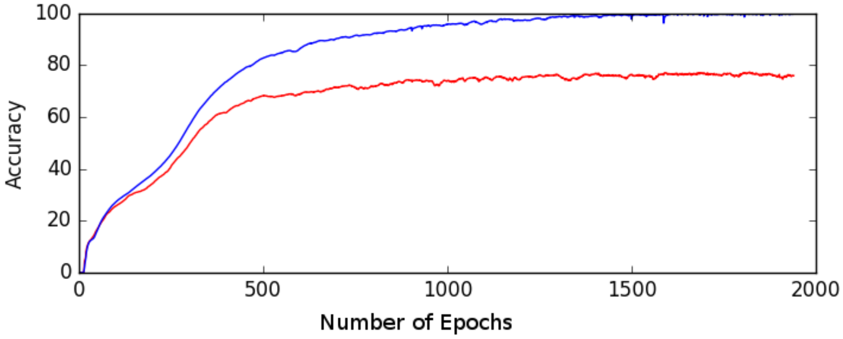


Figure 6.5: Accuracy Curve with Fold-3 data

With a dictionary consists of 330 words, the average accuracy achieved in word level is 91% using BK Tree search method[156]. Predicted word with CLC and dictionary with edit distance is shown in Table 6.3.

Edit distance is calculated using the predicted word with CLC and the matched word with dictionary. The first six rows of table 6.3 are wrong sequences or misclassified words, rest of the rows are classified perfectly.

Table 6.3: Predicted Words and corrected words using dictionary with Edit Distance

Actual Word	Predicted Word with CLC	Predicted with Dictionary	ED
ചിറയിൻകീഴ്	c i zha i zha eu	ചിറ്റാർ	4
നേമം	le ndha m	ഢം	2
ഉള്ളൂർ	o uu eR	ഒറ്റൂർ	1
ഉള്ളൂർ	k uu eR	കയ്യൂർ	1
കുളത്തൂർ	k eR mm uu eR	ഒറ്റൂർ	2
പെരുമ്പാവൂർ	t c npa va ar R uu eR	താനാളൂർ	4
മാരാരിക്കുളം	ma m ra ar ra i kk u La m	മാരാരിക്കുളം	1
മാരാരിക്കുളം	ma ra ar ra i kk u sha m	മാരാരിക്കുളം	1
മരിയാപുരം	ma ra i ar p u ra m	മരിയാപുരം	1
മരിയാപുരം	ma ra i ya ar cc u ra m	മരിയാപുരം	1
മാട്ടൂൽ	ma ar T ma uu l	മാട്ടൂൽ	2
വേളൂക്കര	le va ja kk ra	വേളൂക്കര	2
വാരപ്പെട്ടി	ma ar ra e pp pp T i	വാരപ്പെട്ടി	3
വാഴയൂർ	va ar T ya uu eR	വാഴയൂർ	1
അടൂർ	a ya uu eR	അടൂർ	2
വിജയപുരം	va i ja u nka p u ra m	വിജയപുരം	2
വിജയപുരം	va i ja va p u ra m	വിജയപുരം	1
വണ്ടൂർ	va nch uu eR	വണ്ടൂർ	1
വണ്ടൂർ	va nga uu eR	വണ്ടൂർ	1
വിതുര	va i t u t ra	വിതുര	1
ചേരാനല്ലൂർ	le p ra ar na ll uu eR	ചേരാനല്ലൂർ	1

Minimum edit distance used for the dictionary matching is 4. There are some samples that doesn't find any matching word from the dictionary, example: " e na yy ar RR i en k ra". The same word is matched with Edit Distance - 7 with actuals.

List of words with 100% accuracy is shown in Table 6.4.

Table 6.4: List of Words with 100% Accuracy

1	മേലടി	2	മൊകേരി	3	മുന്നിയൂർ
4	അരീക്കുളം	5	മൊറയൂർ	6	മുത്തോലി
7	മൈനാഗപ്പള്ളി	8	വടക്കൻ പറവൂർ	9	അരുവാപ്പുലം
10	പാലക്കുഴ	11	പള്ളിച്ചൽ	12	പാപ്പിനിശ്ശേരി
13	പട്ടാഴി വടക്കേക്കര	14	അതിരപ്പിള്ളി	15	പെരിങ്ങോട്ടുകുറിശ്ശി
16	പുറമേരി	17	പുത്തൻ വേലിക്കര	18	ആവോലി
19	തിരുമാറാടി	20	തിരുവാലി	21	തിരുവനന്തപുരം
22	തൊടിയൂർ	23	വടക്കേക്കാട്	24	വാണിയംകുളം
25	വാടാനപ്പള്ളി	26	വാടാനപ്പള്ളി	27	ചിറക്കാക്കോട്
28	കോഴിക്കോട്	29	ബാലുശ്ശേരി	30	ആദിച്ചനല്ലൂർ
31	ഏറ്റുമാനൂർ	32	ചടയമംഗലം	33	ചങ്ങരോത്ത്
34	ചാവക്കാട്	35	ചെല്ലാനം	36	പാഠം
37	ആലങ്ങാട്	38	സംഹാരമൂർത്തി	39	ചിറ്റാർ
40	ധർമ്മടം	41	എടക്കാട്	42	കടുത്തുരുത്തി
43	കല്ല്യാശ്ശേരി	44	ആലുവ	45	കണിയാമ്പറ്റ
46	കാസർകോട്	47	ആനക്കര	48	കിഴക്കമ്പലം
49	കൊടംതുരുത്ത്	50	കൊടുവള്ളി	51	കൊല്ലയിൽ
52	കൂട്ടിലങ്ങാടി	53	കോട്ടാങ്ങൽ	54	കൊയിലാണ്ടി
55	കുലുക്കല്ലൂർ	56	കുറ്റിക്കോൽ	57	ആനിക്കാട്
58	മടവൂർ	59	മടിക്കൈ	60	മംഗലം
61	മാണിക്കൽ				

6.4 Summary

In this chapter, a lexicon free recognition of the malayalam words are experimented and the results are promising. The motivation to do this experiment was to make the system generic viz. the beahviour is same for any input data. The accuracy reported is with the JMHRDB1. For a generic system instead of dictionary language model should be used. Sequence mapping with unicode and Grpaheme level representations are performed, the later provides the better result. The ambiguity in the decoding of sequences of each characters in the set { ‘ ◌െ ’, ‘◌ൈ’, ‘◌ൊ’ }, { ‘ ◌േ ’, ‘ ◌ോ ’ } and compound characters can clearly be resolved by using this mapping.

The implementation of dictionary as a post processing method reduces the CER and WER using the JMHRDB1.

Chapter 7

Prototye Form Processing System

I love taking an idea... to a prototype and then to a product that millions of people use.

Susan Wojcicki

7.1 Introduction

Documents can be either structured or unstructured. Recognition of structured documents are comaparatively easy compared with unstructured documents. Structured documents contain tables and forms to fill the data. Form is a document generally use to gather information, which contains one to one mapping between questions and fields to be filled manualy by different individuals. Forms are appear in various applications like school/ college admission form, Birth Certificate/ Live Certificate/ Death Certificate Application forms. DMOS a.k.a Description and MOdification of Segmentation [157][158] is a system developed for generic reognition of all type of documents. For invoice recogntion mainly for healthcare sector, smartFIX [159] sucsessfully implemented for form analysis. In this chapter we propose a knowledge based extraction and recognition method of information from the birth certificate application forms. Document recognition enables the fast retrieval of information from the documents and archiving it in an efficient and effective manner. It consists of two phases called 1) Analysis and 2) Understanding. In analysis phase deals with geometry or layout of the document. Understanding phase extracts the data and recognize using lexicon free/ specific methods.

7.2 Analysis

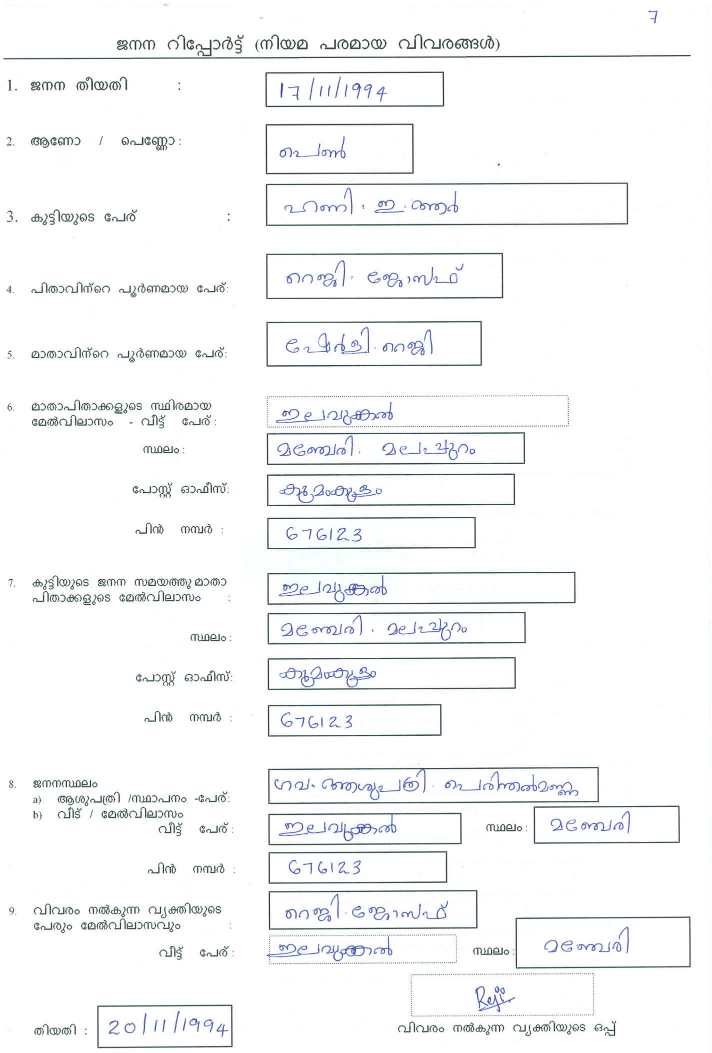
Mapping with the questions and answers are done on the basis of knowledge rule. Sample form is shown in Figure.7.1

The rectangles used to provide answers to the questions are not equal size. The variable size of rectagles helps to make it as template for mapping answer to the questions. The various template matching method [160] for finding the template in the referenced image or document is explained in the following sections.

7.2.1 Template Matching Methods

Sum of Squared Difference Matching

SSD(Sum of Squared Difference) is based on the pixel intensity. Template is placed over the image and move in a sliding window manner and perform the difference between the corresponding input image and template, then square and find the aggregate sum as shown in the equation(7.1). If the value is zero means exact match or if the value is very high means bad match. Close to zero consider for perfect matching.



7

ജനന റിപ്പോർട്ട് (നിയമ പരമായ വിവരങ്ങൾ)

1. ജനന തീയതി : 17/11/1994

2. ആണോ / പെണ്ണോ : പെൺ

3. കുട്ടിയുടെ പേര് : ഹണി. ഇ. ഞ്ഞർ

4. പിതാവിന്റെ പൂർണമായ പേര്: റെജി. ജോസഫ്

5. മാതാവിന്റെ പൂർണമായ പേര്: ഷേർളി. റെജി

6. മാതാപിതാക്കളുടെ സ്ഥിരമായ മേൽവിലാസം - വീട്ട് പേര് : ഇലവുക്കൽ

സ്ഥലം : മഞ്ചേരി. മലപ്പുറം

പോസ്റ്റ് ഓഫീസ്: കൂമങ്കുളം

പിൻ നമ്പർ : 676123

7. കുട്ടിയുടെ ജനന സമയത്തു മാതാ പിതാക്കളുടെ മേൽവിലാസം : ഇലവുക്കൽ

സ്ഥലം : മഞ്ചേരി . മലപ്പുറം

പോസ്റ്റ് ഓഫീസ്: കൂമങ്കുളം

പിൻ നമ്പർ : 676123

8. ജനനസ്ഥലം
a) ആശുപത്രി /സ്ഥാപനം -പേര്: ഗവ. ആശുപത്രി. പെരിന്തൽമണ്ണ
b) വീട് / മേൽവിലാസം
വീട്ട് പേര് : ഇലവുക്കൽ സ്ഥലം : മഞ്ചേരി

പിൻ നമ്പർ : 676123

9. വിവരം നൽകുന്ന വ്യക്തിയുടെ പേരും മേൽവിലാസവും : റെജി. ജോസഫ്

വീട്ട് പേര് : ഇലവുക്കൽ സ്ഥലം : മഞ്ചേരി

Reji

വിവരം നൽകുന്ന വ്യക്തിയുടെ ഒപ്പ്

തിയതി : 20/11/1994

Figure 7.1: Modified Birth Certificate Form

$$SSD(p,q) = \sum_{p',q'} [T(p',q') - I(p+p', q+q')]^2 \qquad (7.1)$$

where $T(p', q')$ is the intesity value in the template at position p'andq'.I stands for the document, $p+p'$ and $q+q'$ are the parameters for slide across the image.

Correlation Matching Method

The difference operation replace with multiplication in SSD is the only modification to achieve CMM (Correlation Matching Method) as shown in equation (7.2)

$$CMM(p,q) = \sum_{p',q'} [T(p',q') \cdot I(p+p',q+q')]^2 \tag{7.2}$$

Interpretation of the results are also just opposite to SSD. ie.In CMM if the result is close to zero or zero it is bad match and higher value leads to perfect match.

Correlation coefficient matching methods

CCMM(Correlation Coefficient Matching Method) follows same basic frame work of CMM.Instead of pixel intensity value, it took the mean value of the template relative to the mean value of the reference image or document. Mathematical representation of CCMM is shown in equation 7.3.

$$CCMM(p,q) = \sum_{p',q'} [T'(p',q') \cdot I'(p+p',q+q')]^2 \tag{7.3}$$

$$\text{where } T'(p',q') = T(p',q') - \frac{1}{(w \cdot h) \sum_{p^{"},q^{"}} T(p^{"},q^{"}}} \tag{7.4}$$

and

$$I'(p+p',q+q') = I(p+p',q+q') - \frac{1}{(w \cdot h) \sum_{p^{"},q^{"}} I(p+p^{"},q+q^{"})} \tag{7.5}$$

Normalized Methods

To reduce the lighting effects between the template and image, Normalized versions of the any of the methods discussed previously can use. Normalization co-efficient,$Z(p, q)$ is expressed in equation (7.6)

$$Z(p,q) = \sqrt{\sum_{p',q'} T(p',q')^2 \cdot \sum_{p',q'} I(p+p',q+q')^2} \tag{7.6}$$

Normalized sum of squared differences,corelation, correlation co-efficient can be mathematically expressed by equation (7.7),equation (7.8),equation (7.9) respectively.

$$SSD_{norm}(p,q) = \frac{SSD(p,q)}{Z(p,q)} \tag{7.7}$$

$$CMM_{norm}(p,q) = \frac{CMM(p,q)}{Z(p,q)} \tag{7.8}$$

$$CCMM_{norm}(p,q) = \frac{CCMM(p,q)}{Z(p,q)} \tag{7.9}$$

The comparison with different template matching methods are shown in Table 7.1.Correlation co-efficient matching method produced 98.81 % accuracy with 50 forms.

Table 7.1: Comaparison of Accuracy with template matching methods

Method	Accuracy
SSD	27.27
CMM	9.09
CCMM	98.81
SSD_norm	25
CMM_norm	22.72
CCMM_norm	68.18

7.2.2 Document Form Processing

A block diagram giving the flow of actions for extraction of contents from form document image is shown in Figure 7.2. To find the proper offset values of the blocks to extract, template matching method is used. All methods explained in Section 7.2.1 experimented with the dataset, JMHRDB2. Correlation coefficient matching method provide better result compared to all other methods. Extracted blocks with borders are shown in Figure 7.3

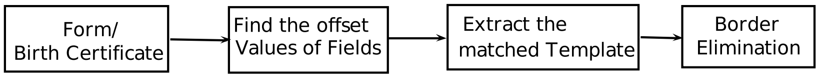


Figure 7.2: Block Daigram of Birth Certificate Form Processing

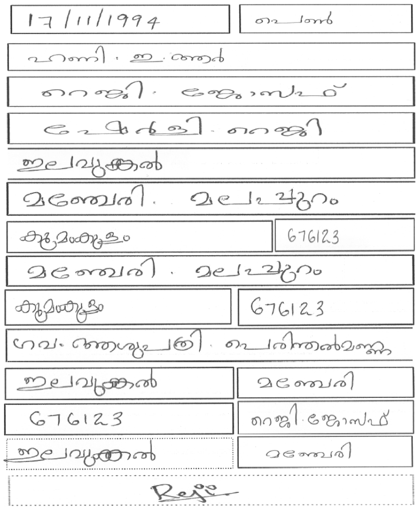

17/11/1994
റെജി. ജോസഫ്
ഇലവുങ്കൽ
മഞ്ചേരി. മലപ്പുറം
676123
മഞ്ചേരി. മലപ്പുറം
676123
ഇലവുങ്കൽ
മഞ്ചേരി
676123
റെജി. ജോസഫ്
ഇലവുങ്കൽ
മഞ്ചേരി

Figure 7.3: Extracted blocks from Birth Certificate Form

Borders are eliminated and remove the space from all the sides before feed to the recognizer. Knowledge mapping is represented in Figure 7.4

Tree representation of the document is shown in Figure 7.5, where internal root nodes represents questions and leaf nodes represents answers.

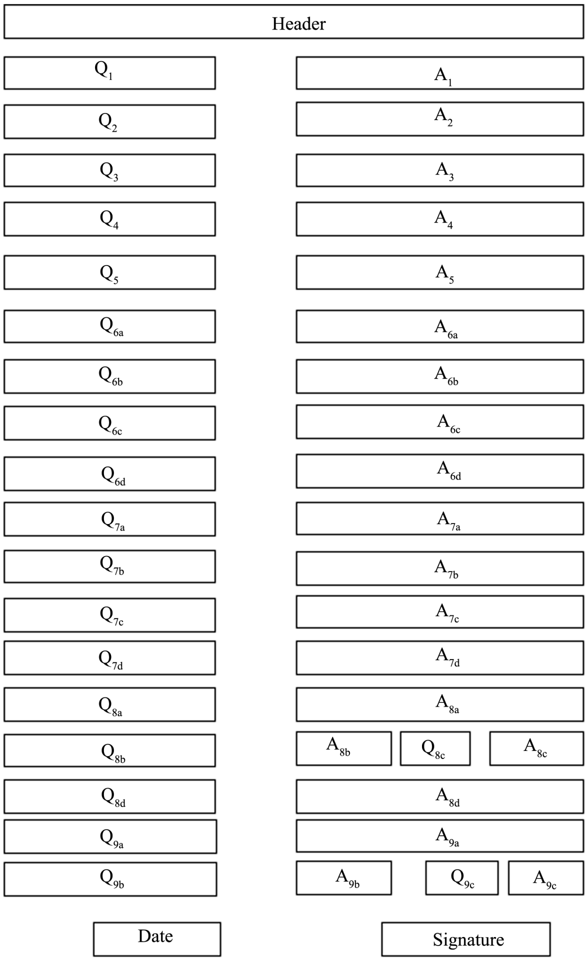


Figure 7.4: Knowledge Mapping of Question and Answers

Signature verification is not consider in our work, even the datset consist of it in a well separated manner.

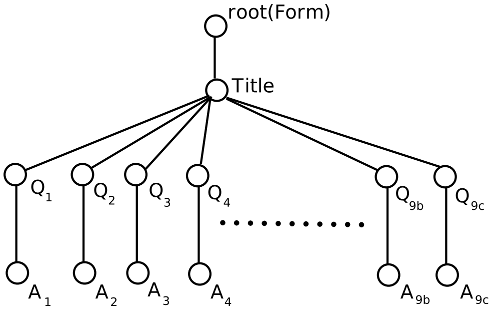


Figure 7.5: Tree Representaion of the Knowledge base-Q's represents questions and A's represents answers

7.3 Recognition

Prototype for the Birth certificate recognition is shown in Figure 7.6. The protoype consists of form data extraction and recognition module. The resesigned JMHRDB1 form data are fed to form data extraction module, where it use correlation co-efficient template matching method . 22 templates are provided for the extraction of ground truth and data, Borders are eliminated from the extracted block. CLC pretrained model dicussed in Chapter6 is used for the recognition purpose. The response to Q9.c provided the maximum accuracy of 79%.

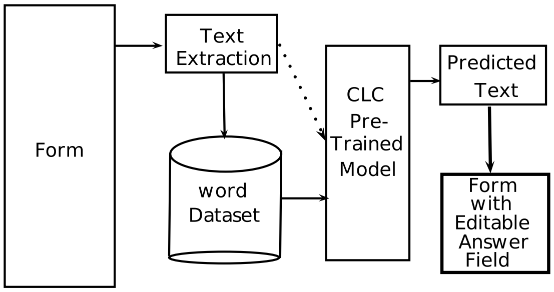


Figure 7.6: Birth Certificate Recognition- Prototype

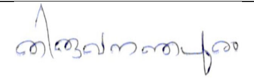
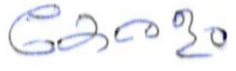
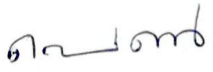
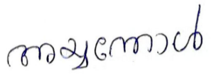
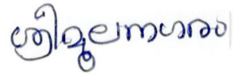
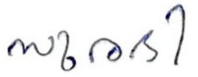
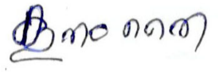
Dataset discussed in Chapter 3 is used for the experiments. Method used for recognition is discussed in Chapter 6. Accuracy is shown in Table 7.2 is question wise and the metric used is CER.

Table 7.2: Question-Wise Result Analysis

Question	Accuracy	Question	Accuracy
Q2	**77.2**	Q7.b	76.9
Q3	72	Q7.c	**77.1**
Q4	73.4	Q8.a	71
Q5	72.3	Q8.b	72.1
Q6.a	70.1	Q8.c	**78**
Q6.b	**78.8**	Q9.a	70.2
Q6.c	**77.9**	Q9.b	77.7
Q7.a	71.2	Q9.c	**79**

Some of the samples and their transcriptions by the model with their Edit distance and CER is furnished in Table 7.3.

Table 7.3: Word Images and transcriptions with ED (Edit Distance) and CER

Si No	Image	Recognized Word	ED	CER (%)
1		തിതവനന്തപുരം	2	18.18
2		കേരളം	0	0
3		പെണ	1	33.33
4		അയന്തോൾ	1	16.67
5		ശീമൂലനഗരം	1	10
6		സുരദി	1	20
7		ക്കനംതൈ	2	28.57

7.4 Summary

Birth Certificate form recognition is one of the application of offline handwriting recognition. As a future work identification of the blocks or fields can consider as a machine learning problem. Performance of our prototype form processing system may be improved by using a language model. Digit recognition sub-module of the form processing system is trained using handwritten samples of MNIST dataset. This prototype model can extend for any applications with the modifications of form structure.

Chapter 8

Conclusions

A conclusion is the place where you get tired of thinking.

Arthur Bloch

This thesis decribes the new techniques for offline handwriting recognition of Malayalam strings. Word recognition using holistic and analytic approach are discussed. Holistic approach are suitable for limited lexicon size applications like Town/ Village/ Corporation/ Panchayath name recognition etc. For a generic recognition the approach can be analytic or hybrid. Sequence wise analysis and labelling is a better option for generic recognition and the system is scalable in this case. Several subproblems or challenges for the transcription of handwritten image are addressed in this thesis. The major challenge for the recogntion of Malayalam or any Indian language is to devise a proper method for segmentation or follows segmentation free approach.

As part of this research work we developed a dataset called JMHRDB, suitable for handwritten Malayalam word and document recognition. It is available in hdf5 format with groud truth of each individual file. Newly adapted methods for offline Malayalam handwriting recognition is shown in Figure 8.1

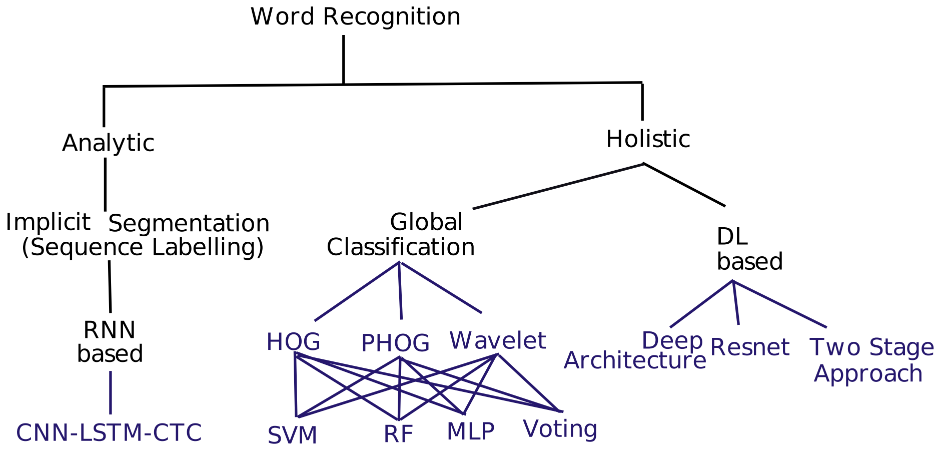


Figure 8.1: Newly adapted methods for offline Malayalam Handwriting Recognition shown in blue colour

Implementation of deep architectures shows that they are suitable for holistic/ analytic recognition. We used two state-of-the-art methods to design/ implement the architecture. These methods are based on Convolutional and Recurrent Neural Network. Traditional or classical machine learning methods also implemented and compared with some existing Hindi and Marathi Dataset. Both Traditional and Deep methods shows that it is suitable for script independent recognition. Prototype for birth certificate recognition system is also proposed, which can be extended to many other forms.

Various applications of handwriting recognition are answer paper evaluation, reading notes, bank cheque processing, address interpretation from the postal mails, application form/invoice processing, signature verification, writer identification, gender classification, keyword spotting and medical applications viz. to identify the progress of paralysed patients after the treatment.

Some of the future works are, in practical applications like postal mail, lecturer notes etc., it is common that different scripts are used in the same pages or documents. Script identification and recognition is an obvious solution for it. Another work that can be done along with this is writer identification, it will be useful for various forensic and demographic investigations. Malayalam handwritten keyword spotting is another area to be focussed, which will enhance the abilities to reteieve the right information quickly. Enhanced Label embedding closely related to PHOC(Pyramidal Histogram Of Characters) for Indian languages is also another future area of research. The recognition of multi word, line, Paragraph, page are another area of research to be focus.

Bibliography

[1] Ji Gan and Weiqiang Wang. In-air handwritten english word recognition using attention recurrent translator. *Neural Computing and Applications*, pages 1–18, 2017.

[2] N. Arica and F. T. Yarman-Vural. An overview of character recognition focused on off-line handwriting. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 31(2):216–233, 2001.

[3] R. Plamondon and S. N. Srihari. On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(1):63–84, 2000.

[4] Øivind Trier, Anil K. Jain, and Torfinn Taxt. Feature extraction methods for character recognition - a survey. *Pattern Recognition*, 29:641–662, 1996.

[5] U. Bhattacharya and B. B. Chaudhuri. Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):444–457, 2009.

[6] U. Bhattacharya, M. Shridhar, S. K. Parui, P. K. Sen, and B. B. Chaudhuri. Offline recognition of handwritten Bangla characters:

an efficient two-stage approach. *Pattern Analysis and Applications*, 15(4):445–458, 2012.

[7] U. Bhattacharya, S. K. Ghosh, and S. Parui. A two stage recognition scheme for handwritten Tamil characters. In *9th International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 1, pages 511–515, 2007.

[8] T. K. Bhowmik, S. K. Parui, U. Bhattacharya, and B. Shaw. An HMM based recognition scheme for handwritten Oriya numerals. In *9th International Conference on Information Technology (ICIT '06)*, pages 105–110, 2006.

[9] J. John, K. V. Pramod, K. Balakrishnan, and B. B. Chaudhuri. A two stage approach for handwritten Malayalam character recognition. In *14th International Conference on Frontiers in Handwriting Recognition*, pages 199–204, 2014.

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

[11] B. Shaw, U. Bhattacharya, and S. K. Parui. Offline handwritten Devanagari word recognition: Information fusion at feature and classifier levels. In *3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 720–724, 2015.

[12] B. Shaw, U. Bhattacharya, and S. K. Parui. Combination of features for efficient recognition of offline handwritten Devanagari words. In *14th International Conference on Frontiers in Handwriting Recognition*, pages 240–245, 2014.

[13] U. Pal, K. Roy, and F. Kimura. A lexicon driven method for unconstrained Bangla handwritten word recognition. In *10th Int. Workshop on Frontiers in Handwriting Recognition*, 2006.

[14] Bin Zhang Sargur N Srihari. Analysis of handwriting individuality using word features. *State University of New York at Buffalo, Buffalo, NY*, 14228, 2003.

[15] Mohamed Cheriet, Nawwaf Kharma, Cheng-Lin Liu, and Ching Suen. *Character recognition systems: a guide for students and practitioners.* John Wiley & Sons, 2007.

[16] Alessandro L Koerich, Robert Sabourin, and Ching Y Suen. Large vocabulary off-line handwriting recognition: A survey. *Pattern Analysis & Applications*, 6(2):97–121, 2003.

[17] Alex Graves. Supervised sequence labelling with recurrent neural networks. 2012. *ISBN 9783642212703.*

[18] https://www.ethnologue.com/language/. [Online; accessed 4-January-2018].

[19] `http://unicode.org/L2/L2008/08039-kerala-order.pdf`. [Online; accessed 22-September-2017].

[20] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):855–868, 2009.

[21] Suen Wang. *Character And Handwriting Recognition:Exapanding Frontiers*, volume 30. World Sceintific Series In Computer Science, 1991.

[22] Sebastiano Impedovo. *Fundamentals in handwriting recognition*, volume 124. Springer Science & Business Media, 2012.

[23] Horst Bunke, Markus Roth, and Ernst Günter Schukat-Talamazzini. Off-line cursive handwriting recognition using hidden markov models. *Pattern recognition*, 28(9):1399–1413, 1995.

[24] Christian Viard-Gaudin, Pierre Michel Lallican, Stefan Knerr, and Philippe Binter. The ireste on/off (ironoff) dual handwriting database. In *Document Analysis and Recognition, 1999. ICDAR'99. Proceedings of the Fifth International Conference on*, pages 455–458. IEEE, 1999.

[25] Yousri Kessentini, Thierry Paquet, and AbdelMajid Ben Hamadou. Off-line handwritten word recognition using multi-stream hidden markov models. *Pattern Recognition Letters*, 31(1):60–70, 2010.

[26] Mario Pechwitz, S Snoussi Maddouri, Volker Märgner, Noureddine Ellouze, Hamid Amiri, et al. Ifn/enit-database of handwritten arabic words. In *Proc. of CIFED*, volume 2, pages 127–136, 2002.

[27] Sabri A Mahmoud, Irfan Ahmad, Wasfi G Al-Khatib, Mohammad Alshayeb, Mohammad Tanvir Parvez, Volker Märgner, and Gernot A Fink. Khatt: An open arabic offline handwritten text database. *Pattern Recognition*, 47(3):1096–1112, 2014.

[28] `https://catalog.ldc.upenn.edu/byproject/`. [Online; accessed 20-April-2018].

[29] R Wilkinson, J Geist, S Janet, P Grother, C Burges, R Creecy, B Hammond, J Hull, N Larsen, T Vogl, et al. The first census optical character recognition systems. *The US Bureau of Census and the National Institute of Standards and Technology (Tech. Rep. NISTIR 4912, National Institute of Standards and Technology.). Gaithersburg, MD*, 1992.

[30] `https://www.nist.gov/srd/nist-special-database-19`. [Online; accessed 13-January-2018].

[31] Ching Y Suen, Christine Nadal, Raymond Legault, Tuan A Mai, and Louisa Lam. Computer recognition of unconstrained handwritten numerals. *Proceedings of the IEEE*, 80(7):1162–1180, 1992.

[32] Malik Waqas Sagheer, Chun Lei He, Nicola Nobile, and Ching Y Suen. A new large urdu database for off-line handwriting recognition. In *International Conference on Image Analysis and Processing*, pages 538–546. Springer, 2009.

[33] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.

[34] `http://yann.lecun.com/exdb/mnist/`. [Online; accessed 15-January-2018].

[35] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012.

[36] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters. *arXiv preprint arXiv:1702.05373*, 2017.

[37] Mohamed E Hussein, Marwan Torki, Ahmed Elsallamy, and Mahmoud Fayyaz. Alexu-word: A new dataset for isolated-word closed-vocabulary offline arabic handwriting recognition. *arXiv preprint arXiv:1411.4670*, 2014.

[38] Farès Menasri, Jérôme Louradour, Anne-Laure Bianne-Bernard, and Christopher Kermorvant. The a2ia french handwriting recognition system at the rimes-icdar2011 competition. In *Document Recognition and Retrieval XIX*, volume 8297, page 82970Y. International Society for Optics and Photonics, 2012.

[39] Cheng-Lin Liu, Fei Yin, Da-Han Wang, and Qiu-Feng Wang. Casia online and offline chinese handwriting databases. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 37–41. IEEE, 2011.

[40] `http://memory.loc.gov/ammem/gwhtml/gwseries2.html`. [Online; accessed 8-May-2018].

[41] Michael Stauffer, Andreas Fischer, and Kaspar Riesen. A novel graph database for handwritten word images. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 553–563. Springer, 2016.

[42] Ujjwal Bhattacharya and BB Chaudhuri. Databases for research on recognition of handwritten characters of indian scripts. In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pages 789–793. IEEE, 2005.

[43] BB Chaudhuri. A complete handwritten numeral database of bangla–a major indic script. In *Tenth International Workshop on Frontiers in Handwriting Recognition*. Suvisoft, 2006.

[44] Bikash Shaw, Swapan Kr Parui, and Malayappan Shridhar. Offline handwritten devanagari word recognition: A segmentation based approach. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[45] R Jayadevan, Satish R Kolhe, Pradeep M Patil, and Umapada Pal. Database development and recognition of handwritten devanagari legal amount words. In *2011 International Conference on Document Analysis and Recognition*, pages 304–308. IEEE, 2011.

[46] Faizal Hajamohideen and S Noushath. Kalanjiyam: Unconstrained offline tamil handwritten database. In *International Conference on Computer Vision, Graphics, and Image processing*, pages 277–287. Springer, 2016.

[47] Showmik Bhowmik, Samir Malakar, Ram Sarkar, Subhadip Basu, Mahantapas Kundu, and Mita Nasipuri. Off-line bangla handwritten word recognition: a holistic approach. *Neural Computing and Applications*, pages 1–16, 2018.

[48] `http://www.iitr.ac.in/media/facspace/proy.fcs/IndicWord.rar`. [Online; accessed 2-May-2018].

[49] Chandranath Adak, Bidyut B Chaudhuri, and Michael Blumenstein. Offline cursive bengali word recognition using cnns with a recurrent model. In *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*, pages 429–434. IEEE, 2016.

[50] Prakash Choudhary and Neeta Nain. A four-tier annotated urdu handwritten text image dataset for multidisciplinary research on urdu script. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 15(4):26, 2016.

[51] Saad Bin Ahmed, Saeeda Naz, Salahuddin Swati, Muhammad Imran Razzak, Arif Iqbal Umar, and Akbar Ali Khan. Ucom offline dataset-an urdu handwritten dataset generation. *Int. Arab J. Inf. Technol.*, 14(2):239–245, 2017.

[52] Alireza Alaei, P Nagabhushan, and Umapada Pal. A benchmark kannada handwritten document dataset and its segmentation. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 141–145. IEEE, 2011.

[53] Akram Khémiri, Afef Kacem Echi, Abdel Belaïd, and Mourad Elloumi. Arabic handwritten words off-line recognition based on hmms and dbns. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 51–55. IEEE, 2015.

[54] Akram Khémiri, Afef Kacem Echi, Abdel Belaïd, and Mourad Elloumi. A system for off-line arabic handwritten word recognition

based on bayesian approach. In *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*, pages 560–565. IEEE, 2016.

[55] Jawad H AlKhateeb, Olivier Pauplin, Jinchang Ren, and Jianmin Jiang. Performance of hidden markov model and dynamic bayesian network classifiers on handwritten arabic word recognition. *knowledge-based systems*, 24(5):680–688, 2011.

[56] Ali Broumandnia, Jamshid Shanbehzadeh, and M Rezakhah Varnoosfaderani. Persian/arabic handwritten word recognition using m-band packet wavelet transform. *Image and Vision Computing*, 26(6):829–842, 2008.

[57] Arik Poznanski and Lior Wolf. Cnn-n-gram for handwriting word recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2305–2314, 2016.

[58] Théodore Bluche, Hermann Ney, and Christopher Kermorvant. A comparison of sequence-trained deep neural networks and recurrent neural networks optical modeling for handwriting recognition. In *International Conference on Statistical Language and Speech Processing*, pages 199–210. Springer, 2014.

[59] Théodore Bluche, Hermann Ney, and Christopher Kermorvant. Tandem hmm with convolutional neural network for handwritten word recognition. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2390–2394. IEEE, 2013.

[60] Patrick Doetsch, Mahdi Hamdani, Hermann Ney, Adrià Giménez, Jesús Andrés-Ferrer, and Alfons Juan. Comparison of bernoulli and gaussian hmms using a vertical repositioning technique for off-line handwriting recognition. In *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*, pages 3–7. IEEE, 2012.

[61] Michal Kozielski, Jens Forster, and Hermann Ney. Moment-based image normalization for handwritten text recognition. In *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*, pages 256–261. IEEE, 2012.

[62] Ronaldo Messina and Jerome Louradour. Segmentation-free handwritten chinese text recognition with lstm-rnn. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 171–175. IEEE, 2015.

[63] Song Wang, Li Chen, Liang Xu, Wei Fan, Jun Sun, and Satoshi Naoi. Deep knowledge training and heterogeneous cnn for handwritten chinese text recognition. In *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*, pages 84–89. IEEE, 2016.

[64] Dewi Suryani, Patrick Doetsch, and Hermann Ney. On the benefits of convolutional neural network combinations in offline handwriting recognition. In *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*, pages 193–198. IEEE, 2016.

[65] Yi-Chao Wu, Fei Yin, Zhuo Chen, and Cheng-Lin Liu. Handwritten chinese text recognition using separable multi-dimensional recurrent neural network. In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, volume 1, pages 79–84. IEEE, 2017.

[66] Umapada Pal, Kaushik Roy, and Fumitaka Kimura. A lexicon-driven handwritten city-name recognition scheme for indian postal automation. *IEICE transactions on information and systems*, 92(5):1146–1158, 2009.

[67] Umapada Pal, Ramit Kumar Roy, and Fumitaka Kimura. Multilingual city name recognition for indian postal automation. In *Fron-*

tiers in Handwriting Recognition (ICFHR), 2012 International Conference on, pages 169–173. IEEE, 2012.

[68] S Thadchanamoorthy, ND Kodikara, HL Premaretne, Umapada Pal, and Fumitaka Kimura. Tamil handwritten city name database development and recognition for postal automation. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 793–797. IEEE, 2013.

[69] Subhadip Basu, Nibaran Das, Ram Sarkar, Mahantapas Kundu, Mita Nasipuri, and Dipak Kumar Basu. A hierarchical approach to recognition of handwritten bangla characters. *Pattern Recognition*, 42(7):1467–1484, 2009.

[70] Thadchanamoorthy Subramaniam, Umapada Pal, H Premaretne, and N Kodikara. Holistic recognition of handwritten tamil words. In *Emerging Applications of Information Technology (EAIT), 2012 Third International Conference on*, pages 165–169. IEEE, 2012.

[71] Supratim Das, Pawan Kumar Singh, Showmik Bhowmik, Ram Sarkar, and Mita Nasipuri. A harmony search based wrapper feature selection method for holistic bangla word recognition. *Procedia Computer Science*, 89:395–403, 2016.

[72] Malik Waqas Sagheer, Chun Lei He, Nicola Nobile, and Ching Y Suen. Holistic urdu handwritten word recognition using support vector machine. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 1900–1903. IEEE, 2010.

[73] Samir Malakar, Pankaj Sharma, Pankaj Kumar Singh, Maitrayee Das, Ram Sarkar, and Mita Nasipuri. A holistic approach for handwritten hindi word recognition. *International Journal of Computer Vision and Image Processing (IJCVIP)*, 7(1):59–78, 2017.

[74] Omar Mukhtar, Srirangaraj Setlur, and Venu Govindaraju. Experiments on urdu text recognition. In *Guide to OCR for Indic Scripts*, pages 163–171. Springer, 2009.

[75] Bikash Shaw, Ujjwal Bhattacharya, and Swapan K Parui. Combination of features for efficient recognition of offline handwritten devanagari words. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 240–245. IEEE, 2014.

[76] S Karthik and MK Srikanta. Segmentation and recognition of handwritten kannada text using relevance feedback and histogram of oriented gradients: a novel approach. *Int J Adv Comput Sci Appl*, 7(1):472–476, 2016.

[77] Parita R Paneri, Ronit Narang, and Mukesh M Goswami. Offline handwritten gujarati word recognition. In *Image Information Processing (ICIIP), 2017 Fourth International Conference on*, pages 1–5. IEEE, 2017.

[78] IIIT CVIT. Towards accurate handwritten word recognition for hindi and bangla. In *Computer Vision, Pattern Recognition, Image Processing, and Graphics: 6th National Conference, NCVPRIPG 2017, Mandi, India, December 16-19, 2017, Revised Selected Papers*, volume 841, page 470. Springer, 2018.

[79] Suman Bhoi, DP Dogra, and PP Roy. Handwritten text recognition in odia script using hidden markov model. In *Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2015 Fifth National Conference on*, pages 1–3. IEEE, 2015.

[80] Partha Pratim Roy, Ayan Kumar Bhunia, Ayan Das, Prasenjit Dey, and Umapada Pal. Hmm-based indic handwritten word recognition using zone segmentation. *Pattern Recognition*, 60:1057–1075, 2016.

[81] Szilárd Vajda, Kaushik Roy, Umapada Pal, Bidyut Baran Chaudhuri, and Abdel Belaid. Automation of indian postal documents written in bangla and english. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(08):1599–1632, 2009.

[82] Partha Pratim Roy, Umapada Pal, and Josep Lladós. Text line extraction in graphical documents using background and foreground information. *International Journal on Document Analysis and Recognition (IJDAR)*, 15(3):227–241, 2012.

[83] Umapada Pal and Partha Pratim Roy. Multioriented and curved text lines extraction from indian documents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(4):1676–1684, 2004.

[84] Tapan K Bhowmik, Swapan K Parui, and Utpal Roy. Discriminative hmm training with ga for handwritten word recognition. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[85] Swapan Kumar Parui and Bikash Shaw. Offline handwritten devanagari word recognition: An hmm based approach. In *International Conference on Pattern Recognition and Machine Intelligence*, pages 528–535. Springer, 2007.

[86] Sitaram Ramachandrula, Shrang Jain, and Hariharan Ravishankar. Offline handwritten word recognition in hindi. In *Proceeding of the workshop on Document Analysis and Recognition*, pages 49–54. ACM, 2012.

[87] Jomy John, KV Pramod, and Kannan Balakrishnan. Unconstrained handwritten malayalam character recognition using wavelet transform and support vector machine classifier. *Procedia Engineering*, 30:598–605, 2012.

[88] G Raju, Bindu S Moni, and Madhu S Nair. A novel handwritten character recognition system using gradient based features and run length count. *Sadhana*, 39(6):1333–1355, 2014.

[89] Binu P Chacko. *Intelligent character recognition: a study and analysis of extreme learning machine and support vector machine using division point and wavelet features.* PhD thesis, 2011.

[90] VL Lajish. Handwritten character recognition using gray-scale based state-space parameters and class modular nn. In *Signal Processing, Communications and Networking, 2008. ICSCN'08. International Conference on*, pages 374–379. IEEE, 2008.

[91] `https://github.com/kaldi-asr/kaldi`. [Online; accessed 20-April-2018].

[92] `http://www.openfst.org/twiki/bin/view/FST/WebHome`. [Online; accessed 22-April-2018].

[93] `http://www.speech.sri.com/projects/srilm/`. [Online; accessed 22-April-2018].

[94] `https://github.com/srvk/eesen/`. [Online; accessed 22-April-2018].

[95] Ismet Zeki Yalniz and Raghavan Manmatha. A fast alignment scheme for automatic ocr evaluation of books. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 754–758. IEEE, 2011.

[96] `http://ciir.cs.umass.edu/downloads/ocr-evaluation/`. [Online; accessed 28-April-2018].

[97] `http://htk.eng.cam.ac.uk/`. [Online; accessed 22-April-2018].

[98] `https://sourceforge.net/projects/rnnl/`. [Online; accessed 23-April-2018].

[99] Parvathy Prasad .S and Rose Mary .A. Emergence of malayalam as an independent classical language-an overview. *International Journal of Science and Research (IJSR)*, 5(8):854–856, 2016.

[100] NV Neeba, Anoop Namboodiri, CV Jawahar, and PJ Narayanan. Recognition of malayalam documents. In *Guide to OCR for Indic Scripts*, pages 125–146. Springer, 2009.

[101] A. R. Raja Raja Varma. *Kerala Panineeyam*. Kozhikode Poorna, 3rd edition, 1997.

[102] NV Neeba. *Large Scale Character Classification*. PhD thesis, International Institute of Information Technology Hyderabad, India, 2010.

[103] S Prema and Manu Joseph. Malayalam frequency count study report. *Technical Report, Department of Linguistics, University of Kerala*, 2001.

[104] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[105] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[106] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[107] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, et al. Going deeper with convolutions. Cvpr, 2015.

[108] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[109] L. Kang, J. Kumar, P. Ye, Y. Li, and D. Doermann. Convolutional neural networks for document image classification. In *22nd International Conference on Pattern Recognition*, 2014.

[110] Durjoy Sen Maitra, Ujjwal Bhattacharya, and Swapan K Parui. Cnn based common approach to handwritten character recognition of multiple scripts. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 1021–1025. IEEE, 2015.

[111] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[112] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[113] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2010.

[114] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, 2(2):121–167, 1998.

[115] Rafael C Gonzalez and Richard E Woods. *Digital Image Processing.* Prentice Hall Press, 2008.

[116] Christopher M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

[117] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

[118] Léon Bottou. *Stochastic Gradient Descent Tricks*, pages 421–436. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[119] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[120] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. 2012.

[121] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[122] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification. 2003.

[123] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[124] Mohammad Sadegh Ebrahimi and Hossein Karkeh Abadi. Study of residual networks for image recognition. *arXiv preprint arXiv:1805.00325*, 2018.

[125] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456, 2015.

[126] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[127] Loris Nanni, Stefano Ghidoni, and Sheryl Brahnam. Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern Recognition*, 71:158–172, 2017.

[128] Sneha Singh, Tharun Kariveda, Jija Das Gupta, and Kallol Bhattacharya. Handwritten words recognition for legal amounts of bank cheques in english script. In *Advances in Pattern Recognition (ICAPR), 2015 Eighth International Conference on*, pages 1–5. IEEE, 2015.

[129] John Jomy and KV Pramod. *Pattern Analysis Techniques for the Recognition of Unconstrained Handwritten Malayalam Character Images*. PhD thesis, Cochin University of Science And Technology, 2014.

[130] David H Wolpert and William G Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.

[131] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[132] Chang Shu, Xiaoqing Ding, and Chi Fang. Histogram of the oriented gradient for face recognition. *Tsinghua Science & Technology*, 16(2):216–224, 2011.

[133] Yang Bai, Lihua Guo, Lianwen Jin, and Qinghua Huang. A novel feature extraction method using pyramid histogram of orientation gradients for smile recognition. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 3305–3308. IEEE, 2009.

[134] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 401–408. ACM, 2007.

[135] Stephane G Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 1989.

[136] Ingrid Daubechies. *Ten lectures on wavelets*. SIAM, 1992.

[137] Wolfgang Karl Härdle, KF Phoon, and David Kuo Chuen Lee. Credit rating score analysis. In *Applied Quantitative Finance*, pages 223–244. Springer, 2017.

[138] Dhanya Jothimani, Ravi Shankar, and Surendra S Yadav. Portfolio selection in indian stock market using relative performance indicator approach. In *Flexibility in Resource Management*, pages 185–201. Springer, 2018.

[139] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2014.

[140] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[141] Jonathan Milgram, Mohamed Cheriet, and Robert Sabourin. one against one or one against all: Which one is better for handwriting recognition with svms? In *tenth international workshop on Frontiers in handwriting recognition*. SuviSoft, 2006.

[142] Mariana Belgiu and Lucian Drăguţ. Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114:24–31, 2016.

[143] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Image classification using random forests and ferns. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[144] Somaya Al Maadeed and Abdelaali Hassaine. Automatic prediction of age, gender, and nationality in offline handwriting. *EURASIP Journal on Image and Video Processing*, 2014(1):10, 2014.

[145] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[146] Kenneth M Sayre. Machine recognition of handwritten words: A project report. *Pattern recognition*, 5(3):213–228, 1973.

[147] Bappaditya Chakraborty, Partha Sarathi Mukherjee, and Ujjwal Bhattacharya. Bangla online handwriting recognition using recurrent neural network architecture. In *Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing*, page 63. ACM, 2016.

[148] Partha Sarathi Mukherjee, Bappaditya Chakraborty, Ujjwal Bhattacharya, and Swapan Kumar Parui. A hybrid model for end to end online handwriting recognition. In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, volume 1, pages 658–663. IEEE, 2017.

[149] Partha Sarathi Mukherjee. `https://github.com/xisnu/CNN-BLSTM-CTC`. [Online; accessed 26-December-2017].

[150] Alex Graves. Supervised sequence labelling. In *Supervised Sequence Labelling with Recurrent Neural Networks*, pages 5–13. Springer, 2012.

[151] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *ICML*, volume 14, pages 1764–1772, 2014.

[152] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318, 2013.

[153] Alex Graves, Marcus Liwicki, Horst Bunke, Jürgen Schmidhuber, and Santiago Fernández. Unconstrained on-line handwriting recognition with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 577–584, 2008.

[154] Utpal Garain, Luc Mioulet, Bidyut B Chaudhuri, Clement Chatelain, and Thierry Paquet. Unconstrained bengali handwriting recognition with recurrent models. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 1056–1060. IEEE, 2015.

[155] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.

[156] Walter A. Burkhard and Robert M. Keller. Some approaches to best-match file searching. *Communications of the ACM*, 16(4):230–236, 1973.

[157] Bertrand Coüasnon and Aurélie Lemaitre. Dmos, it's your turn! In *1st International Workshop on Open Services and Tools for Document Analysis (ICDAR-OST)*, 2017.

[158] Bertrand Coüasnon. Dmos, a generic document recognition method: Application to table structure analysis in a general and in a specific way. *International Journal of Document Analysis and Recognition (IJDAR)*, 8(2-3):111–122, 2006.

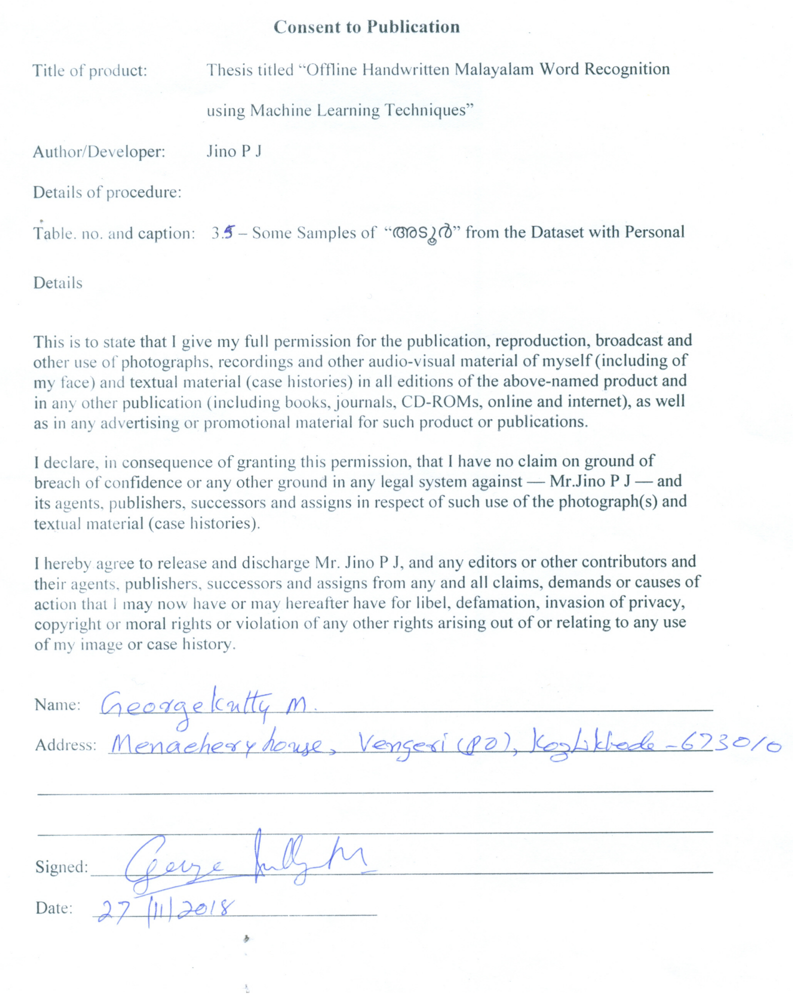
[159] Bertin Klein, Andreas R Dengel, and Andreas Fordan. smartfix: An adaptive system for document analysis and understanding. In *Reading and Learning*, pages 166–186. Springer, 2004.

[160] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library.* " O'Reilly Media, Inc.", 2008.

Appendix A

consent forms

Sample consent forms are shown in Figure: A.1, Figure: A.2, Figure: A.3 and Figure: A.4.



Consent to Publication

Title of product: Thesis titled "Offline Handwritten Malayalam Word Recognition using Machine Learning Techniques"

Author/Developer: Jino P J

Details of procedure:

Table. no. and caption: 3.5 – Some Samples of "അടൂർ" from the Dataset with Personal Details

This is to state that I give my full permission for the publication, reproduction, broadcast and other use of photographs, recordings and other audio-visual material of myself (including of my face) and textual material (case histories) in all editions of the above-named product and in any other publication (including books, journals, CD-ROMs, online and internet), as well as in any advertising or promotional material for such product or publications.

I declare, in consequence of granting this permission, that I have no claim on ground of breach of confidence or any other ground in any legal system against — Mr.Jino P J — and its agents, publishers, successors and assigns in respect of such use of the photograph(s) and textual material (case histories).

I hereby agree to release and discharge Mr. Jino P J, and any editors or other contributors and their agents, publishers, successors and assigns from any and all claims, demands or causes of action that I may now have or may hereafter have for libel, defamation, invasion of privacy, copyright or moral rights or violation of any other rights arising out of or relating to any use of my image or case history.

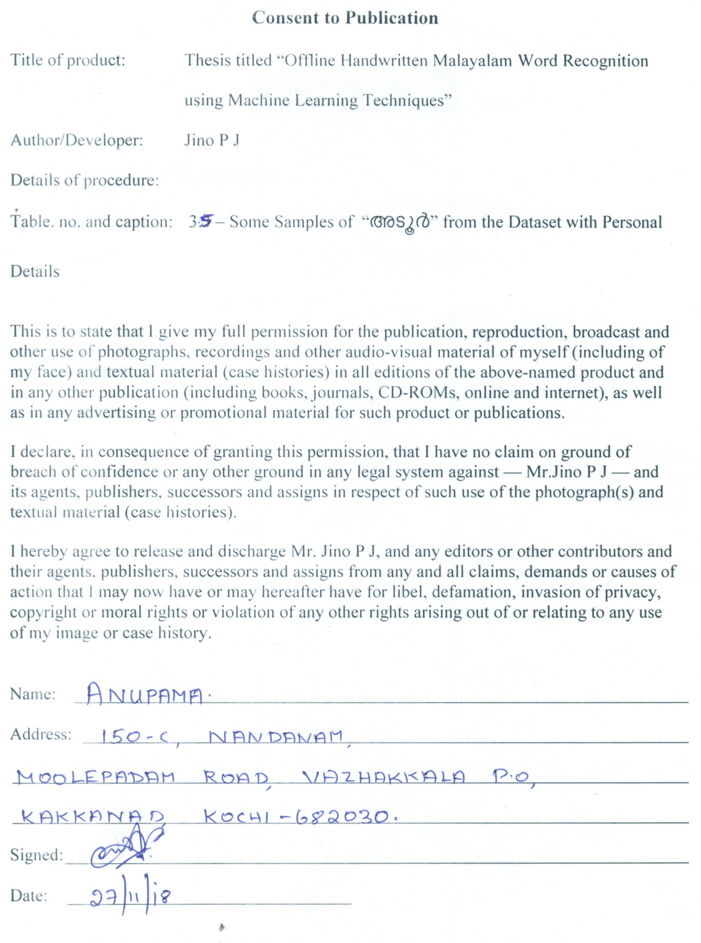
Name: Georgekutty M.

Address: Menachery house, Vengeri (PO), Kozhikode-673010

Signed: George kutty M

Date: 27/11/2018

Figure A.1: Consent of George Kutty



Consent to Publication

Title of product: Thesis titled "Offline Handwritten Malayalam Word Recognition using Machine Learning Techniques"

Author/Developer: Jino P J

Details of procedure:

Table. no. and caption: 3.5 – Some Samples of "ഞട്ടൽ" from the Dataset with Personal Details

This is to state that I give my full permission for the publication, reproduction, broadcast and other use of photographs, recordings and other audio-visual material of myself (including of my face) and textual material (case histories) in all editions of the above-named product and in any other publication (including books, journals, CD-ROMs, online and internet), as well as in any advertising or promotional material for such product or publications.

I declare, in consequence of granting this permission, that I have no claim on ground of breach of confidence or any other ground in any legal system against — Mr.Jino P J — and its agents, publishers, successors and assigns in respect of such use of the photograph(s) and textual material (case histories).

I hereby agree to release and discharge Mr. Jino P J, and any editors or other contributors and their agents, publishers, successors and assigns from any and all claims, demands or causes of action that I may now have or may hereafter have for libel, defamation, invasion of privacy, copyright or moral rights or violation of any other rights arising out of or relating to any use of my image or case history.

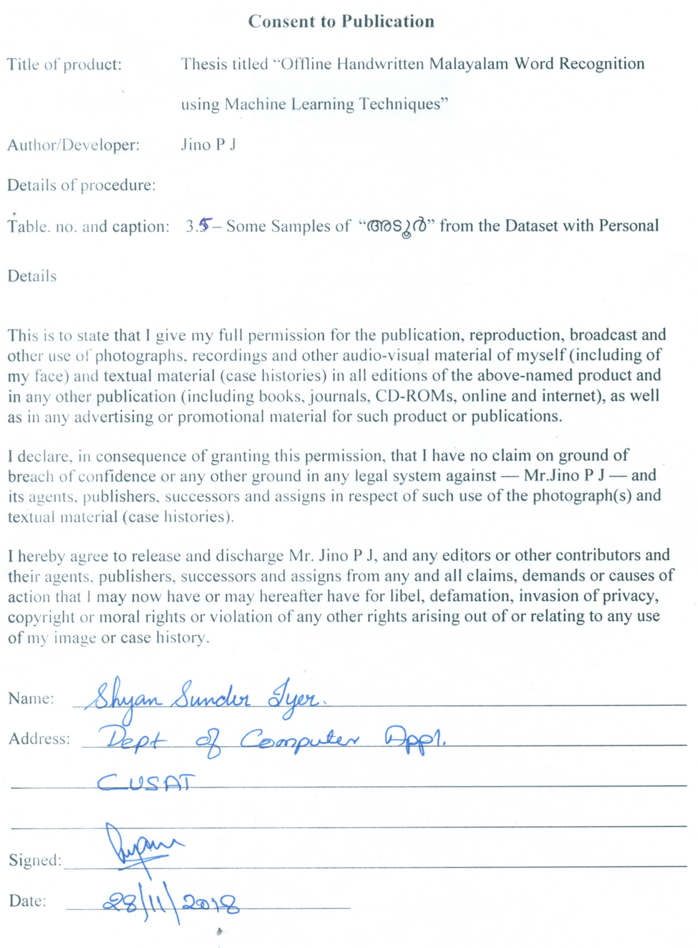
Name: ANUPAMA.

Address: 150-C, NANDANAM,
MOOLEPADAM ROAD, VAZHAKKALA P.O,
KAKKANAD KOCHI-682030.

Signed:

Date: 27/11/18

Figure A.2: Consent of Anupama



Consent to Publication

Title of product: Thesis titled "Offline Handwritten Malayalam Word Recognition using Machine Learning Techniques"

Author/Developer: Jino P J

Details of procedure:

Table. no. and caption: 3.5 – Some Samples of "അടൂർ" from the Dataset with Personal Details

This is to state that I give my full permission for the publication, reproduction, broadcast and other use of photographs, recordings and other audio-visual material of myself (including of my face) and textual material (case histories) in all editions of the above-named product and in any other publication (including books, journals, CD-ROMs, online and internet), as well as in any advertising or promotional material for such product or publications.

I declare, in consequence of granting this permission, that I have no claim on ground of breach of confidence or any other ground in any legal system against — Mr.Jino P J — and its agents, publishers, successors and assigns in respect of such use of the photograph(s) and textual material (case histories).

I hereby agree to release and discharge Mr. Jino P J, and any editors or other contributors and their agents, publishers, successors and assigns from any and all claims, demands or causes of action that I may now have or may hereafter have for libel, defamation, invasion of privacy, copyright or moral rights or violation of any other rights arising out of or relating to any use of my image or case history.

Name: Shyam Sunder Iyer.

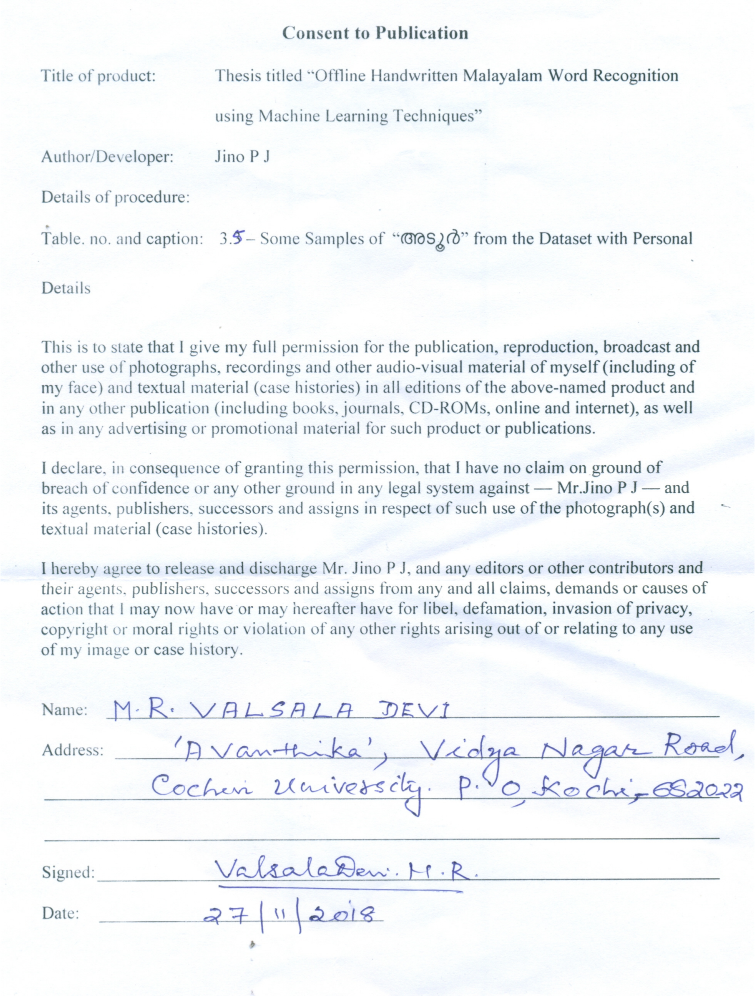
Address: Dept of Computer Appl.

CUSAT

Signed:

Date: 28/11/2018

Figure A.3: Consent of Shyam Sunder Iyer



Consent to Publication

Title of product: Thesis titled "Offline Handwritten Malayalam Word Recognition using Machine Learning Techniques"

Author/Developer: Jino P J

Details of procedure:

Table. no. and caption: 3.5 – Some Samples of "അടൂർ" from the Dataset with Personal Details

This is to state that I give my full permission for the publication, reproduction, broadcast and other use of photographs, recordings and other audio-visual material of myself (including of my face) and textual material (case histories) in all editions of the above-named product and in any other publication (including books, journals, CD-ROMs, online and internet), as well as in any advertising or promotional material for such product or publications.

I declare, in consequence of granting this permission, that I have no claim on ground of breach of confidence or any other ground in any legal system against — Mr.Jino P J — and its agents, publishers, successors and assigns in respect of such use of the photograph(s) and textual material (case histories).

I hereby agree to release and discharge Mr. Jino P J, and any editors or other contributors and their agents, publishers, successors and assigns from any and all claims, demands or causes of action that I may now have or may hereafter have for libel, defamation, invasion of privacy, copyright or moral rights or violation of any other rights arising out of or relating to any use of my image or case history.

Name: M.R. VALSALA DEVI

Address: 'Avanthika', Vidya Nagar Road,
Cochin University. P.O, Kochi, 682022

Signed: ValsalaDevi. M.R.

Date: 27/11/2018

Figure A.4: Consent of Valsala Devi