NOVEL APPROACHES FOR BASIC UNIT MODELING IN ONLINE HANDWRITING RECOGNITION

SUBHASIS MANDAL

NOVEL APPROACHES FOR BASIC UNIT MODELING IN ONLINE HANDWRITING RECOGNITION

A

Thesis submitted

for the award of the degree of

DOCTOR OF PHILOSOPHY

By

SUBHASIS MANDAL

DEPARTMENT OF ELECTRONICS AND ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

GUWAHATI - 781 039, ASSAM, INDIA

August 2019

Certificate

This is to certify that the thesis entitled "**NOVEL APPROACHES FOR BASIC UNIT MODELING IN ONLINE HANDWRITING RECOGNITION**", submitted by **Subhasis Mandal** (136102002), a research scholar in the *Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati*, for the award of the degree of **Doctor of Philosophy**, is a record of an original research work carried out by him under my supervision and guidance. The thesis has fulfilled all requirements as per the regulations of the institute and in my opinion has reached the standard needed for submission. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dated:
Guwahati.

Prof. S. R. Mahadeva Prasanna
Professor
Dept. of Electronics and Electrical Engg.
Indian Institute of Technology Guwahati
Guwahati - 781 039, Assam, India.

Dated:
Guwahati.

Dr. S. Sundaram
Associate Professor
Dept. of Electronics and Electrical Engg.
Indian Institute of Technology Guwahati
Guwahati - 781 039, Assam, India.

To

My Mother

A strong and gentle soul who taught me to believe in hard work and that so much could be done with little

My Father

For earning an honest living for us and for support and encouraging me to believe in myself

My guides

Prof. S. R. Mahadeva Prasanna

&

Dr. Suresh Sundaram

For their guidance, inspiration, literary and philosophical training which I have received throughout my research work

Acknowledgements

The journey of PhD is a long and challenging path and I am deeply indebted to several people for helping me complete this voyage. At the very outset, I would like to express my deepest and most sincere gratitude to my thesis supervisors, Prof. S. R. Mahadeva Prasanna and Dr. Suresh Sundaram for their excellent guidance, encouragement and support throughout my PhD. Their valuable advice, insightful feedbacks have immensely helped me at each and every step throughout this research endeavour. I am greatly inspired by their attitude towards research, creative thinking and enthusiasm for work, which will definitely go a long way in my professional career. I am also highly grateful to them for the pain they undertook in scrutinizing and improving the quality of all my manuscripts as well as the thesis.

I would like to express my gratitude towards my Doctoral Committee (DC) members Prof. Samarendra Dandapat, Prof. Rohit Sinha, and Dr. Priyankoo Sarmah for their suggestions and insightful comments to my dissertation. A special acknowledgement to Prof. Rohit Sinha for helping me in one of the collaborative work and I must admit, I am able to learn many things by interacting with him during that period, for which I shall be grateful forever. I am also thankful to Dr. Prithwijit Guha for his suggestions and motivations at different stages of this journey. I would like to acknowledge other faculty members and the office staffs of the Department of Electronics and Electrical Engineering, IIT Guwahati, for their care and support. Further, I am grateful to the Director, IIT Guwahati for the academic support and the facilities provided to carry out this research work at the Institute.

I also owe my gratitude to my seniors Dr. Deepak K.T., Dr. Syed Shahnawazuddin, Dr. Tousif Khan N, Mrs. Karnika Biswas, Mr. Suman Roy, Dr. Biswajit Dev Sarma, Dr. Nagaraj Adiga and Dr. Banriskhem K Khonglah for their help and suggestions at different stages of my work. A special thanks to Dr. Arghya Chakravarty and Dr. Rohan Kumar Das, my seniors as well as friends, for all their help and support.

I am thankful to my friends Ramesh, Suman, Bidisha, Abhishek, Rajib, Prateek, Ashis, Dipankar, Balaji, Sreeram, Protima, Nagendra, Sukanya, Saswati, Sishir, Akhilesh, Vikram, Sandeep, Shikha, Mrinmoy, Sarfaraz, Moakala, Vineeta, Prabhakar, Samarjeet, Tilendra, Alex, Ato, Indrajit, Debajyoti, Kamakshi, Pradipta, Vivek, Gautam and the rest for their direct/indirect contributions and making my life easier at IIT Guwahati. A special thanks to Himakshi, my friend and colleague, for all of her

help and support and for all the technical / non-technical discussions that we used to do during the journey of PhD. Further, I shall remain grateful to all of my close friends Arindam, Kamal, Dipankar, Sudipta, Chinmoy, Shambo, Pratik, Mrinmoy, Illena, Amitavo, Suvodip and Subhojit for being a pillar of emotional strength throughout my life. Also, I am thankful to all the members of Signal Informatics Lab for being there during my PhD journey and provided help whenever I needed it.

During the PhD period, I was funded by the International Association for Pattern Recognition (IAPR) for attending ICPR 2018 conference in abroad, for which I would like to acknowledge them. Further, I would offer my sincere thanks to MHRD, Govt. of India for providing the fellowship to carry out my PhD thesis work.

Finally, I thank my parents for their constant blessings, support and silent prayers for my success. Last but not least, a special thanks to my sibling Sukla and Souvik who took all the responsibilities of home so that I can work peacefully far away from home.

Subhasis Mandal

Abstract

An unconstrained, writer-independent, large vocabulary online handwritten word recognition system can be developed by modeling the basic recognition unit of the script. The basic recognition unit or the basic unit for brevity, represents the set of patterns (characters/strokes) for which the handwriting models are created. This thesis focuses on developing novel methods for basic unit modeling in online handwriting recognition. Three different directions of work are presented in this thesis for basic unit modeling.

In the first part, we propose two novel strategies towards improving the classification ability of an existing basic unit recognition system by performing a reevaluation of the decision of a classifier. To begin with, we propose a strategy to reevaluate the decision of an HMM-based basic unit recognition system. The choice of the HMM classifier is owing to its success in several prior works of online handwriting recognition. In a conventional system, a test sample is recognized by first computing the log-likelihood scores from each of the class-specific HMMs. The class corresponding to the highest score is assigned to the sample. We demonstrated that, at times, the sole dependence of log-likelihood score of the HMM states may not be effective in capturing the finer nuances of the online trace that discriminates similar basic units / patterns. To alleviate this issue, we propose to analyze the HMM states corresponding to the top-2 confusion classes with the objective of identifying a subset of states (referred to as 'discriminative states') that can help provide cues to discriminate them better. Accordingly, we compare the log-likelihood scores for discriminative states of the two confused classes for the final decision. We also extend our proposal to develop a large vocabulary word recognition system by employing the HMM framework.

In the second reevaluation technique, we propose a novel strategy to detect parts of the trace that present fine structural differences in similar looking basic units, which are referred to as 'Discriminative Region' in this thesis. Subsequent to their extraction, features

are extracted with the hope of classifying the confused classes better. To identify the discriminative region, the proposed technique first splits the trace of the handwritten samples of a class into several segments and represents the distribution of each of them with the parameters of the k-means clustering. Thereafter, we apply the Dynamic Time Warping (DTW) algorithm to match the statistical characterizations associated with the basic units that form a confusion pair.

In addition, we also propose a single-stage classification framework that takes into consideration the discriminative regions extracted between the basic units. In particular, we employ an ensemble of $\binom{C}{2}$ classifiers corresponding to all possible pairs of classes of basic units, wherein each of the classifiers are trained by utilizing the features from the discriminative region. At the time of recognition, a majority voting scheme is applied to the ensemble for assignment of the identity to the test basic unit pattern. Further, the discriminative region analysis is extended for the case of large vocabulary word recognition task.

In the second part of the thesis, we focus on proposing two new feature representations for the basic units, mainly from a probabilistic viewpoint. Firstly, we propose a set of probabilistic features that are derived from a set of Gaussian mixture models (GMMs). The GMMs, being a generative model are intended to capture the class dependent characteristics. We show that the so derived posterior features aid in minimizing intra-class variability in the feature space while at the same time improving the separability between classes.

The second feature representation is extracted directly from the trace of online handwriting by adopting a convolutional neural network (CNN). To the best of our knowledge, this is the first work of its kind that applies CNN directly on the (x, y) coordinates of the online handwriting data. The efficacy of the proposed GMM and CNN features are demonstrated for basic unit and large vocabulary word recognition tasks.

In the third part of the thesis, we explore a novel classification approach to model the basic units. Our proposal is based on the utility of a hybrid deep neural network - hidden Markov model (DNN-HMM) framework, that is quite prominent in the research of speech recognition. In a conventional HMM-based system, the Gaussian mixture model (GMM)

is used to compute the value of the probability distribution of a given observation assigned to a HMM state. Different to it, in this work, a deep neural network (DNN) is trained to output the posterior probabilities of the observations. The posteriors are then converted into quasi-likelihoods by dividing them by the prior of the states. These quasi-likelihoods are then used with the HMM during the time of testing.

The key feature of the proposed system is that the DNNs with many hidden layers are capable of learning very complex and highly non-linear relationships between the input and output. They can also incorporate high-dimensional feature vectors (obtained by considering contextual information) which help the system to make a better prediction of the input data. In the experimental analysis, we demonstrate the effect of several hidden layers in developing the DNN-HMM framework for online handwritten basic unit and word recognition systems.

Furthermore, we combine the different explorations made in the thesis and evaluate the integrated system on online handwritten word samples. In particular, the features derived from the CNN are employed to characterize the data. Thereafter the DNN-HMM is used to model the basic unit model, with the reevaluation of the output class being performed by utilizing the discriminant region-based strategy.

We perform experiments for both basic unit and large vocabulary word recognition tasks on databases of two scripts namely

- the locally collected Assamese character and word datasets
- the publicly available English UNIPEN character and UNIPEN-ICROW-03 word datasets

We empirically show that the results obtained are promising with regard to the prior works reported in the literature.

Contents

List of Figures

List of Tables

List of Acronyms

ANN	Artificial Neural Network
BLSTM	Bidirectional Long Short-Term Memory
CNN	Convolution Neural Network
CTC	Connectionist Temporal Classification
DNN	Deep Neural Network
DNN-HMM	Deep Neural Network - Hidden Markov Model
DFT	Discrete Fourier Transform
DTW	Dynamic Time Warping
DTW-DDH	Dynamic Time Warping - Discriminative Distance Histogram
EM	Expectation Maximization
EMD	Earth Mover Distance
GMM	Gaussian Mixture Model
GMM-HMM	Gaussian Mixture Model - Hidden Markov Model
GDTW	Gaussian Dynamic Time Warping
HMM	Hidden Markov Model
LSTM	Long Short-Term Memory
MLP	Multilayer Perceptron
OnSNT	Online Scanning n-tuple Classifier
RBF	Radial Basis Function
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
SVM	Support Vector Machine
SDTW	Statistical Dynamic Time Warping
TDNN	Time Delay Neural Network

List of Symbols

α	Size of the window employed in the discriminative region selection technique
A	Transition matrix of HMM
$AR(i)$	Aspect ratio feature computed at i^{th} sample point of the trace
$a_{s_{t-1}\ s_t}$	State transition probability from state s_{t-1} to s_t in a HMM
B	Observation probability matrix of HMM
B_f	Context map feature
C	Number of basic unit classes
c_p	p^{th} basic unit class
(c_1, c_2)	A confusing pair
$\hat{c}$	Recognized basic unit class label
$\widehat{c}'$	Revised basic unit class label after reevaluation
Δ	Difference between successive x/y coordinates in the online trace
$DS^{(c_1,c_2)}$	Discriminative state for a confusing pair (c_1, c_2)
$DR^{(c_1,c_2)}$	Discriminative region for a confusing pair (c_1, c_2)
η	Parameter determining the window size for extracting frames from a word sample
γ_p	Learned GMM corresponding to the p^{th} basic unit
$\mathcal{H}$	Cross-entropy cost function
H	Number of hidden layers in the DNN
$h(.)$	Activation function in the DNN
$\mathcal{I}$	Indicator function
K	Number of clusters obtained from the k-means algorithm
λ_p	Learned HMM for the p^{th} basic unit
$\hat{\lambda}_p$	Learned HMM corresponding to the p^{th} word entry in the lexicon

L	Number of data segments obtained from the online trace of a basic unit pattern.
$LN(i)$	Linearity feature computed at i^{th} sample point of the trace
L_c	Number of feature vectors preceding and succeeding the current vector
$L^{(x,y)}$	Number of samples points in the online trace of a handwritten word
$\mathcal{L}$	Log-likelihood score obtained from the HMM trained on basic unit patterns
N_i	Number of basic units present in a word W_i
N_w	Number of unique words employed for collecting the database
$\boldsymbol{\mu}$	Mean vector
$\mathcal{M}$	Number of Gaussian components in a GMM
M	Number of words selected for re-evaluation
m	Length of the filter in CNN
N_s	Number of states in a HMM
N_{tot}	Total number of feature vectors in all the classes.
n_p	Number of samples in the p^{th} class / basic unit pattern
$\mathbf{O}$	Sequence of point-based feature vectors representing a complete basic unit sample
$\mathbf{o}_i$	Feature vector corresponding to the i^{th} sample point of an online handwritten basic unit / word pattern
$\mathbf{o}_{ji}$	Feature vector corresponding to the j^{th} sample point of the i^{th} frame
P	Probability measure
Π	Initial probabilities of the HMM states
q	Number of resampled points in a basic unit sample
$\{q_1, q_2, ..., q_T\}$	State sequence
$R_i^{c_p}$	A tuple of statistical quantities representing the i^{th} segment of basic unit class c_p
r	Value that determines the number of segments to be selected as the discriminant region DR^{c_1,c_2}
$\mathbf{\Sigma}$	Covariance matrix
$\sigma^2_{\mathcal{W}}$	Intra-class variance
$\sigma^2_{\mathcal{B}}$	Inter-class variance
$\mathcal{S}_{ip}$	Log-likelihood score of i^{th} point-based feature vector obtained from GMM γ_p
$\hat{\mathcal{S}}_{ip}$	Log-likelihood score corresponding to i^{th} frame, as obtained from the GMM γ_p

S	The underlying state sequence of an HMM
$SL(i)$	Slope feature computed at the i^{th} sample point of the trace
S_p^i	Revised voting-based recognition score of the p^{th} basic unit of word W_i post reevaluation
$[s_a,\ s_b]$	Starting and ending indices of the discriminative states
$state\_dism(i)$	EMD distance between the i^{th} states of two HMMs
θ_c	Angle used for calculating the curvature feature
θ_w	Angle used for calculating the writing direction feature
$\mathbf{V}$	Sequence of GMM feature vectors for a complete basic unit sample
$\widehat{\mathbf{V}}$	Sequence of GMM feature vectors for a complete word sample
$\boldsymbol{\mathcal{V}}$	CNN feature vector representing the online trace of basic unit sample
$\hat{\boldsymbol{\mathcal{V}}}$	Sequence of CNN feature vector representation for an online handwritten word sample
$\mathbf{v}_i$	Feature vector at the i^{th} point of a basic unit sample, as derived from the GMMs
$\widehat{\mathbf{v}}_i$	Feature vector representing the i^{th} frame of a word sample, as derived from the GMMs
$\mathcal{W}^*$	Optimal warping path in the DTW
W	Number of words in the lexicon
$\hat{w}$	Recognized word label
$\widehat{w}'$	Revised word label after reevaluation
w_{ji}	Weight connection between the j^{th} unit of current layer with the i^{th} unit of the previous layer in the DNN
$x'(i)$	First derivative feature of x coordinates at the i^{th} sample point of the trace
$x''(i)$	Second derivative feature of x coordinates at the i^{th} sample point of the trace
$y'(i)$	First derivative feature of y coordinates at the i^{th} sample point of the trace
$y''(i)$	Second derivative feature of y coordinates at the i^{th} sample point of the trace
$\hat{y}_p$	Probability value at the p^{th} output node, as estimated by the DNN
y_p	Target value at the p^{th} output node of the DNN

1

Introduction

Contents

1.1 Introduction

Handwriting recognition refers to the task of transcribing the content of handwritten messages by providing intelligence to a machine. The input to a handwriting recognition system is a set of patterns that can be obtained from sources such as paper, photograph, touch screen and electronic pen-based devices. Based on the mode of data capture, they are broadly classified into offline and online.

In online handwriting recognition systems, the handwritten data is obtained with the help of a transducer such as an electronic or tablet digitizer [1–3]. These devices capture the pen-tip movement trajectory as a sequence of (x, y) coordinates sampled uniformly over time. As such, pen-tip movements are detected with pen-up/pen-down states. The coordinates of the successive points recorded as a function of time is referred to as a temporal trace. In particular, a pen-down state occurs when the pen touches the digitizer (writing pad) and when the pen is lifted off, a pen-up state is sensed. The set of points captured between successive pen-down to pen-up states is called a stroke. It may be noted that the data being utilized for online recognition contains information on the number of strokes and their order. On the other hand, in offline handwriting recognition systems, the data is captured in the form of an image by scanning a handwritten document [4–6]. The entire message is written on a media such as paper and brought to the scanner to generate the bitmap image of the handwriting. Thereafter, techniques from the area of image processing are used to recognize the handwritten text.

Technology-based on recognizing online handwritten data can be incorporated into a wide range of devices and has applications [1, 7–10] ranging from messaging on personal devices to annotation / transcription of data that can be used for indexing and querying. There is also the possibility of using it in conjunction with speech synthesis, thereby empowering people with vocal disability to communicate with others. Over the past decades, there has been a lot of research in this area, thanks to the availability of electronic tablets and similar devices that are enabled with handwriting-based input. As a matter of fact, such systems are deployed in smart-phones, PDA-style computers and tablet-PC.

The main crux of the thesis is on developing novel methods for online handwriting recognition systems. Our primary focus, as we shall see later, will be on proposing innovative strategies with regard to two aspects, namely feature extraction and recognition.

1.2 Online handwriting recognition system

Recognition accuracy is an important parameter that portrays the performance of an online handwriting recognition system. By placing constraints on the writing styles, writers, or lexicon (set of words that the system is going to recognize), one can get reasonable accuracy. In the following, we outline the different aspects of categorizing online systems, as suggested by works in the literature [6, 11–13]:

- **Constrained and unconstrained systems:** A constrained handwriting recognition system can be developed by placing specific constraints on writing styles. These primarily include forcing the user to write in a discrete manner (by providing a predefined box). Another limitation that may be imposed is to ensure that the users provide the handwritten data in given stroke order, that may be based on a script specific requirement. On the other hand, an unconstrained handwriting recognition system allows users to freely write in their own natural way, where the writing is typically mixed with discrete and cursive styles. Without loss of generality, the recognition of unconstrained handwriting is more challenging when compared to the constrained writing scenario.

- **Writer dependent and writer independent systems:** The goal of a writer-independent system is to recognize handwriting from users whose writing the system may not have seen during training, while writer-dependent systems are trained to recognize handwriting of a single individual. In general, writer dependent systems can present a better accuracy rate when compared to writer independent scenarios. Nevertheless, from the works in literature, one can infer that the focus of handwriting recognition, in general, is to rely on invariant representations of the handwritten content, typically achieved by reducing the extent of inter-writer variations.

- **Lexicon-based and lexicon-free systems:** A lexicon plays an important role in determining the difficulty of a handwriting recognition task. Systems that are based on a lexicon identify a word from a predetermined set of words in a dictionary. This makes them different from lexicon-free systems, where the recognition is performed without the help of a dictionary.

  The size of a lexicon is an important factor that determines the difficulty of modeling lexicon-based handwriting recognition systems. Typically, a system relying on a small lexicon is easier to build, owing to the smaller number of confusing word pairs and direct modeling of individual

words by means of a holistic approach. However, as we encounter larger lexicon sizes, the use of such approach becomes impractical, thereby necessitating the utility of lower level primitives such as characters for modeling. It may be mentioned in passing that lexicon-based and lexicon-free systems are also referred to as closed and open vocabulary systems respectively in the literature.

Further to the above, the handwriting system for word recognition can be developed by either a holistic or analytic-based approach [2]. The former treats the word as a single, indivisible entity and aims to recognize it as a whole without any segmentation. Each word is considered as a class and the system is trained to classify the word directly. As the classifier is trained for all the words, the holistic approach is applicable for the scenario of limited vocabulary word recognition task. On the other hand, in an analytic-based approach, the input word is considered as a sequence of basic recognition units (such as characters, strokes or sub-strokes) that are first segmented either explicitly or implicitly and then recognized to build a word-level interpretation.

In the explicit analytic scheme, a segmentation module fragments the input word into basic units by adopting an over-segmentation and classify framework [14]. In the over-segmentation step, the input string pattern is over-segmented into primitive segments such that each segment comprises a single character or a part of a character. Each segmentation path represents a set of candidate patterns, generated by combining successive primitive segments obtained from the over-segmentation step. Thereafter, by employing a suitable classifier, the segmentation-recognition lattice is generated. The recognition is performed by evaluating the score of each lexicon word in the lattice by using a dynamic programming approach.

In the implicit analytic scheme, no explicit segmentation is performed and as the name implies, the segmentation boundaries in the word are obtained as a bi-product of recognition [11]. This scheme follows the time-sequence interpretation framework [15] by adopting classifiers such as hidden Markov model (HMM), time-delay neural network (TDNN) and recurrent neural network (RNN) [1], to state a few. Typically, in literature, a large vocabulary word recognition system is developed by employing the analytic approach and hence we focus on the same while discussing the contributions of the thesis in the following Section.

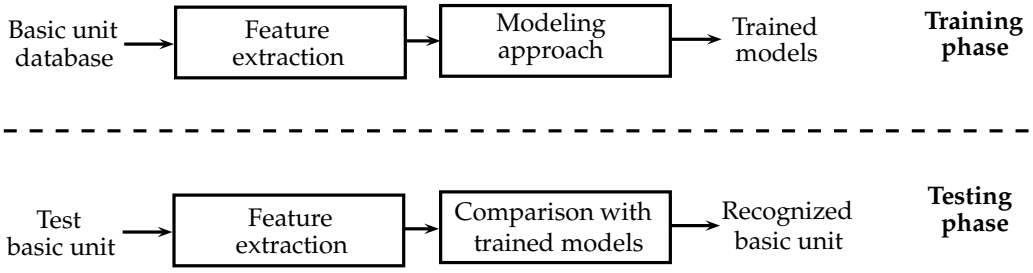


Figure 1.1: Block diagram of a basic unit recognition system. The basic units may corresponds to isolated digits, characters, or any other suitable recognition primitives for which handwriting models are created.

1.3 Contributions of the thesis

An unconstrained, writer-independent large vocabulary online handwritten word recognition system can be developed by modeling the basic recognition unit [1]. The basic unit represents the set of patterns (characters/strokes) for which the handwriting models are created. As stated in the previous Section, the modeling of such a unit is an essential aspect to be considered in the development of analytic-based handwriting recognition systems. In our research, we focus our proposals with regard to two aspects, namely

(i) Novel feature extraction methods to describe the basic units

(ii) New classification modeling techniques to recognize the basic units

Three different directions of work are presented in this thesis with regard to the basic unit recognition system outlined in Fig. 1.1. In the first part, we propose two novel strategies towards improving the classification ability of an existing basic unit recognition system by performing a reevaluation of the decision of a classifier. These approaches are discussed with sufficient elaboration in Chapters 3 and 4 respectively. As a second direction, in Chapter 5, we focus on investigating new features for describing the basic unit, mainly from a probabilistic viewpoint. In Chapter 6, we explore a novel classification approach to model the basic units. Our proposal is based on the utility of a hybrid deep neural network, namely the hidden Markov model (DNN-HMM), that is quite prominent in the research of speech recognition.

With respect to demonstrating the efficacy of our different proposals, we evaluate them on both the basic units and words of two scripts, namely English and Assamese [2]. As we shall see in the

[1] For brevity, hereinafter, we refer to the basic recognition unit as basic unit.

[2] Assamese is a local language spoken in the Indian state of Assam, where our University, the Indian Institute of Technology Guwahati is located.

contributing Chapters, a large vocabulary implicit-segmentation word recognition system is utilized by employing the HMM framework, that serves as a baseline classifier for comparison.

In the following subsections, we elaborate on our different proposals that are employed for the modeling of the basic unit.

1.3.1 Discriminative HMM states

The task of online handwriting recognition becomes often challenging due to the presence of similar shape basic unit classes (referred to as a confusion pair) [16–18]. To address this problem, most works reported in the literature employ a two-stage classification framework, wherein the decision of the first-stage classifier is reevaluated to reduce the confusion. Typically, in such systems, a second-stage classifier is used to analyze the top-2 output classes from the first-stage (that get confused) and to appropriately select the most relevant one.

In this Chapter, we propose a strategy to reevaluate the decision of a hidden Markov model (HMM) based basic unit recognition system, that in turn alleviates the necessity of a different second-stage classifier. Typically, in a conventional HMM-based system, a model is constructed separately for each basic unit by using the Baum-Welch estimation algorithm [19]. Whenever a test basic unit is to be recognized, we first compute the log-likelihood scores from each of the class-specific HMMs. Thereafter, the class corresponding to the highest obtained score is assigned to the sample.

With regard to the single-stage HMM framework, we demonstrate that, at times, the sole dependence of log-likelihood scores may not be effective in capturing the finer nuances of the online trace that discriminate similar shape patterns / basic units. In order to circumvent this issue, we propose to analyze the HMM states corresponding to the top-2 confusion classes with the objective of identifying a subset of them that can help provide cues to discriminate them better. Subsequent to the identification of the so-called 'discriminative states', we compare their log-likelihood scores with regard to the HMMs of the two confused classes and accordingly make the final decision.

Said in another way, our main contribution is coming up with a strategy that can automatically detect the discriminate states between the HMMs of two basic units, that are most likely to get confused. We demonstrate that the utility of employing likelihood scores computed over discriminative states assists in reducing the confusions between similar looking basic units.

We also extend our proposal to develop a large vocabulary word recognition system employing the HMM framework. Given a test word sample, first, we obtain the top-M most probable words

from a baseline HMM-based system and then reevaluate the scores of each of them for refining the recognition decision. The boundaries of the basic units for each top-M word are obtained from the baseline HMM system via the implicit segmentation process. Subsequent to obtaining the segments / basic units of a probable word choice, the likelihood score for each of the basic units is revised by taking into consideration the discriminative states. The lexicon word corresponding to the maximum average revised likelihood score is selected as the recognized output for the handwritten sample.

The performance of the proposed system employing discriminative states is evaluated on databases of two scripts, namely the locally collected Assamese character and word datasets, as well as the publicly available English UNIPEN character and UNIPEN-ICROW-03 word datasets. The results obtained suggest an improvement over the conventional HMM-based system across each of the datasets.

1.3.2 Discriminative Regions in basic units

In this part of the thesis, we present a novel strategy to detect parts of the trace that present fine structural differences in similar looking basic units. In the context of our work, we refer to such parts as 'Discriminative Region'. Subsequent to their extraction, we describe them with point-based local features to help reduce the degree of confusion.

The proposed technique first splits the trace of the handwritten samples of a class into several segments and describes them with the parameters of the k-means clustering. Thereafter, we apply the Dynamic Time Warping (DTW) algorithm to match the statistical information associated with the basic units forming a confusion pair. The distances in the cost matrix along the optimal warping path are analyzed to select the discriminative region. In this work, we utilize the Earth Movers Distance (EMD) [20] measure to compute the distance between two distributions while generating the cost matrix.

With regard to the classification strategy, we propose a single-stage framework that takes into consideration the discriminative region extracted between the basic units. In the conventional setup, the discriminant region-based system employs a two-stage framework [17, 18, 21] wherein, the top-2 outputs of the first classifier are fed to the second classifier for refining the output. Different to it, we propose an ensemble of $\binom{C}{2}$ classifiers [3], corresponding to all possible pairs of basic unit classes by using a one-vs-one strategy. The classifiers are in turn trained by utilizing the features extracted from the discriminative region pertaining to each combination of basic units. Thereafter, at the time

[3] We assume here that C represents the number of basic unit classes in the recognition system.

of recognition, a majority voting scheme is applied to the ensemble for assignment of the identity to the test basic unit sample.

Moving further, we also develop an HMM-based large vocabulary word recognition system by incorporating the discriminant region-based processing, for refining the word recognition output. For a given test word sample, the baseline HMM-based system provides top-M most probable words. Following the generation of the segments of the basic units of a probable word choice, the scores of each of them are revised by passing through a subset of $(C-1)$ classifiers in the ensemble. It is important to note here that each of the selected classifiers are trained by employing the feature vectors extracted from the appropriate discriminant region. For obtaining the recognized output, the lexicon word with the maximum average revised score is considered.

In order to demonstrate the efficacy of the proposed recognition framework, we present several experiments on the English and Assamese character and word datasets.

1.3.3 Utility of novel features

In this Chapter of the thesis, we focus on proposing new feature representations for the basic units that are to be recognized. An online handwriting system is typically developed by considering point-based features that describe different geometric attributes of handwriting [10]. Often, due to the wide variations in the writing styles, the use of point-based features results in high intra-class variability in the feature space. One aspect that has hardly been sought after in the literature is to consider the extracted feature vectors, being utilized, from a probabilistic viewpoint. We show in this Chapter that such a representation can indeed help capture the class dependent characteristics, thereby improving the inter-class discrimination.

As a step in the above direction, in the first part of the Chapter, we propose the use of probabilistic features (referred to as 'posterior features') that are derived from a set of Gaussian mixture models (GMMs). The GMMs, being a generative model are intended to capture the class dependent characteristics. We show that the so derived posterior features aid in minimizing intra-class variability in the feature space. The performance of the proposed GMM posterior features is demonstrated for both basic unit and word recognition tasks of the Assamese and English databases. The results show a notable improvement over the conventional point-based features that have been used for online handwriting recognition.

The GMMs are trained separately for each basic unit class by employing the Expectation Max-

imization (EM) algorithm. In order to extract the posterior features, the sequence of point-based features derived from the trace of the input sample are fed to each of the trained GMMs. Thereafter, a classifier such as the HMM or Support Vector Machine (SVM) is learned on the GMM feature space to develop the system for basic unit recognition.

For the recognition of online handwritten words, the GMMs are trained on the basic units, that are obtained from segmenting the word samples. To obtain the posterior features, first, a sliding window technique is used to extract frames from the word samples. Thereafter, the pattern associated with each frame is pre-processed and a d-dimensional point-based feature vector is extracted at each (x, y) point. These feature vectors are then passed through the GMMs for obtaining the posterior representation. Once the sequence of GMM features are available, the HMMs are built for each of the words in the lexicon by process of the concatenation of their constituent basic units.

In the second part of this Chapter, we aim to extract features directly from the trace of online handwriting, thus alleviating the need for hand-crafted features. In this direction, a convolution neural network (CNN) is developed to process the online handwriting data. It may be noted that prior works in the literature employing such architecture first convert the online handwritten input to its corresponding bitmap image [22]. However in the process of conversion, the utility of important dynamic information such as pen-up/pen-down status, stroke order/directions are likely to get ignored. In order to circumvent this issue, our CNN architecture operates on the online handwriting data directly, thereby eliminating the need of converting it to an offline image.

The first convolution layer accepts the sequence of (x, y) coordinates along the trace of the basic unit as an input and outputs a convolved filtered signal. Thereafter, via alternating steps of convolution and Rectified Linear Unit (ReLU) layers, in a hierarchical fashion, we obtain a set of deep features that can be employed for classification. To the best of our knowledge, this is the first work of its kind that applies CNN directly on the (x, y) coordinates of the online handwriting data. The results on the basic unit and word recognition tasks demonstrate the advantage of the proposed features over the hand-crafted point-based features.

1.3.4 Exploration of DNN-HMMs

The work presented in this Chapter explores a hybrid deep neural network - hidden Markov model (DNN-HMM) framework for basic unit modeling. In a conventional HMM-based system, the Gaussian mixture model (GMM) is used to compute the value of the probability distribution of a given

observation assigned to a state. Different to it, in this work, we investigate the merit of a DNN-HMM framework that has gained much success in the field of speech recognition [23]. In this approach, a deep neural network (DNN) is trained to output the posterior probabilities of the observations. The posteriors are then converted into quasi-likelihoods by dividing them with the prior of the states. These quasi-likelihoods are then used with the HMM during the time of testing.

In general, the DNN-HMM can provide a powerful modeling capability when compared to the GMM-HMM. This is primarily owing to the following reasons.

- The GMMs are generative by nature, while DNNs follow a discriminative modeling paradigm. As such, they can adequately model any kind of non-linear functions of the input and hence do not require any prior assumption of the input distribution.
- Different to the GMMs, the DNNs can process high-dimensional feature vectors (obtained by considering contextual information) to make a better prediction of the input data.
- The GMMs typically make a strong assumption of diagonal covariances while dealing with high-dimensional data. The DNNs, on the other hand, do not make any such assumptions and can well handle such data with the fully connected layers.

The DNNs with many hidden layers are capable of modeling very complex and highly non-linear relationships between the input and output. To train the DNN, first, a GMM-HMM system is built. Thereafter, each of the d-dimensional observations in the training data are assigned to one of the HMM states. The input to the DNN is a single $[d \times (2L_c + 1)]$ dimensional feature vector which is generated by combining $(2L_c + 1)$ context feature vectors (center feature vector with a context of L_c feature vectors at each side). The associated target label for this generated feature vector is the assignment of the state index corresponding to the center feature vector (namely, the $(L_c + 1)^{th}$ feature vector). The DNN is trained discriminatively by using the back-propagation algorithm with the cross-entropy cost function.

In the experimental analysis, we demonstrate the effect of several hidden layers in developing the DNN-HMM system for online handwriting. The results demonstrate that a notable improvement is achieved with the proposed DNN-HMM system when evaluated for both the basic unit and word recognition tasks.

In the final part of this contributing Chapter, we combine the different explorations made in

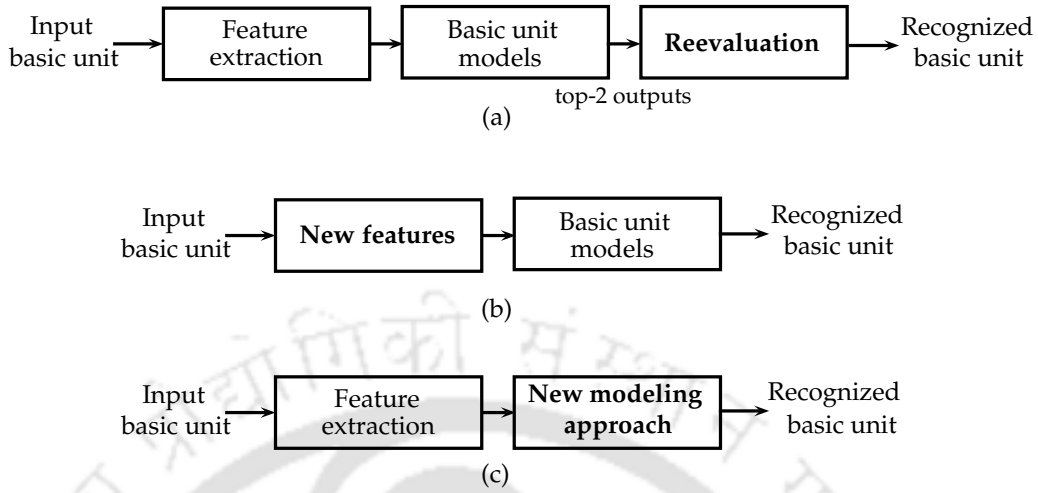


Figure 1.2: Overview of the three proposed directions explored in this thesis to improve the basic unit modeling. The sub-figure (a) depicts the first part of work where we reevaluate the decision of the classifier to refine the output (Chapters 3 and 4). Likewise, the sub-figure (b) depicts the second direction where we explored new features (GMM posterior and CNN features) to represent the online handwriting (Chapter 5). The sub-figure (c) presents the third part wherein the DNN-HMM framework from the area of speech recognition is investigated to model the basic units. This is discussed in Chapter 6.

the thesis and evaluate the integrated system on online handwritten word samples. In particular, the features derived from CNN are employed to characterize the data. Thereafter the DNN-HMM framework is used to model the basic unit, with the reevaluation of the output class being performed by considering the discriminant region-based strategy.

Before moving ahead, we summarize pictorially in Fig. 1.2, a pipeline of the contributions made in this thesis.

1.4 Databases for the present study

We have considered two online handwritten databases namely Assamese and UNIPEN database to conduct the experiments in this thesis. In the two following subsections , we give the details of each of them, with a description of the lexicon to be used for the experiments in word recognition. However, to begin with, we provide a brief description of the Assamese script below.

Assamese is one of the Indic scripts that originate from the Indo-Aryan language. It is mainly used in the state of Assam in North-East India and is the official language of this state, spoken by around 13 million people. It is also spoken in parts of Arunachal Pradesh and other North-East Indian states. Assamese has derived its phonetic character set and its behavior from Sanskrit. The language is written from left to right using the Assamese script.

The basic character set used to write Assamese consists of 10 digits, 11 vowels, 41 consonants, 10 vowel modifiers and 2 consonant modifiers [24]. Apart from these, there are several compound characters which are generated by combining potentially all the consonants among themselves. Likewise, the modifiers can be merged with the majority of the consonants/compound characters thereby producing thousands of new modified characters. Thus, by considering all possible combination, the character set ranges from several hundred to a few thousand. However, it is to be borne in mind that some of the characters are not frequently used to write Assamese.

1.4.1 Assamese database

The Assamese database contains both isolated character and word samples. The samples are collected using a Lenovo Tablet PC-X230, that captures the sampled (x, y) coordinate values of pen movement, pen-down/pen-up status. The participants who contributed data comprised senior high school students studying in schools in Guwahati and adults in the age group 18 to 25 years with minimum Bachelor qualification.

- *Character dataset:* This dataset contains samples for 10 digits, 52 basic characters and 95 modified characters collected from 200 native Assamese writers. It is divided into disjoint training, testing and validation sets and the number of samples present in each set is given in Table 1.1. It may be noted that the division is performed in a writer independent fashion, without taking into consideration the information of the contributors.

- *Word dataset:* The word dataset contains samples for 182 unique words, collected from 163 writers. Two samples for each of the words are collected in two different sessions. Each participant is asked to write the words in the respective box displayed on the screen of Tablet PC. Similar to the character dataset, the samples are divided into disjoint training, validation and testing sets in a writer independent way. The statistics of each of these sets is shown in Table 1.1.

Both the above datasets were collected in 2012 and were available at the start of this research in 2014. The generation of the databases and selection of word list for data collection were pursued as a deliverable of the project - "Online Handwriting Recognition System for Assamese", funded by Technology Development for Indian Languages, Ministry for Communication and Information Technology [25]. Moreover, the number of basic units (namely 157 and 173 for the Assamese character

Table 1.1: Description of the classes and number of samples in the Assamese character and word datasets.

Assamese character dataset					Assamese word dataset				
Category	Class	Train	Validation	Test	Category	Class	Train	Validation	Test
Digit	10	1630	404	1066					
Basic character	52	7238	1753	4366	Word	182	16208	2941	8414
Modified character	95	16271	3803	9817					

and word datasets [4]) were pre-decided by personnel involved in the annotation of the collected data. These are presented in Figures 1.3 and 1.4 respectively.

Fig. 1.5 and 1.6 depicts several samples of characters and words from the datasets. For research use, we have made the data publicly available in the following link

`https://drive.google.com/drive/folders/134nOvI9Lb60JCiz63mPTStbDJm-8ceUR?usp=sharing`

.

1.4.2 UNIPEN database

This database is a publicly available online handwritten English database provided by International UNIPEN Foundation. More specifically, we have considered UNIPEN character and UNIPEN ICROW-03 word datasets, the details of which are given as follows.

- *UNIPEN character dataset:* The UNIPEN character set [26] contains isolated English digit, uppercase and lowercase letters corresponding to the sections $1a$, $1b$ and $1c$ of the UNIPEN Train-R01/V07 dataset. The dataset has several mislabeled and fragments of characters, that are removed by applying an aspect-ratio based cleaning method [27]. Each category of data is further split into disjoint training, validation and test sets. The total number of samples present in each category is given in Table 1.2.

- *UNIPEN ICROW-03 word dataset:* This dataset [28] is written in English and contains freestyle writing such as hand-print, cursive and mixed. A set of 72 writers of different nationalities contributed a total of 13,119 samples that consist of 884 unique words. The total data as such is divided into a disjoint training set (7689 words from 41 writers), validation set (1334 words from 8 writers) and testing set (3943 words from 23 writers). The transcription of the dataset contains 44 basic units, including all lowercase letters, uppercase letters and one special symbol.

[4] Without loss of generality, the number of basic units used for recognition of data of a given script is not unique.

(a)

১	২	৩	৪	৫	৬	৭	৮	৯	০

(b)

অ	আ	ই	ঈ	উ	ঊ	ঋ	এ	ঐ
ও	ঔ	ক	খ	গ	ঘ	ঙ	চ	ছ
জ	ঝ	ঞ	ট	ঠ	ড	ঢ	ণ	ত
থ	দ	ধ	ন	প	ফ	ব	ভ	ম
য	ৰ	ল	ৱ	শ	ষ	স	হ	ক্ষ
য়	ড়	ঢ়	ৎ	ংঁ	ঃঁ	ঁ		

(c)

কা	খি	গী	ঘু	ঙূ	চৃ	ছূ	জৈ	ঝো
ঞৌ	ট্য	ঠ্য	ডা	ঢী	ণী	তু	থূ	দৃ
ধে	র্ন	পো	ফৌ	ব্য	র্ভ	মা	যি	ৰি
লু	ৱূ	শৃ	ষে	সৈ	হো	ক্ষৌ	ড়্য	ঢ়
য়া	শু	গু	হু	ৰু	ৱৃ	কূ	কি	কী
কু	কূ	কৃ	কে	কৈ	কো	কৌ	র্ক	খা
খী	খু	খূ	খৃ	খে	খৈ	খো	খৌ	গা
গি	গূ	গৃ	গে	গৈ	গো	গৌ	র্গ	ঘা
ঘি	ঘী	ঘূ	ঘৃ	ঘে	ঘৈ	ঘো	ঘৌ	র্ঘ
মি	মী	মু	মূ	মৃ	মে	মৈ	মো	মৌ
ৰা	ৰী	ৰো	র্থ	ক্য				

Figure 1.3: Illustration of the basic units used for recognizing the Assamese character dataset. These comprise (a) 10 digits (b) 52 basic characters and (c) 95 modified characters.

Figure 1.4: List of basic units employed for the recognition of the Assamese words.

Figure 1.5: Depiction of several samples from the Assamese online handwriting character dataset.

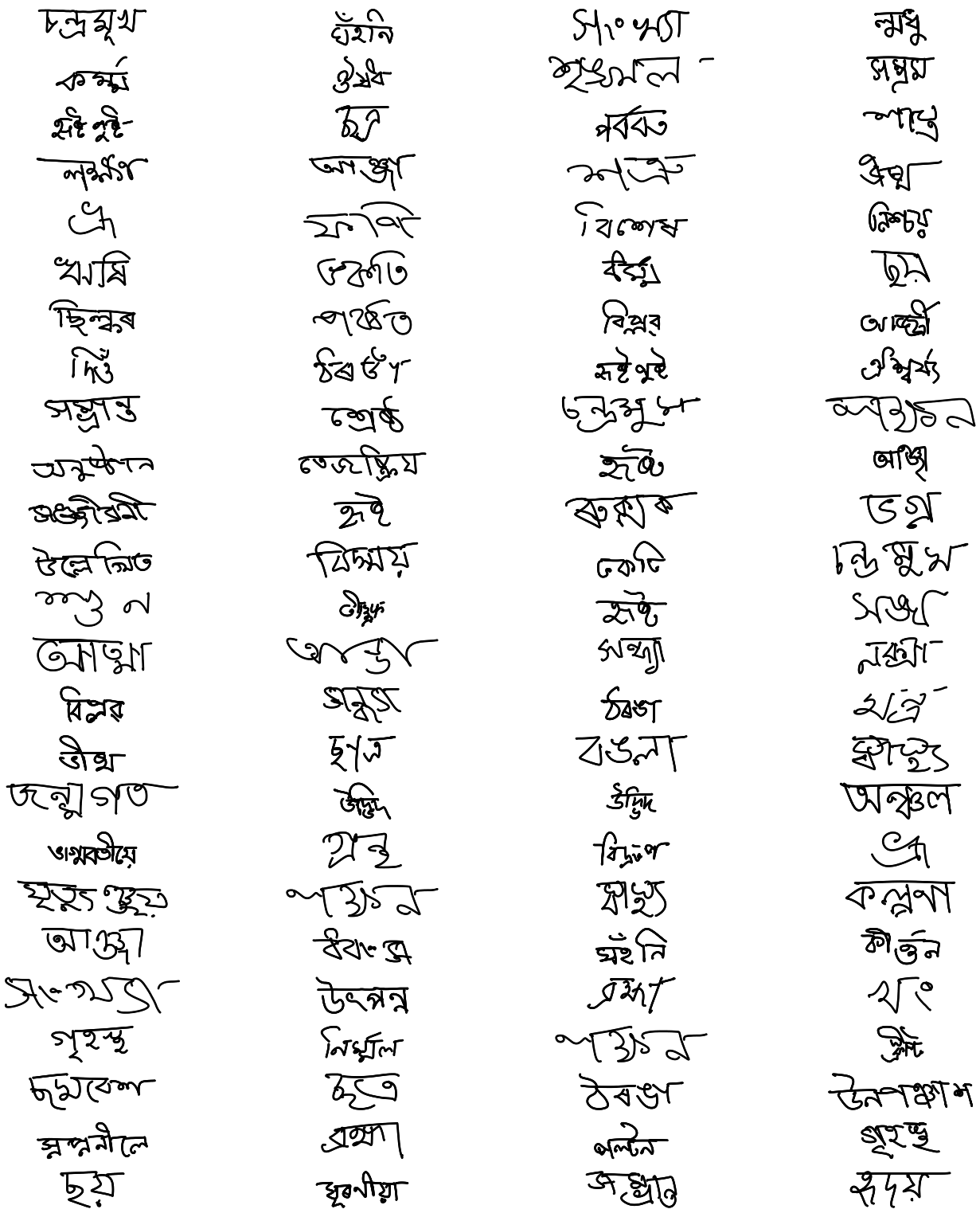


Figure 1.6: Presentation of online handwritten samples from the Assamese word dataset.

Table 1.2: Description of the classes and number of samples in the UNIPEN character and UNIPEN ICROW-03 word datasets.

UNIPEN character dataset					UNIPEN ICROW-03 word dataset				
Category	Class	Train	Validation	Test	Category	Class	Train	Validation	Test
Digit	10	8061	1992	4887					
Uppercase	26	13334	3330	8168	Word	884	7689	1334	3943
Lowercase	26	24929	6122	15352					

Figure 1.7: Examples of handwritten samples from the UNIPEN character dataset.

Figures 1.7 and 1.8 present some of the handwritten samples of the character and word data respectively.

It is to be noted that, with regard to the both the word datasets, we also generate the basic unit level data by implicitly segmenting the word samples of the training set. For this, a trained HMM-based system is used to forcibly align the handwritten word sample with the transcription, thereby resulting in the boundaries of the basic units. These boundaries are then utilized to generate the labeled basic unit data. The generated dataset is then used to train the basic unit classifier for the word recognition systems discussed in this thesis.

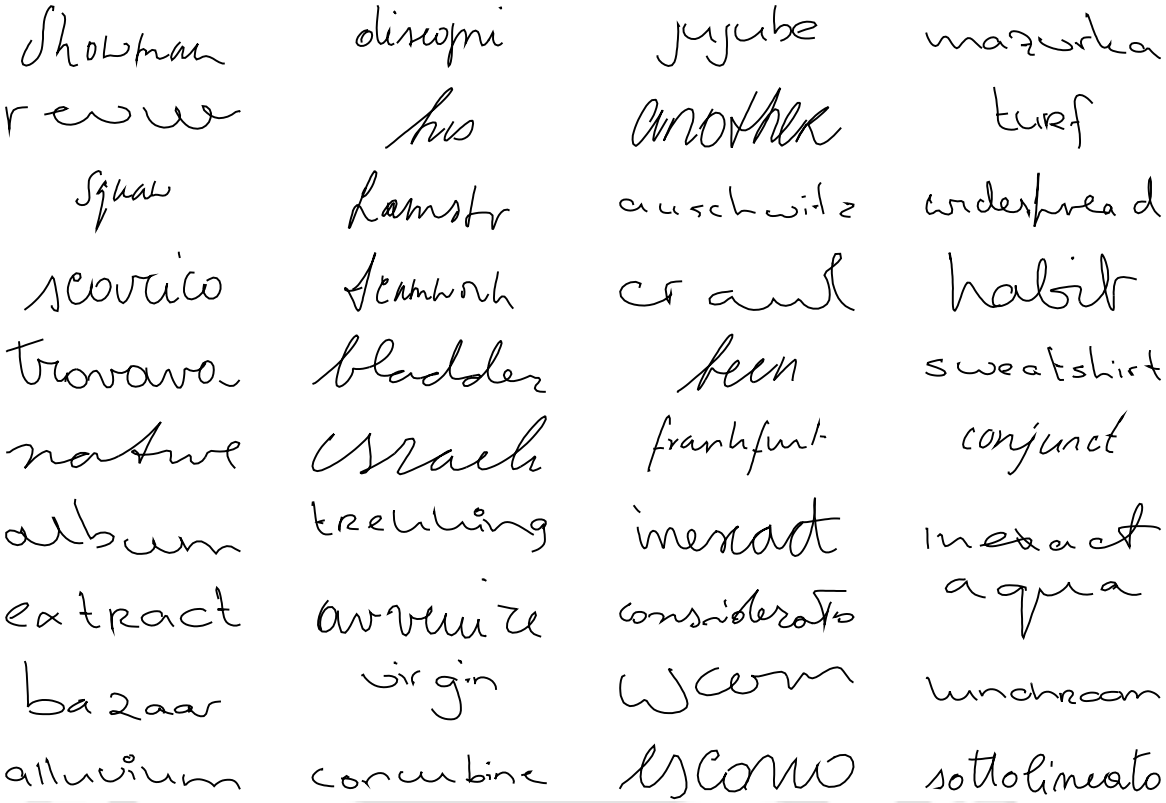


Figure 1.8: Illustration of online handwritten words from the UNIPEN ICROW-03 word dataset.

1.4.3 Lexicon creation

We also create lexicons of varying sizes W while performing the word recognition tasks on the two scripts. For generating a lexicon of a given size, we adopt the strategy presented in [11], where a W-word lexicon contain the top $W - N_w$ most frequent words selected from the corpus. Added to this, we also consider the N_w unique words used to collect the databases. It may be recalled from Section 1.4 that the value of N_w is 182 and 884 for the Assamese and UNIPEN ICROW-03 word database, respectively. The corpus used to generate the remaining $W - N_w$ Assamese and English words are selected from Assamese OCR [24] and "google-book-common-word" list [29] respectively.

We conclude our discussion by providing the following information in the Appendix to the thesis.

- List of Assamese and English words being collected in Sections A.1 and A.2
- Distribution of the length of the Assamese and English words (in terms of basic units) in Table A.1.

- Number of samples of each basic unit class from Tables A.2 to A.5 with respect to the character and word databases.

1.5 Summary

In this Chapter, we provided an overview of a typical online handwriting recognition system. Thereafter, the focus of the thesis was discussed with an elaboration of each of the four contributing Chapters of the thesis. The datasets used for the present study is also discussed.

In the next Chapter, we provide a detailed enumeration of works from recent literature within the scope of our work.

2

Literature Review

Contents

2.1 Introduction

The topic of online handwriting recognition has been a very active area for research exploration, since the past three decades, with the survey of works being well documented in [3, 30, 31]. As mentioned in the previous chapter, a large vocabulary online handwritten word recognition system can be developed by modeling the basic recognition unit of the script where the basic unit represents a set of patterns (e.g. characters/strokes) for which the handwriting models are created. The modeling of the basic unit recognition system is important and it is hoped that any improvement of the same will enhance the overall accuracy of word recognition systems.

The motivation of the thesis as outlined in Chapter 1 is on developing novel methods for online handwriting recognition system. In particular, we propose innovative strategies for the modeling of basic units with regard to aspects, namely feature extraction and recognition. Keeping this in perspective, in the remainder of this Chapter, we focus our review primarily in the paradigm of features and classification methods in Section 2.2 and 2.3 respectively.

2.2 Review of features

The features of online handwriting can be obtained directly from the online trace (x, y) coordinates of the data or from the offline image by converting the original input to an image. The different kinds of features used in the literature to describe the basic units are reviewed in this subsection. For the sake of better readability, they are categorized as follows

(i) Point based spatial features

(ii) Global shape features

(iii) Frequency domain features

(iv) Hand movement based features

(v) Offline features

In the following, we provide an overview of each of them by citing some representative works from the literature of online handwriting recognition.

2.2.1 Point-based features

The point-based features capture different kinds of shape attributes of online handwriting at each point of the trajectory. The simplest feature is that of the pre-processed (x, y)-coordinates itself that represent the spatial position of the basic unit pattern [1,32]. A number of features are derived in the literature by utilizing the (x, y) coordinates, which in turn can be divided into three groups on the basis of the following criterion [1, 2, 6, 7, 9, 10, 15, 32–35]:

(i) Features that are computed by considering successive points of the trace

(ii) Features that utilize the neighbouring information by considering a window in vicinity of a sample point

(iii) Features that provides only binary value of '0' or '1' by executing a logical statement.

The features such as derivatives, writing direction and curvature are some of the popular features from the first category. The first and second derivatives of the (x, y) coordinates are two widely used derived features and they measure the change and rate of change in the online trace [1, 7, 33, 34]. The writing direction describes the direction of pen movement at each point and is measured by using sine and cosine functions [9,10], while the curvature measures the angular difference between preceding and successive direction vectors [1,6,9]. The writing direction can also be represented by the Freeman chain code as used in the works [9, 35]. Beside these, additional features such as angular displacement [2], tangent slop angle [32], velocity [6, 15] have also been investigated in online handwriting recognition.

The popular features from the second category include those of linearity, curliness and aspect ratio features. These are computed by capturing the information from the vicinity of the sample point under question. The linearity and curliness characterize the degree of straightness of the handwriting trace in the vicinity [1, 7], while the aspect ratio feature determines the height-to-width ratio of the bounding box encapsulating the current sample point [7]. Apart from these, a shape context feature is proposed in [7], that measures the number of points belonging to each bin of a fan-shaped mask.

At this juncture, we also mention additional vicinity based features such as path-tangent angle, path velocity magnitude, log curvature radius and acceleration magnitude that have been found useful for basic unit description [9]. Though these features may not be as good in capturing the finer nuance of the pattern as compared to those from first category, they are less affected by the sudden changes

in the trajectory. Thus, when used in conjunction with the other features, they often provide added information that can help enhance the discrimination.

With regard to the third category, namely the binary features, the pen-down, hat and loop are used in many works. Essentially, these are computed by evaluating a logical statement at each sample point of the online trace. The pen-down feature at a sample point is encoded as 1 when the pen tip is in contact with the writing pad and 0 otherwise [1,7]. The hat feature is used to specify if the current (x, y) position is part of a delayed stroke [6,9]. Likewise, the loop feature is used to indicate whether the current point is part of a loop or not [9]. The binary features in general, provide complementary information and help in discriminating the basic unit patterns.

2.2.2 Global shape features

The global shape features describe the overall geometric characteristic of the basic unit by considering the entire pattern [1]. In this regard, there are works in literature [1,9,33,36,37] that utilize one or more of the following as a feature description at a global level.

- The number of strokes, stroke-crossings and dots in a basic unit
- The length of trajectory
- Width, height and aspect ratio of the entire basic unit
- Average curvature of each stroke in the basic unit

Apart from the above, global features are also computed by considering the number of points in the basic unit. For example, the number of points lying above and below the horizontal line passing through each point, and to the left and right of the vertical line are considered in the work [33]. Likewise, the number of point above the corpus line (ascender feature) and below baseline (descender features) are employed in [6].

Yet another direction for global feature extraction is the use of statistical quantities computed from the point-based features [1] such as mean, variance and x/y covariance. At this point, it is also worth mentioning the work [38], where a histogram constructed from the Chain Code encoding of the trace is utilized as features.

2.2.3 Frequency domain features

The frequency domain features for online handwriting are obtained by adopting different mathematical tools from the signal processing domain such as the Fourier Transform, Discrete Cosine Transform and Wavelet Transform. These methods are applied on the (x, y) coordinates of the online handwriting and their coefficients values are considered as features. Since these tools provide a frequency based representation of the sequence of (x, y) coordinates, the obtained features are referred to as the frequency domain features.

In the literature, frequency based methods are applied either along points of the trace, or separately on the x and y coordinates of the handwriting input. The authors in [39] used the discrete Fourier Transform (DFT) to extract the features. At each (x, y) coordinate point, the DFT is computed from the result of modulating the original point sequence by the Hamming window centered at the current point. Alternatively, the author in [40] applied Fourier descriptor separately on the x and y signals for computing the features.

The other two techniques namely, the Discrete Cosine Transform (DCT) [37, 41] and Wavelet Transform [42] are also computed separately on the x and y signals. In the DCT based approach, the first few coefficients corresponding to low frequency components are used for description. In the Wavelet-based approach, the Haar wavelet is mainly investigated in [42] where a third level Wavelet decomposition is carried out to extract the approximation coefficients as features.

2.2.4 Hand movement based features

The features for online handwriting are also computed from the handwriting generation model. The motivation stems from the fact that the conventional features (such as point-based) aim to represent the geometric shape of the handwriting. On the contrary, a generation model of handwriting tries to capture the movement based features of the trajectory. To compute the same, the values of the parameters are extracted from each of the handwritten samples by using models such as Beta-elliptical, sigma log-normal and sinusoidal.

The work of [43] employed the Beta-elliptical model of handwriting to represent the hand-movement and the resulting parameters were used as features. In another exploration [44], the handwriting trace is encoded as a sequence of strokes that in turn are characterized with the six sigma log-normal parameters. Likewise, the parameters of a sinusoidal model are also used for feature description of the

online data in [45].

2.2.5 Offline features

The features for online handwritten input can also be extracted from its corresponding offline bitmap image. To begin with, the intensity of the pixel values are directly used as features in [46]. Instead of considering each pixel value, the authors in [47] divided the image into zones and computed the pixel density in each of them for basic unit description. In yet another study [48], the authors used a sliding window technique to obtain frames from the image. Thereafter, for each of them, the number of foreground pixels and the values of the gradient are used for recognition of the pattern. Similar to the preceding study, the authors in [49] compute features such as center of gravity, moments, location of uppermost and lowermost black pixels from each frame. Likewise, the use of density, aspect ratio and character alignment ratio from the offline image data have also been explored in [36].

A few more offline features being utilized for online handwriting recognition include Hu's moments [50], water reservoir feature [51], stroke direction feature [1] and context map [10]. Very recently, the exploration of CNN features from the offline bitmap image of handwriting input [22] has also been advocated by researchers. The resulting description eliminates the necessity of deriving hand-crafted features for recognition.

We conclude the discussion of this section with a summary in Table 2.1 of the recent features for basic unit modeling in online handwriting. It is to be noted that the works being tabulated have been arranged chronologically.

2.3 Review of recognition approaches

In this section, we focus on presenting the different recognition strategies proposed in the literature for basic unit modeling. To begin with, the early explorations in online handwriting relied primarily on structural and rule based approaches, wherein a basic unit was described in an abstract fashion, without paying much attention to the irrelevant shape variations of the pattern [3, 57, 58]. These methods however presented difficulties with regard to the generation of reliable rules for the case of large vocabulary based recognition. Owing to this, they have not been employed in recent works of online handwriting recognition.

The majority of the explorations in the literature rely on statistical modeling approaches, that are based on either generative or discriminative modeling principles. In these techniques, the shape of the

Table 2.1: Summary of the various features used for basic unit modeling in online handwriting recognition. Note that the tabulated works have been arranged chronologically.

Study	Features
[Nguyen et. al. 2018], [7]	x and y difference of two adjacent coordinates, pen-up/pen-down, vicinity curvature, aspect, curliness, slope, linearity, offline context map and online shape context.
[Choudhury et. al., 2018], [45]	sinusoidal parameters.
[Keysers et. al., 2017], [1]	(x, y) and its derivatives, curvature, curliness, context map, bounding box, mean, variance, stroke crossing, dots, histogram of writing direction, water reservoir.
[Mukherjee et. al., 2017], [52]	polar coordinates, writing direction, curvature, aspect, velocity, y-coordinates, average square distance and Fourier coefficients.
[Du et. al., 2017], [22]	CNN features learn from offline image.
[Bhattacharya et. al., 2016], [33]	difference of initial and final respective x and y coordinates, basic unit aspect ratio, trajectory length, direction feature, number of points in various zone, the second order derivatives.
[Abdelaziz et. al., 2016], [9]	chain code, curliness, aspect ratio, writing direction, curvature, baseline and zones, loop, hat and few derived features.
[Yang et. al., 2015], [53]	Path signature and directional features.
[Samanta et. al., 2014], [2]	(x, y) coordinates, writing direction, angular displacement, vicinity aspect ratio, curliness, linearity and circular features.
[Du et. al., 2014], [54]	bottleneck features.
[Chowdhury et. al., 2013], [55]	directional and positional information based string, horizontal and vertical distance features.
[Bharath et. al., 2012], [11]	y-coordinates, writing direction, curvature, aspect, curliness, linearity and slope.
[KP. et. al., 2011], [42]	(x, y)-coordinates, writing direction, curvature and wavelet transform.
[Graves et. al., 2009], [6]	pen-up/pen-down, hat, velocity, normalized x and y coordinates, writing direction, curvature, aspect, slope, linearity, ascenders/descenders and context map.
[Kherallah et. al., 2008], [56]	Beta-elliptical features.

character classes are described by a statistical profile, where the associated parameters are estimated from the feature vectors of the training data.

The generative model based approaches evaluate the relation between the feature vectors of the input pattern and the trained models. The outcome of the recognition system is a set of likelihood scores - that quantify how probable the input pattern belongs to each of the classes. The hidden Markov models (HMMs) and Gaussian mixture models (GMMs) are typical examples that follow this paradigm. The discriminative model, on the other hand, determines the recognition score on the basis of the distance between the input pattern and the decision boundary of separation of all classes. It may be mentioned that the neural networks, such as SVM and MLP belong to this framework. Apart from the model based approaches, the distance-based Dynamic Time Warping (DTW) approach and its variants have also been used for online handwriting recognition.

In the following, we provide a survey of the relevant recognition approaches used for basic unit recognition.

2.3.1 Dynamic Time Warping

Dynamic time warping (DTW) is a technique of pattern matching that finds the ordered correspondence between two patterns. The distance is computed as a sum of the distances between the correspondences along the warping path. The recognition is carried out by measuring the similarity between the input basic unit test pattern and the reference patterns / templates of each class.

One of the earlier works with this approach is reported in [59], where one reference template for each basic unit is built by averaging the (x, y)-coordinates of the samples. To accommodate more variability of a class, the works reported in [46, 60–65] built multiple templates for each class, that are typically obtained from a clustering algorithm. Different to the preceding explorations, the works reported in [32, 34] built the reference template statistically by using the statistical DTW (SDTW) approach. Here, reference templates are not represented by a sequence of feature vectors, but rather by a sequence of quantities that are statistical in nature.

There are a few studies that report performance on word recognition with the DTW framework. However, it is to be borne in mind that the matching process is still performed at the basic unit level. One such work is reported in [66], where the system is developed by employing a explicit segmentation-based approach with a dictionary lookup table. The entries in the table contain the sequence of characters forming a word. In another exploration, the authors in [67] proposed a SDTW

based word recognition system, where the input data is first segmented to basic recognition units that are then recognized by matching. The result of this approach is a string of recognized basic unit labels, representing the output word.

2.3.2 Hidden Markov Model

Hidden Markov Models (HMMs) are one of the successful methods used for large vocabulary online handwriting recognition tasks. One reason for their popularity is due to their innate ability to model sequential data. For the scripts like Latin, where the main challenge is the segmentation of cursive writing into the constituent letters, the HMMs provide a solution by adopting an implicit-segmentation approach. The main idea is to model each basic unit separately by a single HMM. Thereafter, word models are built by concatenation of the basic unit HMMs. The word-HMM is then used to implicitly segment the input word into its constituent basic units.

The works addressing the use of HMM for online word recognition have been presented in [6, 9, 11, 13, 15, 45, 50, 68–74]. It is worth emphasizing here that for scripts like Chinese, the HMMs are often built at the stroke level. Such models are then connected to form a large network that, in a way describes the different stroke orders. In order to recognize input words independent of symbol writing order, a recurrent HMM with a "Bag-of-Symbols" representation and matching scheme is explored by the authors in [11] for two Indic scripts, namely Devanagari and Tamil.

Apart from the preceding applications, HMMs are also used to model the words by considering a holistic approach [2, 69, 75]. In these papers, the number of words considered in the lexicon are 211, 110 and 50 words, respectively. A few works also exist in the literature wherein the performance of HMMs built using sub-strokes [76, 77], strokes [8, 78–80], and characters [39, 81, 82] are reported for the recognition of basic unit patterns. Lastly we also mention explorations [73, 83] that utilize the HMMs to the problem of writer adaptation.

2.3.3 Neural networks

Artificial neural networks (ANN) have been found to be quite promising to the problem of online handwriting recognition. Techniques adopted in the literature include those of Multi Layer Perceptron (MLP), time-sequence interpretation approaches such as time delay neural network (TDNN) and recurrent neural network (RNN).

The proposals of online handwriting recognition using the MLP framework have been addressed

mainly at the character level for scripts such as English [84], Arabic [43] and Indic [35,41,85,86]. In the preceding papers, a generalized structure is considered wherein the hidden and output layers employ the sigmoid and soft-max activation function respectively. Further, the backpropagation algorithm is used for training the network with the computation of the various gradients by the gradient descent algorithm.

In TDNNs, words are represented as a sequence of basic units with each of them being modeled by one or more states [10,87–90]. Essentially, in these networks, a sliding window moves over the temporal sequence. The features extracted from the sample points within a window are fed to a feed-forward neural network. The activation level of each output node, one per class identity, gives the likelihood for the sequence of points in the sliding window belonging to that class. The end result of this operation is a sequence of generated likelihood values, that can be then used to find the best sequence of character identities using methods like Dynamic Time Warping [10, 90] and Viterbi search [89].

Recurrent neural networks (RNNs) are models that consist of a self-connected hidden layer [6, 91]. One of their main advantages is in their ability to access contextual information, an important asset for handwriting recognition. Typically, a Connectionist temporal classification objective function is employed in the network, that performs a mapping from the input sequence data to the sequence of output labels. This in turn removes the necessity of segmenting the data a-priori. Improved variants of RNNs such as the bidirectional long short-term memory (BLSTM) architecture have also been proposed to provide access to long-range input context in both directions. In particular, the BLSTM with CTC has been adapted to handwriting recognition systems in [7, 92, 93].

2.3.4 Support Vector Machine

The SVM is a binary classifier that searches for the optimal hyper-plane to maximize the margin between the training samples of the two classes [94]. For the case of multiple classes, the recognition is realized by combining several two-class SVMs by employing techniques such as one-versus-all or one-versus-one [95]. To handle the case of non-separable data, a so called kernel function is incorporated in the SVM formulation. Some of the commonly used kernels include linear, polynomial and radial basis function (RBF). These provide an implicit mapping of data to a high dimensional space to ensure that they become linearly separable The works from the literature of online handwriting recognition adopting the aforementioned kernels approach can be found in [8, 16, 45, 78, 96–99].

Apart from the above studies, few works have also been reported to accommodate temporal vari-

ability of online handwriting into the SVM framework, by proposing kernels that can work on variable length data [100–102]. The authors in [102] employed a Gaussian DTW (GDTW) kernel while the exploration in [101] proposed a piecewise linear interpolation technique to generate various SVM kernels for character recognition task. The performance of the same is compared with the GDTW kernel and a notable improvement in the performance is achieved.

2.4 Summary

To summarize, we provided a detailed review of works from the perspective of basic unit modeling in online handwriting recognition. We focused our survey primarily on feature extraction techniques and recognition approaches.

In the next Chapter, we present the details of the first contribution of this thesis - namely, the reevaluation of the decision of a classifier by adopting a Hidden Markov Model (HMM) framework.

3

Discriminative HMM States for Recognition

Contents

3.1 Introduction

The task of online handwriting recognition often becomes challenging due to the presence of similar shaped basic unit classes which differ only in a small portion of the trace [16–18]. This is owing to the fact that the classifier working on features at the first level, at times, fails to capture finer nuances that distinguish these basic units. One way to alleviate this drawback is to perform a reevaluation of the classifier decision by reassigning the label of the input pattern under question with the aim of reducing the confusion between similar looking basic units. This will be the focus of the current as well as the following contributing Chapter of the thesis.

Literature has many proposals to reduce the confusions of similar shaped basic unit classes [17, 18, 21, 103–106]. The broad pipeline of these works is in adopting a two-stage system wherein a pairwise trained second-stage classifier refines the top-2 outputs of the first-stage classifier. The authors in [18] present a scheme to detect confusion pairs and subsequently attempt to classify them by employing a two-stage system with an SVM classifier. Likewise, the exploration of [106] investigate the use of a discriminative classifier in both the stages by utilizing the MLP and SVM. Generative and discriminative classifiers have also been considered in a two-stage framework, thereby ensuring that the benefits from both approaches are adequately captured [17, 105]. The use of the k-nearest neighbors in conjunction with an SVM has been investigated in [103].

In this chapter, we propose a strategy to reevaluate the decision of an HMM-based basic unit recognition system, that in turn alleviates the necessity of a different second-stage classifier. The choice of using the HMM classifier can be attributed to the fact that it has been quite widely used in several prior works to model the basic units in online handwriting recognition.

Typically, in a conventional HMM-based system, a model is constructed separately for each basic unit by using the Baum-Welch estimation algorithm [19]. Whenever a test basic unit is to be recognized, we first compute the log-likelihood scores from each of the class-specific HMMs. Thereafter, the class corresponding to the highest score is assigned to the sample.

With regard to the single-stage HMM framework, we demonstrate in this chapter, that, at times, the sole dependence on log-likelihood scores may not be effective in capturing the finer nuances of the online trace that discriminates similar shape patterns / basic units. As a circumvention to this issue, we propose to analyze the HMM states corresponding to the top-2 confusion classes with the objective of identifying a subset that can help provide cues to discriminate them better. Subsequent

to the identification of the so-called 'discriminative states', we compare their log-likelihood scores with regard to the HMMs of the two confused classes and accordingly make the final decision.

Said in another way, our main contribution is coming up with a strategy that can automatically detect the discriminate states between the HMMs of two basic units, that are most likely to get confused. We demonstrate that the utility of employing likelihood scores computed over discriminative states assists in reducing the confusions between similar looking basic units.

We also extend our proposal to develop a large vocabulary word recognition system employing the HMM framework. Given a test word sample, first, we obtain the top-M most probable words from a baseline HMM-based system and then reevaluate the scores of each of them for refining the recognition decision. The likelihood score for each of the basic units segmented implicitly from a probable word choice is revised by taking into consideration the discriminative states. The lexicon entry with the highest average revised score is taken as the recognized word.

The performance of the proposed system of employing discriminative states is evaluated on two databases: the locally collected Assamese character and word datasets, as well as the publicly available English UNIPEN character and UNIPEN-ICROW-03 word datasets. The results obtained suggest an improvement over the conventional HMM-based systems across each of the datasets.

Based on the preceding discussion, the following are the research highlights of this chapter.

- Enhancement of the classification ability of an HMM-based system by exploring the discriminative HMM states.
- Refining the top-2 most relevant outputs in a single-stage classification framework.
- Development of basic unit and large vocabulary word recognition systems with the proposed approach.

The remainder of this Chapter is organized as follows: The details of the baseline HMM system for the basic unit and word recognition tasks are presented in Section 3.2. Section 3.3 elucidates the proposed technique used for detecting the set of discriminative states between the trained HMMs of the similar looking basic units. The utility of the obtained states for recognition is demonstrated in Section 3.4. This is followed by a discussion in Section 3.5 on how our proposal can be extended for recognizing online handwritten words. The result of the baseline and proposed systems are presented in Section 3.6 and 3.7, respectively. In Section 3.8, we summarize our contributions.

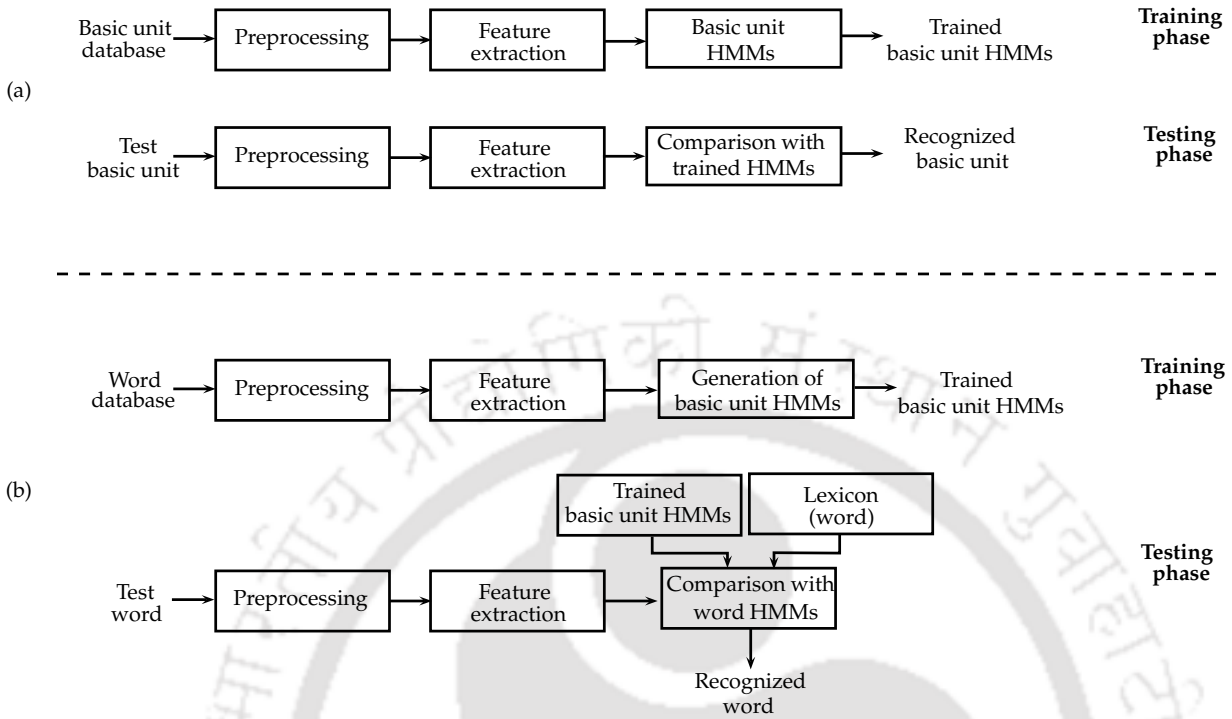


Figure 3.1: Sub-figures (a) and (b) depict the block schematic of the baseline HMM-based basic unit and word recognition systems.

3.2 Baseline system

In this section, we provide the details of the baseline basic unit and word recognition systems against which all the proposals in this thesis are compared. In our work, we use the HMM framework for the same.

Fig. 3.1 depicts the overall framework of the baseline basic unit and word recognition systems. The input sample is first passed through a preprocessing module, that comprises the operations of smoothing, size normalization and resampling. Thereafter, the preprocessed data is fed to the feature extractor module to obtain the description of the data.

For the basic unit recognition system, the classification is performed by evaluating the log-likelihood score of each model for the sequence of feature vectors. The category corresponding to the highest score is assigned to the test data. For the word recognition system, we concatenate the basic unit HMMs corresponding to the transcription of the basic units present in the word. It is to be noted that each entry in the lexicon is modeled separately by a word HMM, that is obtained from the process of concatenation. During the testing phase, the sequence of feature vectors of the data is forcibly aligned against all the word HMMs of the lexicon. The model with the maximum value of the likelihood score

is selected, with its corresponding word being assigned to the handwritten test sample.

In the following, we present the details of preprocessing, feature set and classifier used to develop the baseline systems.

3.2.1 Preprocessing

As discussed in Chapter 1, the online handwriting captured from the digitizer is a sequence of (x, y) coordinates with pen-up and pen-down events. The preprocessing steps are applied to compensate for variations in scale and time among the samples of the same class. The procedures employed are smoothing, size normalization and resampling. Smoothing reduces the amount of high-frequency noise in the input data resulting from either the capturing device or due to jitters in writing. To smooth the data, a moving average filter with a window size of 3 points is applied to each stroke.

The size normalization and resampling steps are different for the basic unit and word samples. The variation in the size of a basic unit sample is eliminated by normalizing the y coordinates of the pattern to $[0, 1]$ range while preserving the aspect ratio. Thereafter, the resampling step is performed to make the basic unit samples in to a fixed arc length with q uniformly sampled points.

To resample the basic unit sample, first, the total trajectory length is obtained by calculating the cumulative distance. Next, the trajectory length is divided by the required number of intervals to obtain uniform spacing between successive points in the resampled data. This results in a sequence of new points along the trajectory that are determined by using linear interpolation. For a basic unit sample written with multiple strokes, each stroke is resampled separately ensuring that the number of resampled points in each stroke is proportional to its trajectory length. After preprocessing, each basic unit sample is represented as a sequence of (x, y) coordinates:

$$\{(x_1, y_1), (x_2, y_2), ..., (x_i, y_i), ..., (x_q, y_q)\} \tag{3.1}$$

It is to be noted that for our experiments, the samples of Assamese and UNIPEN character dataset are resampled to 100 and 50 points, respectively.

To normalize the size of a word sample, first, the baseline and corpus line for the word are determined [15]. The region between these two lines is normalized to the [0,1] range. The remaining portion is normalized proportionately while preserving the aspect ratio. In addition, we resample the word data to obtain a constant spacing between two consecutive points [15].

3.2.2 Features

The feature extractor module computes a set of features at each of the preprocessed (x_i, y_i) coordinates of the online trace. Our features have been adapted from the works of [10, 34] and are enumerated below:

- **Coordinate:** The preprocessed x_i and y_i coordinates.
- **First order derivative:** The first derivative of x and y coordinates, defined as,

$$x'(i) = \frac{\sum_{j=1}^{2} j(x(i+j) - x(i-j))}{2\sum_{j=1}^{2} j^2}$$
$$y'(i) = \frac{\sum_{j=1}^{2} j(y(i+j) - y(i-j))}{2\sum_{j=1}^{2} j^2} \tag{3.2}$$

- **Second order derivative:** The second derivative of x and y coordinates, defined as,

$$x''(i) = \frac{\sum_{j=1}^{2} j(x'(i+j) - x'(i-j))}{2\sum_{j=1}^{2} j^2}$$
$$y''(i) = \frac{\sum_{j=1}^{2} j(y'(i+j) - y'(i-j))}{2\sum_{j=1}^{2} j^2} \tag{3.3}$$

  Here, the $x'(.)$ and $y'(.)$ values are computed using Equation (3.2).

- **Writing direction:** The writing direction at each (x_i, y_i) point, described by cos θ_w and sin θ_w:

$$cos\ \theta_w(i) = \frac{\Delta x(i)}{\Delta s(i)}$$
$$sin\ \theta_w(i) = \frac{\Delta y(i)}{\Delta s(i)} \tag{3.4}$$

  where

$$\begin{aligned} \Delta x(i) &= x(i+1) - x(i-1) \\ \Delta y(i) &= y(i+1) - y(i-1) \\ \Delta s(i) &= \sqrt{\Delta x^2(i) + \Delta y^2(i)} \end{aligned}$$

- **Curvature:** The curvature value at each (x_i, y_i) point, described by cos θ_c and sin θ_c:

$$\begin{aligned} cos\ \theta_c(i) &= cos\ \theta_w(i-1) \times cos\ \theta_w(i+1) + sin\ \theta_w(i-1) \times sin\ \theta_w(i+1) \\ sin\ \theta_c(i) &= cos\ \theta_w(i-1) \times sin\ \theta_w(i+1) - sin\ \theta_w(i-1) \times cos\ \theta_w(i+1) \end{aligned} \tag{3.5}$$

- **Linearity:** This feature is computed by employing a set of points in the vicinity of (x_i, y_i). In our implementation, we consider five points namely $\{(x_{i-2}, y_{i-2}), (x_{i-1}, y_{i-1}), (x_i, y_i), (x_{i+1}, y_{i+1}), (x_{i+2}, y_{i+2})\}$ and accordingly define

$$LN(i) = \frac{1}{N} \times \sum_j d_j^2 \tag{3.6}$$

  Here d_j corresponds to the perpendicular distance of the j^{th} vicinity point to the straight line joining the first and last points, *viz* $(x_{(i-2)}, y_{(i-2)})$ and $(x_{(i+2)}, y_{(i+2)})$.

- **Aspect ratio:** The vicinity aspect ratio $AR(i)$ at each (x_i, y_i) point is described by

$$AR(i) = \frac{\Delta y(i) - \Delta x(i)}{\Delta y(i) + \Delta x(i)} \tag{3.7}$$

  where

$$\begin{aligned} \Delta x(i) &= x(i+2) - x(i-2) \\ \Delta y(i) &= y(i+2) - y(i-2) \end{aligned} \tag{3.8}$$

- **Slope:** This feature is computed as the slope of the straight line joining the first and last vicinity points, *viz* $(x_{(i-2)}, y_{(i-2)})$ and $(x_{(i+2)}, y_{(i+2)})$.

$$SL(i) = \frac{\Delta y(i)}{\Delta x(i)} \tag{3.9}$$

- **Context map:** To calculate this feature, first, the sample is transformed into a bitmap image having a height of 8 pixels and width, that is fixed by preserving the aspect ratio. Assume, (x_i, y_i) falls into the pixel (j, k) of the bitmap image. A 3×3 context map B_f, centered at (j, k) pixel is slid along the trajectory of the pen. The total number of black points falling in B_f is taken as the feature.

Each of the derived features are normalized by using the z-norm method [107]. Let the sequence of feature vectors corresponding to a basic unit sample having q-points be denoted as:

$$\mathbf{O} = [\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_q] \tag{3.10}$$

Each of the feature vectors in $\mathbf{O}$ can in turn be written as $\mathbf{o}_i = [o_{i1}\ o_{i2}\ ...\ o_{id}]^T \in \mathbb{R}^d$. Here, d corresponds to the number of features extracted at each (x, y) point of the sample (14 in our case).

It may be mentioned, however that for word recognition, a 16-dimensional feature vector is extracted at each (x, y) point of the word sample. These comprise the aforementioned 14 features with two additional features, namely ascender and descender [6].

3.2.3 Classifier

To build the handwriting recognition systems, we have employed the hidden Markov model (HMM) framework with N_s states $\{s_i\}_{i=1}^{N_s}$ for each basic unit. By associating a state sequence $q_1, q_2, ...q_T$ to the observations $[\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_T]$, we can characterize the HMM by $\lambda = [\Pi, A, B]$. By definition,

- $\Pi = \{\pi_i\}$ is the initial state distribution

$$\pi_i = P(q_1 = s_i) \qquad i = 1, 2, ...N_s \tag{3.11}$$

- $A = \{a_{ij}\}$ is the state transition matrix

$$a_{ij} = P(q_{t+1} = s_j | q_t = s_i) \qquad i, j = 1, 2, ...N_s \tag{3.12}$$

- $B = \{b_i(\mathbf{o}_t)\}$ is the probability distribution of observation feature vector $\mathbf{o}_t$ in the state s_i.

  Typically, the probability distribution is a Gaussian mixture model (GMM) comprising $\mathcal{M}$ components.

To develop the basic unit recognition system having C basic unit classes, we build C HMMs denoted by λ_p, $p = [1, 2, ..., C]$ by using the HTK Toolbox [108]. A left-to-right topology is employed to model the transition of the states. The HMM parameters i.e. the number of states and mixture components in the GMM are optimized during training. The Baum-Welch algorithm is used to train the models. For recognition, we compute the log-likelihood score of each model λ_p corresponding to the test sample $\mathbf{O}$. The class / basic unit label is decided based on the highest log-likelihood score and is given by

$$\hat{c} = \arg\max_{1 \le p \le C} p(\mathbf{O}|\lambda_p) \tag{3.13}$$

In the case of word recognition, the HMMs corresponding to the basic units present in the transcription of a word are concatenated to form the model. In other words, each word entry in the lexicon is represented by an HMM obtained by the process of concatenating its constituent basic unit HMMs.

To begin with, the initial models are trained using the flat-start initialization, with each basic unit of a word being assigned an equal number of feature vectors. Thereafter, through an iterative process

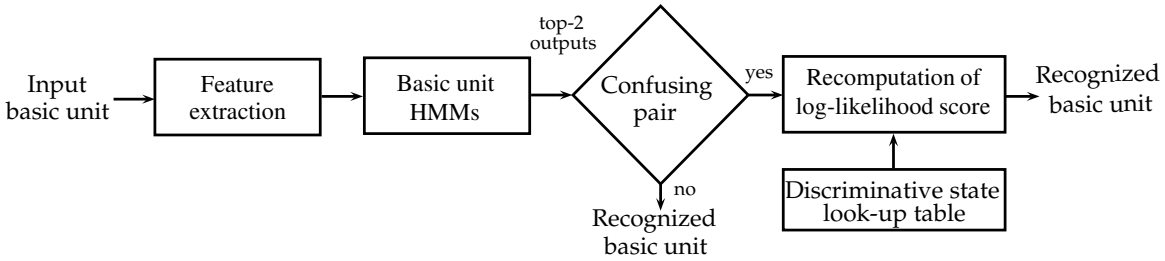


Figure 3.2: A block diagram of the proposed discriminative state based basic unit recognition system.

of training and boundary re-estimation of the basic units using the Baum-Welch algorithm, each of the word HMMs get modeled better. The iteration terminates when a convergence criterion is met.

For the recognition of a word, first, we compute the log-likelihood score of each lexicon word for the given sequence of feature vectors $\mathbf{O}$ by using the Viterbi algorithm described in [19]. The word $\hat{w}$ corresponding to the highest score among all the words in the lexicon is declared as the output.

$$\hat{w} = \underset{1 \leq p \leq W}{\arg\max}\ P(\mathbf{O}|\hat{\lambda}_p) \tag{3.14}$$

Here, $P(\mathbf{O}|\hat{\lambda}_p)$ is computed from the respective word HMM $\hat{\lambda}_p$.

In the following three Sections, we present the proposed technique to improve the baseline HMM system. More specifically, we describe a strategy of exploiting the discriminative states between the HMMs of confused basic units to improve the recognition performance.

3.3 Exploration of discriminative states

Fig. 3.2 presents the block schematic of the proposed HMM-based basic unit recognition system. The input sample is passed through the feature extraction module. It converts the data to a sequence of feature vectors which is then fed to an HMM-based classifier for recognition. We select the top-2 most relevant classes from the output. Now, if these form a confusion pair, we analyze the discriminative states of their respective HMMs to refine the outputs by revising the likelihoods.

Suppose (c_1, c_2) represent a confusion pair. To identify the discriminative states, we consider their HMMs λ_{c_1} and λ_{c_2}. Recall from the theory of HMMs, that the parameter B captures the data distribution through the use of GMMs in each state. Accordingly, in our approach, we attempt to exploit the same for determining the discriminative state of the two HMMs.

Given a trained model λ_p, we can think of it as sequence of GMMs

$$B^{\lambda_p} = \{b_1^{\lambda_p}, b_2^{\lambda_p}, ..., b_{N_s}^{\lambda_p}\} \tag{3.15}$$

where N_s is the number of states in λ_p. Formally, for the basic unit pair (c_1, c_2), we can write their corresponding sequence of GMMs as

$$\begin{aligned} B^{\lambda_{c_1}} &= \{b_1^{\lambda_{c_1}}, b_2^{\lambda_{c_1}}, ..., b_{N_s}^{\lambda_{c_1}}\} \\ B^{\lambda_{c_2}} &= \{b_1^{\lambda_{c_2}}, b_2^{\lambda_{c_2}}, ..., b_{N_s}^{\lambda_{c_2}}\} \end{aligned} \tag{3.16}$$

Our goal is to formulate a distance metric between the respective b^{λ_p} (i.e. the GMMs) of the two basic units, that can help quantify the dissimilarity between their respective states. Considering that both $b^{\lambda_{c_1}}$ and $b^{\lambda_{c_2}}$ are a sequence of Gaussian distributions, we employ the Earth Movers Distance (EMD) [109]. In particular, the dissimilarity value is calculated for each state

$$state\_dism^{(\lambda_{c_1}, \lambda_{c_2})}(i) = EMD(b_i^{\lambda_{c_1}}, b_i^{\lambda_{c_2}}) \quad i = 1, 2, \ .... \ , N_s \tag{3.17}$$

The variable $state\_dism$ is a vector that contains N_s dissimilarity value corresponding to the N_s states of λ_{c_1} and λ_{c_2}.

A smaller value of $state\_dism(i)$ represents that the respective states have similar characteristic while a higher value implies that the corresponding states are more discriminative. Accordingly, a set of states (with state dissimilarity value above a threshold) surrounding the i^{th} state with the highest dissimilarity value of $state\_dism$ can be considered. Thus, for a basic unit pair (c_1, c_2), we can write the set of discriminative states as

$$DS^{(c_1, c_2)} = [s_a \ s_b], \quad 1 \leq s_a < s_b \leq N_s \tag{3.18}$$

where s_a and s_b are the starting and ending indices of the discriminative states respectively [1].

Though the preceding discussions have been based on analyzing the HMM states of a specific confusion pair (c_1, c_2), the same procedure does hold applicability for any two similar looking basic units. This leads us to generate a look-up table, the entries of which capture the discriminative state information between all confusion pairs that are identified from the confusion matrix of the baseline system.

[1] In the present discussion, we assume an HMM framework with left-to-right topology.

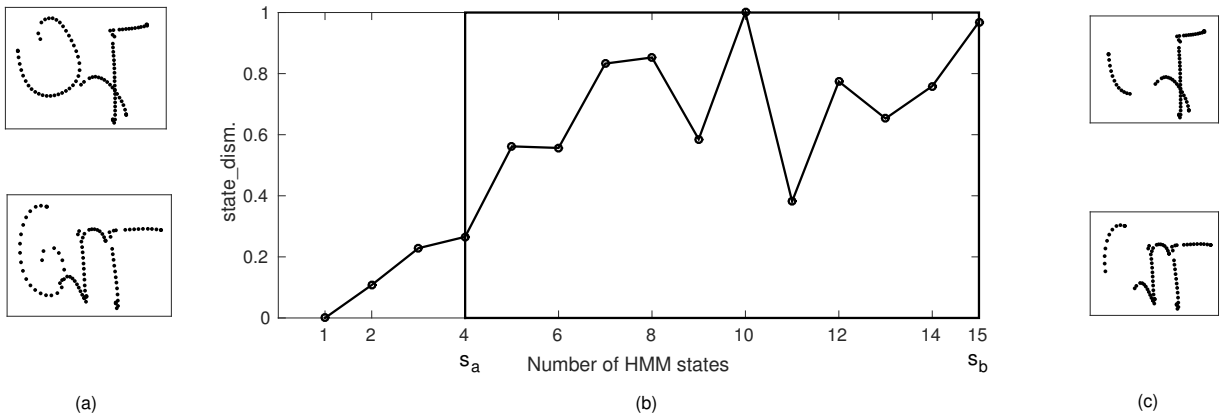


Figure 3.3: (a) A representative sample of a basic unit অ (/o/)and আ (/a/) that form a confusion pair. (b) Degree of dissimilarity *state_dism* value computed by applying the Earth Movers Distance between the distributions of each of the individual HMM states. The identified discriminative states surrounding the highest dissimilarity value is marked by the rectangle where s_a and s_b denotes the starting and ending indices respectively. (c) The parts of the trace as obtained by selecting the discriminative states.

As an illustration, we present in Fig. 3.3 (a) a basic unit pair from Assamese script, namely অ (/o/) and আ (/a/). The patterns are modeled using 15 HMM states with the observation emission of each state following a GMM of 20 Gaussians. The identified discriminative states surrounding the highest dissimilarity value (with state dissimilarity threshold value of 0.3) is marked by the rectangle (in sub-figure(b)), where s_a and s_b denote the starting and ending index respectively. By using this information, the parts of the trace differentiating অ (/o/) and আ (/a/) are illustrated in sub-figure (c).

3.4 Recognition methodology

During the time of recognition, for a confusion pair, the discriminative states of their corresponding HMMs are fetched from the look-up table. Thereafter, the log-likelihood scores for both these classes are recomputed by considering only the retrieved states. The label of the basic unit corresponding to the maximum log-likelihood score is assigned to the test input data.

Formally, given a basic unit pair (c_1, c_2), consider that the starting and ending indices of the discriminative state $DS^{(c_1,c_2)}$ are s_a and s_b. The revised log-likelihood score $\mathcal{L}_p$ of the p^{th} basic unit can be calculated as:

$$\mathcal{L}_p = \frac{\phi_{t_v}(s_b) - \phi_{t_u}(s_a)}{t_v - t_u}, \quad p = c_1 \ , \ c_2 \tag{3.19}$$

Here, t_u and t_v are the time instances in the optimal state sequence where state s_a begins and state

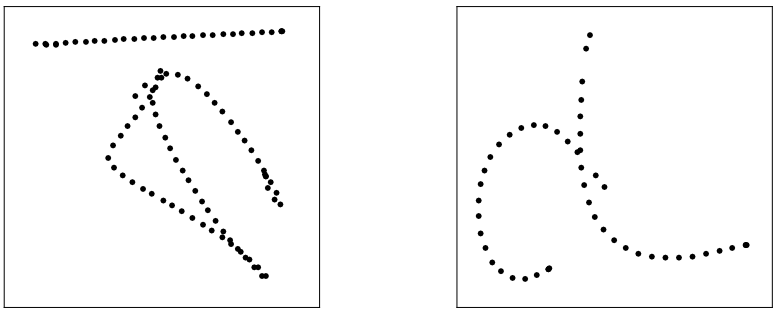


Figure 3.4: The online trace of (a) an Assamese character ক (/k/) and (b) English lowercase letter '*d*'. These patterns get wrongly identified by the base-line system to ফ (/ph/) and English lowercase letter '*a*'. However, by utilizing the likelihood scores from the discriminative HMM states of the top-2 outputs, they get corrected to the correct label.

s_b ends for the confused basic unit under consideration. The term $\phi_t(j)$ stores the highest possible probability along a single path from observation point o_1 to o_t with state at o_t being j and is given by

$$\phi_t(j) \quad = \quad \underset{1\leq i\leq N}{max}[\phi_{t-1}(i) + log\ a_{ij}] + log[b_j(\mathbf{o}_t)]$$

Let $\mathcal{L}_{c_1}$ and $\mathcal{L}_{c_2}$ correspond to the revised log-likelihood scores of basic unit c_1 and c_2, respectively obtained by employing Equation (3.19). The assignment of the handwritten sample to the class is based on the following criterion

$$\widehat{c'} = \underset{p=c_1,c_2}{\arg\max}\ \mathcal{L}_p \tag{3.20}$$

To demonstrate the utility of the discriminative states in recognition, we consider the online trace of an Assamese character ক (/k/) in Fig. 3.4(a). When the sequence of feature vectors of this pattern are fed to the baseline HMM system, log-likelihood scores of -578.88 and -675.40 are assigned to the top-2 symbols ফ (/ph/) and ক respectively. Clearly, on the basis of the higher log-likelihood, the test sample gets mis-classified to ফ. As an alleviation to resolving this issue, we revise the aforementioned scores by relying solely on the information of the discriminative states of the two models. Post reevaluation, we obtain a log-likelihood score of -249.15 for ক, that is higher when compared to -295.76 for the symbol ফ. This in turns ensures that the depicted test sample indeed gets correctly recognized as ক in the proposed reevaluation framework.

An explanation similar to the above is also applicable to the pattern in Fig. 3.4(b). This gets recognized as character '*a*' by the baseline HMM system with '*d*' being the second best choice. However, on revising the scores based on the information of the discriminative states, the pattern gets corrected

Table 3.1: Recognition score (log-likelihood value) of the baseline and proposed systems in classifying the test samples of Fig. 3.4. For this experiment, the Assamese and English characters are modeled with 15 and 11 HMM states respectively with 20 Gaussians in the GMM. The choice of these parameters are based on the minimum error rate performance, that will be discussed in Section 3.6. The state dissimilarity threshold value is set to 0.3.

Test sample	Class	Baseline system	Proposed system
Assamese basic unit of Fig. 3.4(a)	ক (/k/)	-675.40	**-249.15**
	ফ (/ph/)	-578.88	-295.76
English basic unit of Fig. 3.4(b)	*d*	-530.13	**-170.57**
	a	-425.33	-179.34

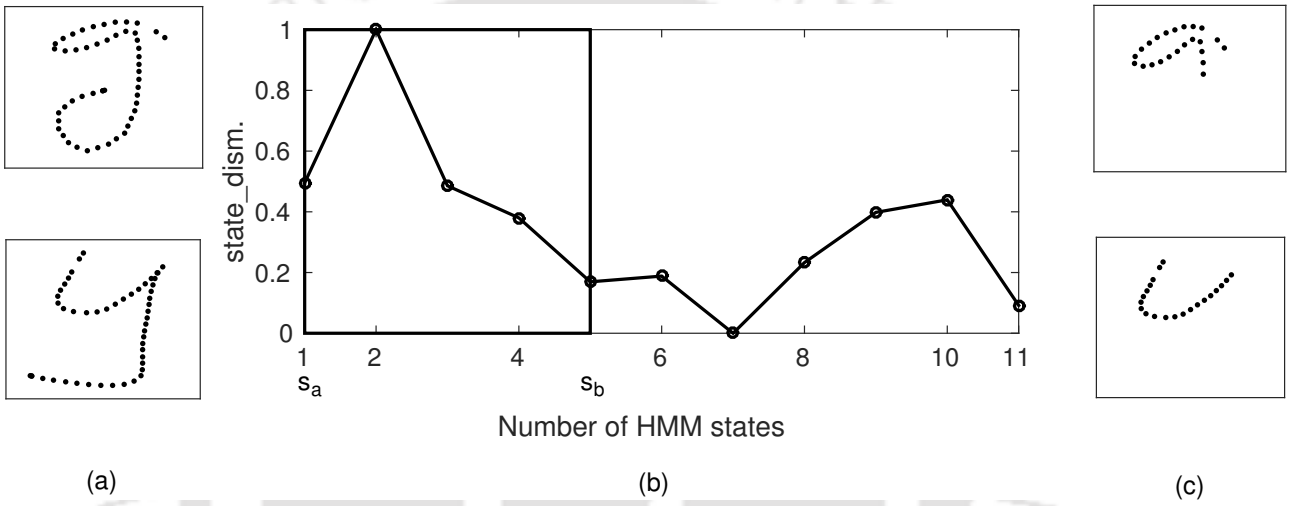


Figure 3.5: Another illustration depicting the finer nuances of the trace between two confusion basic units '*g*' and '*y*', as obtained by exploring the information of the discriminative HMM states.

to '*d*' post reevaluation.

For completeness, in Table 3.1, we summarize the respective log-likelihood scores of the baseline and proposed systems for the two patterns shown in Fig. 3.4.

We conclude by providing another illustration in Fig. 3.5. Here, the finer nuances of the trace between two confusion basic units '*g*' and '*y*' are decided, based on information of the discriminative HMM states.

3.5 Extension to word recognition

Fig. 3.6 presents a pictorial overview of the word recognition system that employs the reevaluation strategy based on discriminative state analysis. To begin with, the input online word is fed to the feature extraction module that computes the feature vectors at each point of the entire trace. Thereafter, the baseline HMM-based word recognition framework of Section 3.2 is used to compute

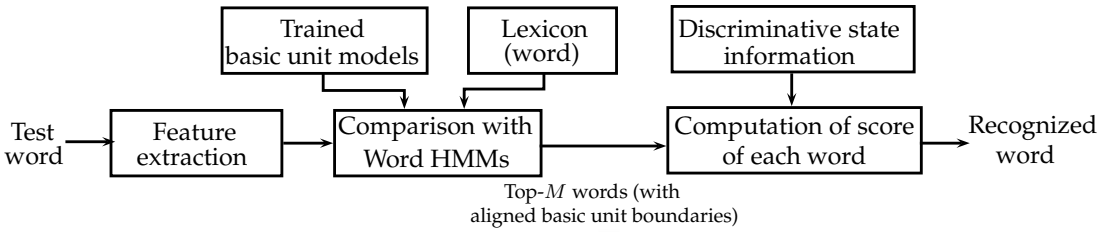


Figure 3.6: A block diagram of the proposed word recognition system that employs the information of the discriminative states between the HMMs of frequently confused basic unit pairs.

the log-likelihood scores on each of the different word HMM models built from a lexicon. On the basis of the obtained scores, the top-M most probable words are chosen for further processing, where the top-1 choice is the most likely word transcription of the online data.

Following the generation of the probable words, we reevaluate the scores of each of them for refining the recognition decision, if necessary. It is worth reemphasizing here that the reevaluation strategy is performed at the basic units making up the word. Keeping this in perspective, the implicit segmentation of the basic units for each top-M lexicon words is performed by employing the baseline HMM system [2]. Subsequent to obtaining the segments / basic units of a probable word choice, the likelihood score for each of them is revised by taking into consideration the discriminative states of the baseline HMM as discussed in the following.

For sake of illustration, consider the i^{th} probable lexicon word W_i to comprise N_i basic units, namely $\{c_1^i, c_2^i, ..., c_{N_i}^i\}$. When the online input sample is forcibly aligned with this word, the baseline HMM divides it to N_i segments. For the reevaluation of the p^{th} segment with the basic unit label c_p^i $(1 \leq p \leq N_i)$, its most competing class from the look-up table is first selected. Thereafter, the set of discriminative states between the chosen classes are employed to revise the likelihood score $\mathcal{L}_p^i$ by employing Equation (3.19).

On the whole, we obtain N_i revised log-likelihood scores $\{\mathcal{L}_p^i\}_{p=1}^{N_i}$ corresponding to each of the basic units constituting the word sample W_i. Accordingly, we can now compute the revised average likelihood score of the word W_i with

$$\bar{\mathcal{L}}_i = \frac{\sum_{p=1}^{N_i} \mathcal{L}_p^i}{N_i} \tag{3.21}$$

where $\mathcal{L}_1^i$, $\mathcal{L}_2^i$,..,$\mathcal{L}_{N_i}^i$ are the individual revised log-likelihood scores of the N_i basic units as obtained

[2]For obtaining the boundaries, we apply the Viterbi algorithm on the word-HMM and analyze the sequence of states.

Table 3.2: Error rate (%) of the baseline HMM system for the Assamese and English basic unit recognition tasks.

Dataset	Validation set	Test set
Assamese digit	1.23	1.13
Assamese basic character	3.65	3.87
Assamese modified character	3.76	4.00
English digit	1.25	1.13
Uppercase letter	3.90	3.67
Lowercase letter	6.40	6.82

from the discriminative state based reevaluation analysis for the word W_i. The lexicon word entry with the highest average likelihood score is assigned to the handwritten test sample. Mathematically, we can now write

$$\widehat{w'} = \arg\max_{1 \leq i \leq M} \bar{\mathcal{L}}_i \tag{3.22}$$

where $\widehat{w'}$ is the revised word obtained from the proposed system.

3.6 Result of the baseline HMM system

In this Section, we evaluate the performance of the baseline HMM system described in Section 3.2.3 for the basic unit and word recognition tasks. We use databases outlined in Tables 1.1 and 1.2 for our experimentation. For each of them, the training set is used to create the models while the validation and test sets are used to optimize and test the system respectively.

The performance is judged in terms of percentage of wrong classifications defined by

$$\text{Error rate} = 100 \times (1 - \frac{\text{\# of correct prediction}}{\text{\# of samples tested}}) \tag{3.23}$$

It may be recalled here that the HMM for each basic unit is trained on the feature set enumerated in Section 3.2.2. In particular, we vary the number of states and mixture components in the GMM and select the values leading to the minimum error rates for each of the datasets. The best performances are obtained for 15 and 11 states with regard to the Assamese and English character datasets. A set of 20 Gaussians are employed in the GMM for modeling each state. In Table 3.2, we present the results of the basic unit recognition system on the validation and test sets of English and Assamese characters. The list of confusion pairs are shown in Tables 3.3 and 3.4 for the two scripts.

As a next experiment, we evaluate the performance of the baseline HMM system at the word level

Table 3.3: List of similar looking confusion basic unit pairs in Assamese character dataset.

Dataset	Confusion pair
Digit	(৩,৬), (৫,৬),
Basic character	(অ,আ), (ই,ঈ), (ই,হ), (উ,ঊ), (উ,ড), (ঋ,খ), (এ,ঐ), (ক,ফ), (খ,ঘ), (খ,থ), (গ,ণ), (ঘ,য), (ঙ,ড), (ঙ,ভ), (চ,ঢ), (ড,ত), (ড,ভ), (ত,ভ), (ধ,ব), (ন,ল), (ফ,য), (ম,স), (য,ষ), (ষ,ষ)
Modified character	(কা,ক্য), (খি,যি), (খি,ঘি), (গী,ণী), (মা,খা), (মা,ঘা), (কু,কৃ), (কু,কূ), (কূ,কৃ), (কী,ঘী), (খু,খূ), (খু,খৃ), (খূ,খৃ), (খু,মু), (ঘু,ঘূ), (ঘূ,ঘৃ), (মু,মূ), (মু,মৃ), (মূ,মৃ), (শু,গু), (ৰু,ৰূ))

Table 3.4: List of similar looking confusion basic unit pairs in English character dataset.

Dataset	Confusion pair
Digit	*(1,7), (4,9)*
Uppercase	*(A,H), (C,G), (D,O), (D,P), (K,R), (O,Q), (O,U), (U,V), (V,Y)*
Lowercase	*(a,d), (a,q), (a,u), (b,k), (c,e), (e,l), (f,t), (g,y), (h,k), (h,n), (r,k), (r,n)*

for different sizes of the lexicon. Based on the transcription, each word in the lexicon is represented by an HMM, that is constructed from the concatenation of the basic unit HMMs. Considering the minimum error rate obtained on the validation data,

- The number of HMM states for modeling each of the basic units of the Assamese and English word datasets were optimized to 15 and 11 respectively
- The number of mixtures in the GMM were optimized to 20.

Table 3.5 presents the error rate of the word recognition systems, evaluated on the validation and test samples of the Assamese and English datasets. The performance for different lexicon sizes is also given for both datasets.

3.7 Result of proposed system

In this subsection, we present the performance of the proposed basic unit and word recognition systems. Recall that our framework employs the discriminant state information to reduce the degree of confusion between basic unit patterns that are likely to be confused by the baseline HMM classifier.

3.7.1 Basic unit recognition

To begin with, we study the performance of the proposed system by varying the value of the state dissimilarity threshold, discussed in Section 3.3. Table 3.6 presents the results obtained on the

Table 3.5: Error rate (in %) of the baseline HMM-based word recognition system evaluated on the validation and test sets of Assamese and English word datasets.

Dataset	Lexicon	Validation set	Test set
Assamese word	5000	12.64	22.92
	10000	14.85	26.13
	20000	17.13	28.86
English word	5000	28.63	26.27
	10000	33.13	29.65
	20000	36.80	33.52

Table 3.6: Error rate (in %) of the proposed basic unit recognition system with varying values of the state dissimilarity threshold on the different **validation sets**. The performance of the baseline HMM system (without reevaluation) is also reported for comparison.

Dataset	Baseline system	Proposed system				
		state dissimilarity threshold value				
		0.1	0.2	0.3	0.4	0.5
Assamese digit	1.23	1.23	0.99	0.99	0.99	1.48
Assamese basic character	3.65	3.36	3.19	2.85	3.36	4.22
Assamese modified character	3.76	3.47	3.36	3.05	3.62	4.47
English digit	1.25	1.20	1.15	1.00	1.10	1.40
Uppercase letter	3.90	3.78	3.60	3.45	3.84	4.65
Lowercase letter	6.40	6.10	5.92	5.39	6.10	7.23

validation set of different datasets. For comparison, the performance of the baseline HMM system (without reevaluation) is also reported.

In general, a lower state dissimilarity threshold value selects more states while a higher one results in fewer discriminative states during the reevaluation of the confused basic units. Accordingly, with increasing threshold values from 0.1 to 0.3, we get a reduction in the error rate. Any further increments in the value increase the error rate. This trend is owing to the selection of only a few states, that may not be adequate enough for reevaluation. From Table 3.6, it can be observed that a state dissimilarity threshold value of 0.3 results in the best recognition performance on the different validation sets across both the scripts.

Next, we evaluate our system on the test sets of the different databases with state dissimilarity threshold value of 0.3 and report the performance in Table 3.7. It can be seen that there is a notable improvement in the performance of the proposed system when compared to the baseline on both the validation and test sets. For the digit recognition tasks, since only a few of the 10 digit classes form confusion pairs, the corresponding improvement in performance is less. However, with regard to the

Table 3.7: Error rate (in %) of the baseline and proposed basic unit recognition systems on the **test sets**.

Dataset	Baseline system	Proposed system
Assamese digit	1.13	0.75
Assamese basic character	3.87	3.13
Assamese modified character	4.00	3.15
English digit	1.13	0.86
Uppercase letter	3.67	3.15
Lowercase letter	6.82	6.18

basic character, modified character and lowercase letter datasets, the number of confusion pairs are relatively higher. It is worth mentioning that, for these samples, the consideration of discriminative HMM states is found to enhance the recognition performance, thus bolstering our contribution further. Table 3.8 enumerates the improvement achieved for some of the encountered confusion pairs in the test set by the proposed system.

We conclude this subsection by considering how our proposed system fares with prior works reported on the Assamese and UNIPEN character datasets and presented in Table 3.9. It is important to take note, that in general, a direct one to one comparison is not possible. This is owing to the possible use of different features, classifier architectures, training and test protocols. With regard to the entries in sub-table (a) for Assamese dataset, the two-stage system in [110] is based on combination of HMM and SVM, where the SVM reevaluates the decision of the HMM classifier, if it happens to be a confused pair. The work of [25] considers a combination strategy that selects final output either from HMM or SVM classifier by utilizing the a-priori information obtained from confusion matrix analysis. Referring to Table 3.9(b) for English character, the systems in these references are based on DTW in [32], Online Scanning n-tuple Classifier (OnSNT) in [27], ANN in [1], and HMM in [8]. Based on the entries, a comparable performance has been achieved by employing the proposed system.

Table 3.8: Some encountered confusion pairs and their frequency of occurrence in the test set with the baseline and proposed systems. The % of improvement achieved by the proposed system is also reported.

Confusion pair	# of test samples	# of confusions of baseline system	# of confusions of proposed system	% of improvement
(৩,৬)	216	4	2	50.0
(অ,আ)	163	12	8	33.3
(খূ,খৃ)	239	13	9	30.7
(কা,ক্য)	228	14	10	28.5
(মূ,মৃ)	260	17	12	29.4
(ক,ফ)	170	13	10	23.0
(গী,ণী)	174	18	14	22.2
(ম,স)	143	19	15	21.0
(খ,থ)	161	10	8	20.0
(O,Q)	740	19	13	31.5
(K,R)	633	13	9	30.7
(4,9)	994	7	5	28.5
(a,u)	1612	39	29	25.6
(A,H)	635	11	8	27.2
(U,V)	538	16	12	25.0
(g,y)	896	36	28	22.2
(r,v)	1142	39	32	17.9
(r,n)	1832	43	37	13.9

Table 3.9: Performance comparison (error rate in %) with other works reported on (a) Assamese character and (b) UNIPEN character dataset.

(a)

Method	Digit	Basic character	Modified character
Two-stage system [110]	1.20	-	4.30
HMM and SVM combination [25]	1.70	-	-
Proposed system	0.75	3.13	3.15

(b)

Method	Digit	Uppercase	Lowercase
DTW [32]	2.90	7.20	9.30
OnSNT [27]	1.10	4.30	7.90
ANN [1]	0.80	3.10	5.10
HMM [8]	1.73	-	-
Proposed system	0.86	3.15	6.18

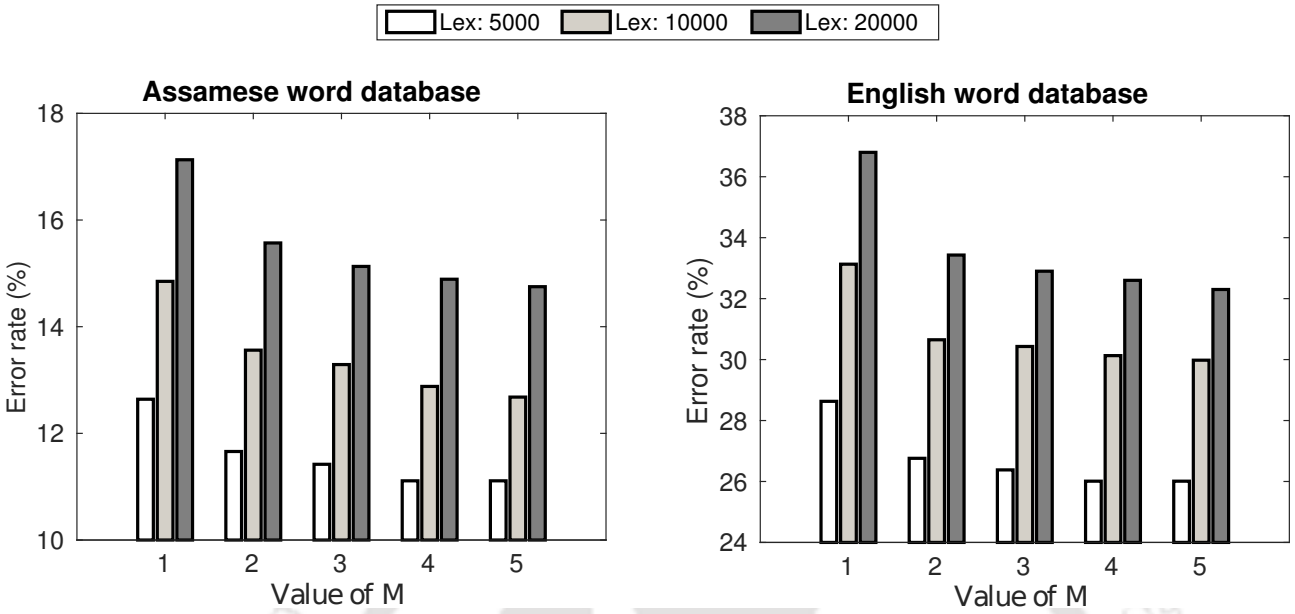


Figure 3.7: Error rate (in %) of the proposed word recognition system for varying number of top-M words on the **validation sets** of (a) Assamese and (b) English word datasets.

3.7.2 Word recognition

Fig. 3.7 presents the error rate of the proposed word recognition system for varying top-M words on the validation set across three different lexicon sizes of 5000, 10000 and 20000 words. It is to be noted that with increasing value of M, the probability that the correct word label appears in the top-M choices also increases. Accordingly, by reevaluating the score of more words, the system performance improves. However, it can be observed from the plots that, beyond M=4, the change in the performance becomes quite marginal for all sizes of the lexicon.

The performance of the proposed word recognition system on the test sets of Assamese and English databases is presented in Table 3.10. It can be seen that, with increasing size of the lexicon, the error rate increases. This can be attributed to the fact that larger lexicons can result in a greater degree of structural similarity among the words. This, in turn, leads to generating models with similar characteristics, that are susceptible to a higher number of mis-classifications, especially with challenging test samples.

However, at the same time, it is important to realize that since the proposed system works on refining the top-M words that may have structural similarities, the degradation in performance is not as severe when compared to the baseline HMM. This is indeed a commendable aspect worth observing from the results obtained on the word datasets of English and Assamese.

Table 3.10: Error rate (in %) of the baseline and proposed word recognition system on the **test sets**.

Lexicon size	Assamese		English	
	Baseline	Proposed	Baseline	Proposed
5000	22.92	20.57	26.27	23.89
10000	26.13	23.72	29.65	26.83
20000	28.86	25.84	33.52	29.34

Table 3.11: Average computational time (in second) for recognition of Assamese words of different length, for the baseline and proposed systems.

Word length	Baseline system			Proposed system		
	Lexicon size			Lexicon size		
	5000	10000	20000	5000	10000	20000
2	8.67	19.35	40.03	8.73	19.40	40.10
4	11.92	27.30	56.55	12.01	27.39	56.64
6	19.64	45.27	95.04	19.75	45.38	95.16
8	24.22	56.67	117.85	24.37	56.83	118.02

As an additional experiment, we present in Tables 3.11 and 3.12, the average computational time (in seconds) for recognition of Assamese and English words of different lengths. This analysis is done for both the baseline and proposed systems.

Last but not least, we see how our approach fares with the performance of the work [111] for the English dataset in Table 3.13. The results are mentioned for different sizes of the lexicon. Here again, we note that the proposed method outperforms with improvements of 5.27%, 6.15% and 8.07% for the lexicon sizes of 5000, 10000, and 20000 words, respectively.

Table 3.12: Average computational time (in second) for recognition of English words of different length, for the baseline and proposed systems.

Word length	Baseline system			Proposed system		
	Lexicon size			Lexicon size		
	5000	10000	20000	5000	10000	20000
2	4.20	9.21	18.88	4.24	9.24	18.92
6	11.34	25.48	47.94	11.39	25.54	48.01
10	17.54	37.94	75.15	17.66	38.08	75.31
14	24.78	52.75	100.86	24.99	53.02	101.10

Table 3.13: Performance comparison of the proposed system with the literature reported work on English word database.

Method	Lexicon		
	5000	10000	20000
SVM [111]	29.16	32.98	37.41
Proposed system	23.89	26.83	29.34

3.8 Summary

In this Chapter, we proposed a strategy to reevaluate the decision of an HMM-based basic unit recognition system. We demonstrated that, at times, the sole dependence of log-likelihood score of the HMM states may not be effective in capturing the finer nuances of the online trace that discriminate similar basic unit patterns. To alleviate this issue, we proposed to make the decision by revising the likelihood scores from the information obtained from the discriminative states. The efficacy of the proposal is demonstrated for the basic unit and word recognition tasks evaluated on Assamese and English databases and an improvement in performance is achieved.

In the following Chapter, we present yet another proposal for reevaluating the decision of the classifier by focusing on the selection of the discriminant region.

4

Discriminative Regions in Basic Units

Contents

4.1 Introduction

In the literature, the task of classifying similar shaped basic unit classes is largely addressed by adopting a two-stage system [17, 18, 21, 103–106]. In this framework, a pairwise trained second-stage classifier refines the top-2 outputs of the first-stage classifier. Typically, the extracted features fed to the second-stage pertain to parts of the pattern that present fine structural differences between the similar looking basic units. In the context of the present chapter, we refer to such parts as 'Discriminative Region'.

In the literature, various techniques have been reported for automatic identification of discriminative regions in a confusion pair. Almost all the strategies published so far are related to the domain of offline handwriting recognition. A selective partitioning algorithm is presented in [112], which splits the image of the pattern into four sectors. Thereafter, by employing an exhaustive search, the discriminative region of a confused character pair is identified. Xu *et al.* in [113] suggested the use of average symmetric uncertainty algorithm to select the discriminative region from offline handwriting data. Likewise in the work [114], Leung *et al.* propose Fisher's linear discriminant to address the same problem.

However, with regard to online handwriting recognition, we came across only one work [16] where the authors have presented the 'DTW-DDH algorithm' for obtaining the discriminative region. This method has been developed in the context of online Tamil word recognition.

In this Chapter, we present a novel strategy to detect the discriminative region in a basic unit pair for online handwriting recognition. The proposed technique splits the trace of the handwritten samples of a class into several segments and describes their distribution by the parameters of the k-means clustering. Thereafter, the Dynamic Time Warping (DTW) algorithm is used to match the statistical characterizations, associated with the basic units that form a confusion pair. The distances in the cost matrix along the optimal warping path are analyzed to select the discriminative region. During the DTW cost matrix generation, it is to be emphasized that the confusion basic units being matched are represented by a sequence of distributions (rather than a sequence of feature vectors).

With regard to the classification strategy, we propose a single-stage framework that takes into consideration the discriminative region extracted between the basic units. In the conventional setup, a two-stage framework is employed wherein, the top-2 outputs of the first classifier are fed to the second classifier for refining the output. Thereafter, in the second stage, features are extracted from

the discriminative region to resolve the confusions of similar basic units. Different to it, we propose an ensemble of $\binom{C}{2}$ classifiers [1], corresponding to all possible pairs of basic unit classes by using a one-vs-one strategy. The classifiers are in turn trained by utilizing the features extracted from the discriminative region pertaining to each combination of basic units. Thereafter, at the time of recognition, a majority voting scheme is applied to the ensemble for assignment of the identity to the test basic unit. An advantage of this approach is that by employing parallel processing, the scores from all the classifiers can be computed at one go. This makes the run time of the proposed system similar to that of the baseline while exploiting the advantage of discriminative region-based processing.

We also develop an HMM-based large vocabulary word recognition system by incorporating the discriminant region-based processing, for reevaluating the word recognition output. The overall framework of this system is similar to that described in Section 3.5. The only difference, however, is that we employ the feature vectors from the extracted discriminative region. For an input word, first, the boundaries of the basic units for each top-M word are obtained from the baseline HMM system via the implicit segmentation process. Subsequent to obtaining the segmented basic units of a probable word choice, the scores of each of them are revised by passing through a subset of $(C-1)$ classifiers in the ensemble.

Based on the preceding discussions, the following are the research highlights of this work.

- Proposal of a novel strategy for selecting discriminative region in a pair of basic unit patterns.
- Proposal of a discriminative region-based single-stage classification framework for online handwriting recognition.
- Development of a basic unit and large vocabulary word recognition systems using the aforementioned proposed technique.

The chapter is organized in the following order. Section 4.2 presents the details of the proposed discriminative region selection technique. This is followed by an elucidation of the single-stage framework in Section 4.3 with its extension to word recognition in Section 4.4. Experimental evaluation of our proposal is demonstrated in Section 4.5. Finally, the Section 4.6 summarizes the proposals made in this chapter.

[1] We assume here that C represents the number of basic unit classes in the recognition system.

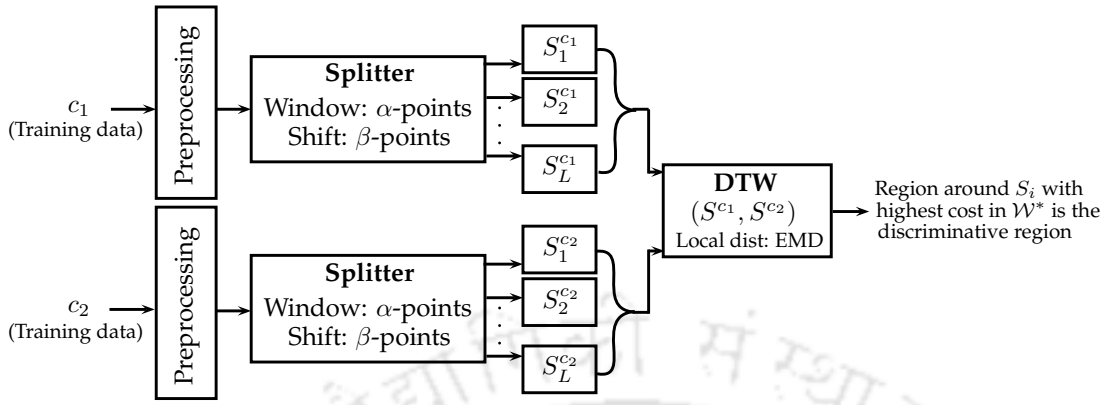


Figure 4.1: Pictorial overview of the proposed discriminative region selection methodology for a pair of basic unit patterns (c_1, c_2). For each basic unit class, we first segment the samples to L data segments and represent their distribution by the parameters of the k-means clustering approach. The DTW algorithm is then applied to match the statistical characterization by considering the EMD as a distance measure. Thereafter, an analysis of the distances obtained along the warping path is used to determine the discriminative region $DR^{(c_1,c_2)}$.

4.2 Proposed methodology

Fig. 4.1 depicts the block schematic of the proposed methodology. As we shall see in this section, our technique relies on the estimation of statistical information of the confusion basic units, that are matched by the DTW algorithm. The identification of the discriminative region is based upon analyzing the costs along the warping path, resulting from the match.

Recall that a pre-processed online handwritten sample can be represented as a sequence of q two dimensional (x, y) coordinates, namely $\{(x_1, y_1), ..., (x_q, y_q)\}$. As a first step, we partition the sample into L data segments described by employing a "splitter" with a window of α-points and a shift of β-points. Assuming that there are n_p samples available for training in the p^{th} basic unit class c_p, we can denote s_i^j to represent the set of points belonging to the i^{th} segment of the j^{th} sample. In our proposal, we consider analyzing parts of the trace corresponding to a given segment / section. Keeping this in perspective, we can now write the data segment S_i of the basic unit class as a pooling of the i^{th} segments from across all the training samples. Mathematically speaking, we have

$$S_i = [s_i^1 \quad s_i^2 \quad ... \quad s_i^{n_p}] \tag{4.1}$$

A straightforward extension to the above is in representing a basic unit class c_p as a collection of the L segments defined as $[S_1^{c_p}, ..., S_L^{c_p}]$. As a next step, we model the (x, y) samples in each segment $S_i^{c_p}$

by $R_i^{c_p}$, a tuple of statistical quantities obtained by employing the k-means clustering. In other words,

$$R_i^{c_p} = [w_k^{c_p}, \boldsymbol{\mu}_k^{c_p}, \boldsymbol{\Sigma}_k^{c_p}] \qquad i = 1, ..., L \qquad k = 1, ..., K \tag{4.2}$$

where $w_k^{c_p} \in \mathbb{R}^{[1\times1]}$, $\boldsymbol{\mu}_k^{c_p} \in \mathbb{R}^{[2\times1]}$, and $\boldsymbol{\Sigma}_k^{c_p} \in \mathbb{R}^{[2\times2]}$ are the weight, mean vector and covariance matrix of k^{th} cluster.

In order to obtain the discriminant region $DR^{(c_1,c_2)}$, we consider a confusing pair (c_1, c_2) with their associated statistical representations

$$\begin{aligned} R^{c_1} &= [R_1^{c_1}, ..., R_L^{c_1}] \\ R^{c_2} &= [R_1^{c_2}, ..., R_L^{c_2}] \end{aligned} \tag{4.3}$$

We can in fact view the above two sequences of R_i as a set of two time series signals having equal length L.

In this work, we attempt at matching R^{c_1} and R^{c_2} with a dynamic programming approach such as DTW [2]. We compute the cost matrix of size $[L, L]$ where the $(i, j)^{th}$ element measures the dissimilarity $d(i, j)$ between the i^{th} segment of c_1 class (represented by the tuple $R_i^{c_1}$) to the j^{th} segment of c_2 class (represented by $R_j^{c_2}$). However, measuring dissimilarity between two distributions ($R_i^{c_1}$ and $R_j^{c_2}$) is not straightforward as in the case of a vector space. Moreover, the distributions may have different lengths, owing to the number of clusters that can vary based on the pattern complexity.

In order to compute the dissimilarity between $R_i^{c_1}$ and $R_j^{c_2}$ we use the Earth Movers Distance (EMD) [20]. The EMD has two advantages. Firstly, it can be applied between two variable length representation of distributions which may be suitable for online handwriting. Secondly, the EMD computes all pairwise distances across the clusters, thus ensuring that both the size and the location of all the data are utilized in calculating the distance.

Subsequent to determining the DTW score, the obtained optimal warping path $\mathcal{W}^*$ is analyzed from the cost matrix. Without loss of generality, a small value of $d(i, j)$ along $\mathcal{W}^*$ corresponds to similar segments in the basic unit pair (c_1, c_2). Likewise, it can be inferred that a higher value, on the other hand, suggests a greater degree of distinction. Accordingly, by exploiting the value of $d(i, j)$ in the cost matrix, the discriminative region $DR^{(c_1,c_2)}$ can be identified.

We compute the indices (i^*, j^*) corresponding to the maximum value of the dissimilarity on the

[2]For the implementation of DTW, we rely on the code provided in the MATLAB central exchange

warping path $\mathcal{W}^*$. Subsequent to it, the selection of discriminant region is determined on the values of i^* and j^* as follows

- For the case where $i^* = j^*$, the part of the trace corresponding to the $(i^* - r)^{th}$ to $(i^* + r)^{th}$ segments are considered to be the $DR^{(c_1,c_2)}$.
- When $i^* < j^*$, the set of (x, y) points from the $(i^* - r)^{th}$ to $(j^* + r)^{th}$ segments form the $DR^{(c_1,c_2)}$.
- When $i^* > j^*$, the segments from $(j^* - r)^{th}$ to $(i^* + r)^{th}$ constitute the $DR^{(c_1,c_2)}$.

The value of r is chosen empirically for each of the datasets [3] .

4.2.1 Illustration

To highlight the ability of the proposed methodology in selecting the discriminative region of a confusion pair, we consider a few basic unit pairs (খূ, খৃ), (কী, ঘী), (U, W), and (g, y) for the present discussion. The first two panels in each row of Fig. 4.2 depicts the online trace of the confused patterns. The third and forth panels thereof highlight the discriminative region selected by the proposed technique.

For each of the four illustrations, we consider windows of length $\alpha = 6$ with a shift $\beta = 3$ to generate the segments from the online trace. Thereafter, the statistical characterizations of the same are obtained by employing a cluster size of $K = 4$. The details of the tuning of these parameters is given in Section 4.5.1, as part of our experiments.

We also visualize the two-dimensional distributions of the aforementioned confusing pairs by extracting feature vectors from the entire basic unit pattern and the discriminative region pattern respectively. In general, for a recognition task, a feature space should exhibit small intra-class variability while enhancing the separation between different classes. Thus, feature visualization in a way can serve as a tool to depict the discrimination ability of our approach. Accordingly, we visualize the different feature spaces by mapping them from 14-dimensional to the 2-D plane by employing the t-SNE algorithm [115]. The results of the same are presented in Fig. 4.3. From the plots, we see that the feature space associated with the discriminative region pattern is more separable when compared to the entire trace of the confused characters / basic units.

[3] In our work, we select a value of $r = 5$ across the databases.

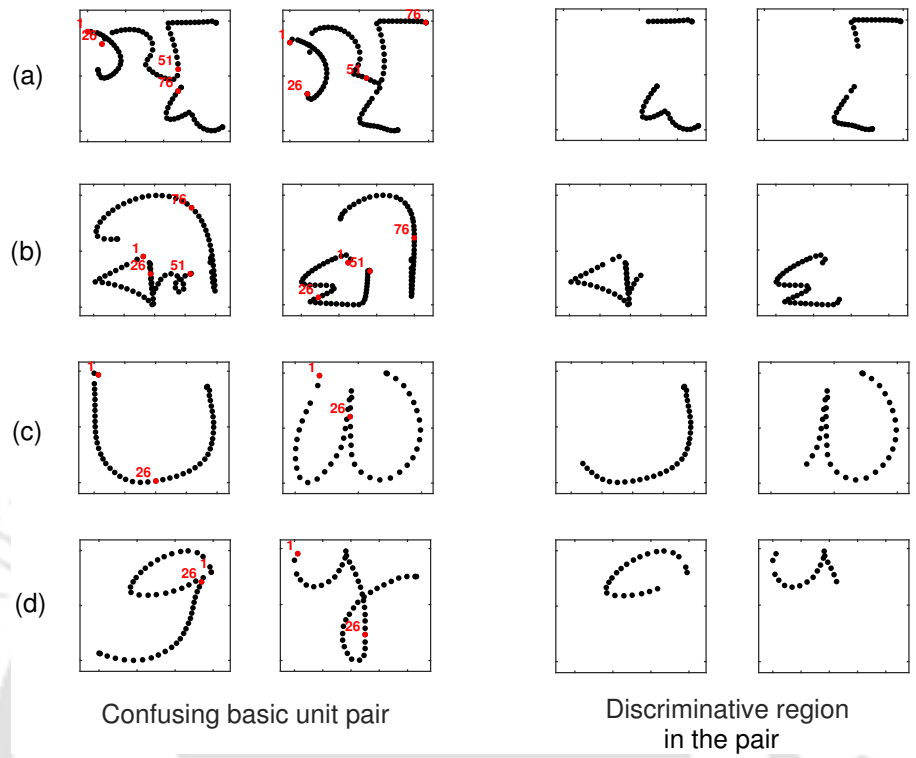


Figure 4.2: (a) (From left) First and second panels represent the whole pattern of খূ and খূ respectively. The next two panels represent the discriminative region pattern of (খূ , খূ) that are extracted using the proposed technique. The sub-plots (b)-(d) represent the same as in (a) for basic unit pair (কী, ঘী), (U, W), and (g, y) respectively.

4.3 Discriminative region-based single-stage system

Fig. 4.4 illustrates the proposed single stage system employing the discriminative region-based processing. Let $\{c_p\}_{p=1}^{C}$ correspond to the set of labels associated with the basic unit classes of the handwriting recognition system. We generate n unique basic unit pairs with $n = \frac{C(C-1)}{2}$. For each such pair, namely (c_i, c_j) $(i \neq j)$, we identify their discriminative region with our proposed technique and extract the d-dimensional features. Thereafter, a dedicated two-class classifier is trained only on the features in the discriminative region, thereby leading to an ensemble of $\frac{C(C-1)}{2}$ two-class classifiers.

For recognition, the unknown sample is passed through the feature extraction module that extracts the d dimensional feature vector corresponding to each (x, y) point of the pattern. Thereafter, each of the two-class classifiers of the ensemble performs a binary classification on this sample. In particular, it may be noted that a given classifier is trained on feature vectors associated with the appropriate discriminant region of the basic unit patterns under consideration. For making the decision, a score function S_p is computed for each class c_p, $p = [1, 2, ..., C]$ that takes into regard, the favorable and

Figure 4.3: The two panels of sub-figure (a) depict the visualization of the feature distribution of a confusing pair (খ়, খ) using t-SNE algorithm. The features of the two classes are marked with red and blue colors respectively. In the top panel, we consider the distribution obtained from the whole basic unit pattern, while in the bottom, we depict the same from the discriminative region selected by our proposed methodology. On a similar note, the sub-figures (b)-(d) show the corresponding plots for the basic unit pairs (কী, ঘী), (U, W) and (g, y) respectively.

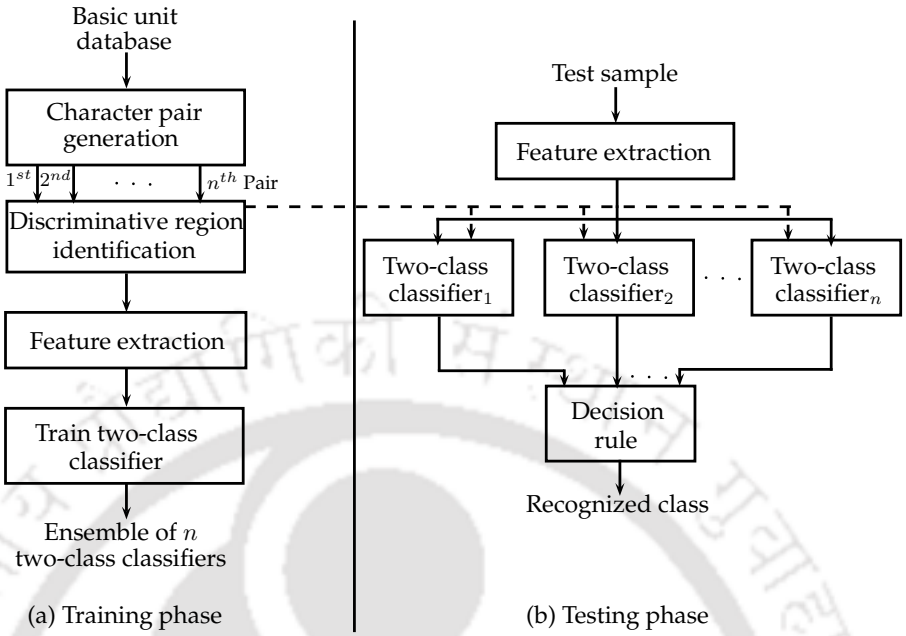


Figure 4.4: Block diagram of the proposed discriminative region based basic unit recognition system. The sub-figures (a) and (b) depict the training and testing phase of the system. The value of n is $\frac{C(C-1)}{2}$ where C is the total number of basic unit classes whose test data are to be recognized. The dotted line indicates that the respective discriminative regions are considered in the development of the two-class classifiers.

unfavorable votes. In other words, we have

$$S_p = \sum_{j=1, j\neq p}^{C} \mathcal{I}(p \neq j) \tag{4.4}$$

Here, $\mathcal{I}$ is an indicator function that outputs $+1$ when the feature vector $\mathbf{O}$ of the test sample is assigned to class c_p and 0 otherwise. The final decision about the class label of the unknown sample is taken on the basis of the criterion

$$\hat{c} = \arg\max_{1 \leq p \leq C} S_p \tag{4.5}$$

4.4 Extension to word recognition

In this Section, we extend the aforementioned strategy to developing a word recognition system. Fig. 4.5 depicts the architecture that bears semblance to the one discussed in Section 3.5. The only difference, however, is that the reevaluation is carried out by employing the feature vectors from the discriminative region.

For sake of illustration, consider the i^{th} probable lexicon word W_i to comprise of N_i basic units $\{c_1^i, c_2^i, ..., c_{N_i}^i\}$. When the online input sample is forced aligned with this word, the baseline HMM divides

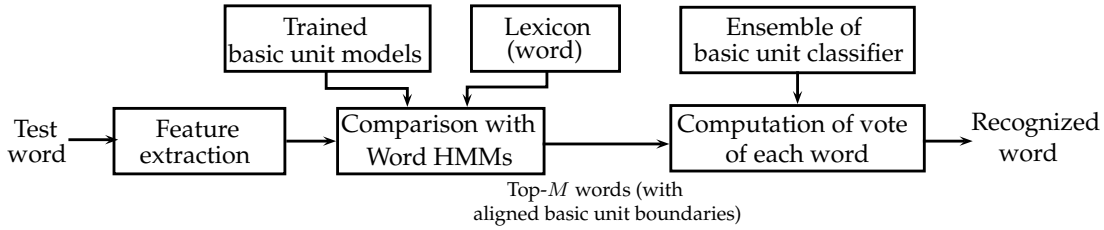


Figure 4.5: Block diagram of the developed discriminative region-based word recognition system. Each classifier of the ensemble is trained on a pair of basic units by utilizing the feature vectors associated with their corresponding discriminant regions.

it to N_i segments. For the reevaluation of the p^{th} segment with the basic unit label c_p^i $(1 \leq p \leq N_i)$, we pass it to the subset of $(C-1)$ classifiers in the ensemble to revise the recognition score. The choice of the classifiers is based on whether they have been trained by the data from the basic unit class c_p^i under consideration. Accordingly, we update the votes by S_p^i using Equation (4.4). It is important to take note that each of the selected classifiers in the ensemble is trained on a pair of basic units by utilizing the feature vectors associated with their corresponding discriminant regions.

On the whole, we obtain N_i revised voting scores of $\{S_p^i\}_{p=1}^{N_i}$ corresponding to each basic unit constituting the word W_i. Accordingly, we can now compute the average votes for the word W_p as:

$$\bar{S}_i = \frac{\sum_{p=1}^{N_i} S_p^i}{N_i} \tag{4.6}$$

where S_1^i, S_2^i,..,$S_{N_i}^i$ are the individual revised voting scores of the N_i basic units for the word W_i. The lexicon entry with the highest average number of votes is taken as the recognized word. In other words, we can write

$$\widehat{w'} = \underset{1 \leq i \leq M}{\arg\max}\ \bar{S}_i \tag{4.7}$$

where $\widehat{w'}$ is the revised word by the proposed system.

4.5 Result and discussion

In this Section, we evaluate the efficacy of our proposals for basic unit and word recognition tasks. The experiments are conducted on Assamese and English databases, the details of which are given in Section 1.4. The training set is used to create the models while the validation and test sets are used to optimize and evaluate the system respectively.

Table 4.1: Performance (error rate in %) of the proposed discriminative region-based single-stage system, where the discriminative regions are selected by varying the cluster K and window size α in proposed discriminative region selection technique. The results are separately reported for the HMM and SVM classifiers. The **validation set** of Assamese modified character and English lowercase letter are used for the experiment. The minimum error rates are denoted in **bold**.

Classifier	No of cluster K	Assamese modified character					English lowercase letter				
		Window size α					Window size α				
		2	4	6	8	10	2	4	6	8	10
HMM	2	4.57	3.73	3.83	4.25	4.57	5.47	5.34	5.39	5.39	5.47
	4	4.75	3.47	**2.83**	3.36	3.83	5.60	5.34	**5.21**	5.29	5.39
	6	4.91	3.83	3.73	3.10	3.83	5.60	5.29	5.34	5.29	5.34
	8	5.12	3.73	3.73	3.36	3.62	5.81	5.39	5.39	5.29	5.21
SVM	2	7.36	6.20	6.65	6.65	6.88	3.46	3.38	3.46	3.46	3.46
	4	7.54	6.33	**5.12**	5.12	5.54	3.46	3.32	**3.25**	3.32	3.38
	6	7.75	6.65	5.54	5.81	6.33	3.46	3.32	3.25	3.32	3.38
	8	7.94	6.65	6.20	5.54	6.20	3.51	3.38	3.32	3.32	3.25

4.5.1 Basic unit recognition

In this section, we present the performance of the proposed single-stage system where the discriminative regions in the basic unit pairs are selected by the proposed selection technique of Section 4.2.

To begin, first, we develop a baseline basic unit recognition system that considers features extracted from the whole pattern as given in Equation (3.10). In this Chapter, we consider two baseline classifiers namely, the HMM and SVM. The details of the HMM-based system is already presented in Section 3.6. For the SVM-based system, the one-vs-one strategy is used for training with Radial Basis Functions (RBF) as the kernel. The parameters of the kernel are optimized by employing the grid search approach. The LIB-SVM toolkit [116] is used for our implementation.

In the proposed discriminative region selection technique, one is required to tune the window size (α), shift (β) and cluster size (K) to achieve the optimal performance. Table 4.1 presents the evaluation of the proposed system on the validation set of the Assamese modified character and English lowercase datasets [4]. The discriminative regions are selected by varying the window size α (with shift $\beta = \alpha/2$) and cluster size K. The reasoning behind keeping the shift to half of the window size is inspired by the work of [117] in the speech recognition literature.

It can be observed that when increasing the value of α, the error rate decreases and reaches the minimum at $\alpha = 6$. Thereafter, it starts increasing for the basic unit recognition task. In particular, with a lower value of α, the variations of data due to different writing styles may not be modeled

[4]The details of these datasets can be found in Tables 1.1 and 1.2.

Table 4.2: Performance of the proposed discriminative region-based single-stage system evaluated on the different **validation sets**. The corresponding cluster size K and window size α values used in discriminative region selection are also given. The performance of baseline system is also reported.

Dataset	Baseline system		Proposed system			
	HMM	SVM	HMM		SVM	
			(K, α)	Error rate	(K, α)	Error rate
Assamese digit	1.23	0.99	(6,6)	0.74	(4,6)	0.74
Assamese basic character	3.65	4.50	(4,6)	2.57	(4,6)	3.36
Assamese modified character	3.76	7.94	(4,6)	2.83	(4,6)	5.12
English digit	1.25	1.35	(2,4)	1.00	(4,6)	1.10
Uppercase letter	3.90	3.30	(6,8)	3.24	(4,6)	3.06
Lowercase letter	6.40	4.59	(4,6)	5.21	(4,6)	3.25

Table 4.3: Error rate (in %) of the baseline and proposed discriminative region-based single-stage system for Assamese and English basic unit recognition tasks, evaluated on the **test sets**.

Dataset	Baseline system		Proposed system	
	HMM	SVM	HMM	SVM
Assamese digit	1.13	0.75	0.56	0.56
Assamese basic character	3.87	4.60	2.89	3.71
Assamese modified character	4.00	6.39	2.98	4.76
English digit	1.13	1.21	0.86	0.90
Uppercase letter	3.67	3.09	2.96	2.49
Lowercase letter	6.82	5.05	5.98	3.92

adequately. This, in turn, has an effect on the selection of the appropriate discriminative region. However, at the same time, a large value of α may over-smooth the data, thus leading to a sub-optimal performance in recognition.

Another parameter that deserves to be analyzed is that of the cluster size K in the proposed technique. It may be noted that a large value captures more variation of the data while a lower one attempts to provide a gross representation. In general, for a given value of α, an increment in $K > 4$ starts to emphasize irrelevant variations of the trace during the selection of the discriminative region, thus leading to an increase in the error rate.

Table 4.2 reports the best performance (minimum error rate) of the remaining datasets for the basic unit recognition task. For each case, the number of clusters K and the size of the window α are explicitly specified. From the entries, it can be seen that a cluster size of $K = 4$ and window size of $\alpha = 6$ largely captures the discriminative region well across each of them.

Table 4.3 presents the error rates of the baseline and the proposed discriminative region based single-stage system on the test set. The results are provided for both the HMM and SVM classifiers.

Table 4.4: Processing time (in millisecond) of the baseline and proposed discriminant region based single-stage systems for basic unit recognition task.

Classifier	System	Assamese			English		
		Digit	Basic character	Modified character	Digit	Uppercase	Lowercase
HMM	Baseline	13.10	138.42	244.50	15.04	39.54	40.17
	Proposed	13.49	150.87	276.28	15.59	42.40	43.28
SVM	Baseline	14.71	207.59	450.62	69.50	162.34	335.56
	Proposed	14.85	217.89	482.15	70.29	168.81	349.98

The HMM system performs better on the basic and modified character recognition tasks, while the SVM achieves higher accuracy on the UNIPEN and Assamese digit datasets.

It may be noted that the improvement in Assamese digit and English digit/ uppercase letter is less compared to the other datasets. The reasoning for this observation is that these have fewer basic units with high shape similarity. Contrast to it, the basic character, modified character as well as English lowercase datasets contain many similar shape characters. Accordingly, when the proposed system is employed for these datasets, a notable reduction in error rate is achieved.

On the whole, we can conclude that the proposed discriminative region based single-stage system can be a replacement for the baseline, specifically when the dataset contains many similar shape basic units. Further, for the sake of completeness, we mention in Table 4.4, the average run-time of both these systems built using the SVM and HMM classifiers.

Table 4.5 presents the efficacy of our approach in resolving some of the encountered confusions in the test set by the proposed system.

A performance comparison of the proposal with prior works reported on the Assamese and UNIPEN character dataset is presented in Table 4.6. It can be seen that a comparable character recognition performance has been achieved on both the datasets.

In addition, the evaluation of two-stage systems are also reported, wherein the discriminative region is selected by employing the proposed and the DTW-DDH technique of [16] respectively. In our implementation of these systems, the HMM and SVM classifiers are considered at the first and second stages. Based on the entries in the table, it is worth noting that the performance of the proposed single-stage system is almost at par or even better to that of the two-stage system on all three datasets.

Table 4.5: Some encountered confusion pairs and their frequency of occurrence in the test set employing the baseline and proposed systems. The HMM classifier is considered for the Assamese character recognition system, whereas the SVM classifier is used to build the English character recognition system. The % of improvement achieved by the proposed system is also reported.

Confusion pair	# of test samples	# of confusions of baseline system	# of confusions of proposed system	% of improvement
(৩,৬)	216	4	1	75.0
(অ,আ)	163	12	7	41.6
(মূ,মৃ)	260	17	10	41.1
(খূ,খৃ)	239	13	8	38.4
(খ,থ)	161	10	7	30.0
(কা,ক্য)	228	14	10	28.5
(গী,ণী)	174	18	13	27.7
(ম,স)	143	19	14	26.3
(ক,ফ)	170	13	10	23.0
(g,y)	896	21	12	42.8
(A,H)	635	6	4	33.3
(4,9)	994	7	5	28.5
(O,Q)	740	14	10	28.5
(a,u)	1612	26	19	26.9
(r,v)	1142	30	22	26.6
(K,R)	633	8	6	25.0
(U,V)	538	12	9	25.0
(r,n)	1832	29	22	24.1

4.5.2 Word recognition

In this subsection, we evaluate the performance of our proposal at the word level. As discussed in Section 4.4, in the first step, a baseline HMM-based word recognition system is developed by concatenating the trained HMMs of its constituent basic units. Thereafter, we train an ensemble of classifiers on pairs of labeled basic units, obtained from the word samples through segmentation.

Fig. 4.6 presents the error rate of the proposed word recognition system for varying top M words on the validation set across three different lexicon sizes of 5000, 10000 and 20000 words. It can be observed that the performance improvement beyond M=4 is negligible. Further, in the Assamese word dataset, the use of HMM classifiers in the ensemble outperforms slightly the SVM. However, at the same time, the converse is observed with the English dataset.

The error rate of the proposed system on the test set is given in Table 4.7. An improvement is noted across all of the considered lexicons, with an average of approximately 3.2% and 3.5% for the Assamese and English datasets, respectively.

Table 4.6: Performance comparison (error rate in %) with other works reported on (a) Assamese character and (b) English UNIPEN character dataset. Unless specifically mentioned, the numbers are the error rates as reported in the respective explorations. For brevity, we abbreviate Discriminate Region as DR. It is important to note that the systems [110,25] despite being two-stage do not consider the features from the discriminative region

(a)

Method	Digit	Basic character	Modified character
Two-stage system [110]	1.20	-	4.30
HMM and SVM combination [25]	1.70	-	-
Two-stage system with :			
DTW-DDH technique [16]	0.56	3.69	3.93
Proposed DR technique	0.56	3.18	3.26
Proposed single-stage DR based system	0.56	2.89	2.98

(b)

Method	Digit	Uppercase	Lowercase
DTW [32]	2.90	7.20	9.30
OnSNT [27]	1.10	4.30	7.90
ANN [1]	0.80	3.10	5.10
HMM [8]	1.73	-	-
Two-stage system with :			
DTW-DDH technique [16]	0.86	2.99	4.45
Proposed DR technique	0.86	2.58	4.07
Proposed single-stage DR based system	0.86	2.49	3.92

Table 4.7: Error rate (in %) of the baseline and proposed discriminative region-based word recognition systems, evaluated on **test sets**.

Lexicon size	Assamese			English		
	Baseline system	Proposed system		Baseline system	Proposed system	
		HMM	SVM		HMM	SVM
5000	22.92	19.85	20.10	26.27	23.76	23.43
10000	26.13	23.00	23.57	29.65	26.62	26.27
20000	28.86	25.32	25.65	33.52	29.67	28.98

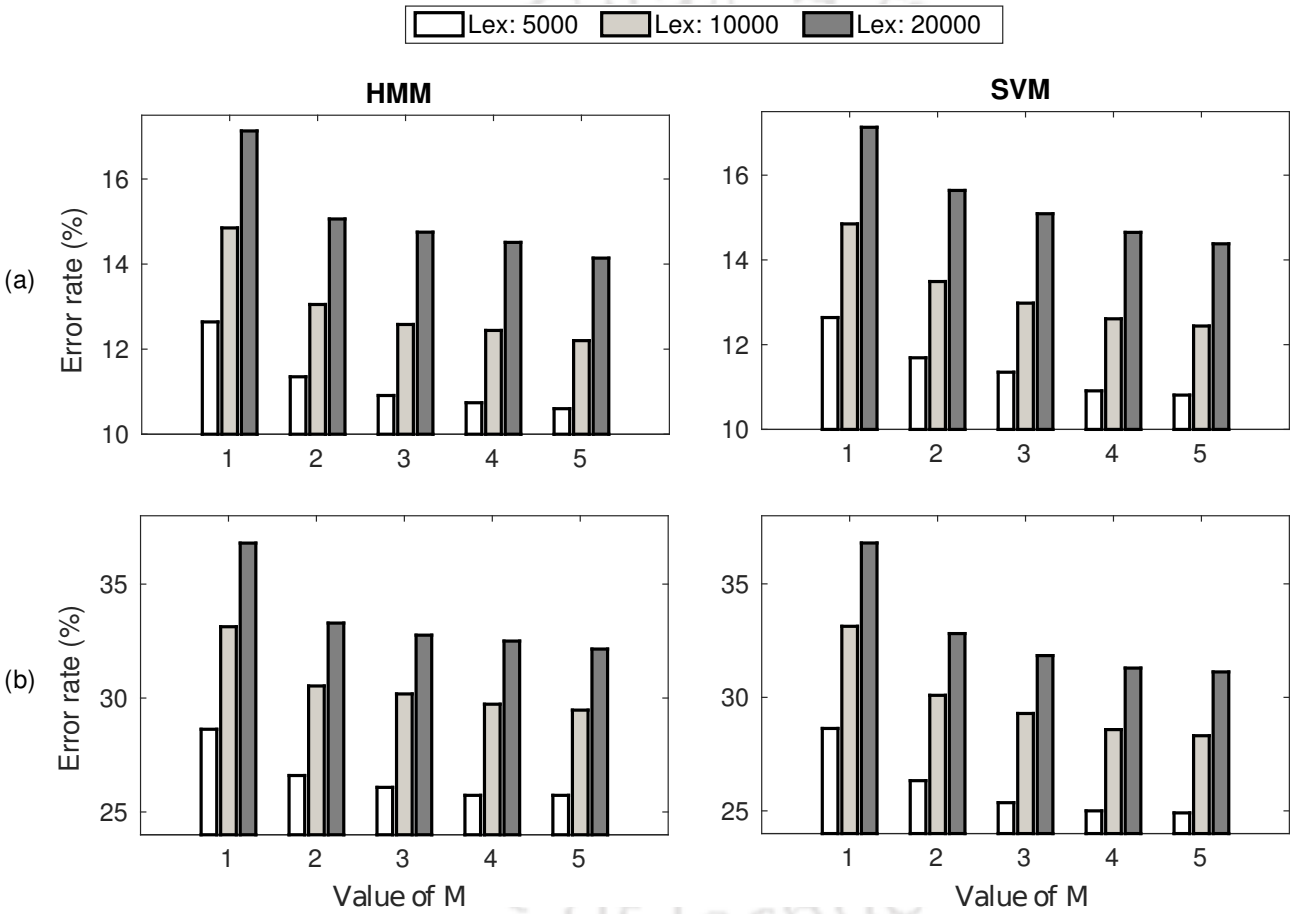


Figure 4.6: The error rate (in %) of the proposed word recognition system on the **validation set** of the (a) Assamese and (b) English word databases for varying number of top-M words (that are reevaluated to revise the decision). The results are depicted for three lexicon sizes by employing the ensemble of HMM and SVM classifiers, that are trained on the feature vectors from the discriminative region.

Table 4.8: Performance comparison of the proposed word recognition system with the literature reported work on the English word dataset.

Method	Lexicon		
	5000	10000	20000
SVM [111]	29.16	32.98	37.41
Proposed system	23.43	26.27	28.98

Moreover, for the English dataset, we compare the performance of the proposed system with the work of [111] given in Table 4.8. It can be seen that our system improves the system performance by 5.73%, 6.71% and 8.43% for the lexicon size of 5000, 10000, and 20000 words respectively.

4.6 Summary

In this Chapter, we proposed a discriminative region selection technique that detects parts of the trace that present fine structural differences in similar looking basic units / patterns. In addition, we presented a single-stage classification framework that takes into consideration the discriminative regions extracted between the basic units. In particular, we employ an ensemble of $\binom{C}{2}$ classifiers corresponding to all possible pairs of classes of basic units, where each of the classifiers are trained utilizing the features from the discriminative region. At the time of recognition, a majority voting scheme is applied to the ensemble for assignment of the identity to the test basic unit pattern. Finally, the discriminative region analysis is extended for a large vocabulary word recognition task. The effectiveness of the proposals has been demonstrated for both basic unit and word recognition tasks on the Assamese and English databases.

5

Novel Features for Basic Units

Contents

5.1 Introduction

In this Chapter, we focus on proposing new feature representation for the basic units that are to be recognized. Recall from Section 2.2 that an online handwritten system is usually developed by considering point-based features that describe the different geometric attributes of handwriting. Due to wide shape variations among the samples of the same class, often the point-based features exhibit a high degree of intra-class variation. One aspect that has hardly been sought after in the literature is to consider the feature vectors, being utilized, from a probabilistic viewpoint. To the best of our knowledge, there are only two prior works in this paradigm [54, 118]. In the first, the authors used class-conditional probabilities from an MLP network as features, while in [54] a six-layer Deep Neural Network is developed with the output from the penultimate layer being considered for feature description.

In the first part of the Chapter, we propose the use of probabilistic features (referred to as 'posterior features') that are derived from a set of Gaussian mixture models (GMMs). The GMMs, being a generative model are intended to capture the class dependent characteristics. We show that the so derived posterior features aid in minimizing intra-class variability in the feature space while at the same time improving the separability between classes. The performance of the proposed GMM posterior features is demonstrated for both basic unit and word recognition tasks of the Assamese and English databases. The results show a notable improvement over the conventional point-based features that have been used for online handwriting recognition.

It is worth mentioning however that GMMs have earlier been employed to build several handwriting systems such as offline digit/character/word recognition [119–121]; gender and handedness detection [122]; signature verification [123, 124] and writer identification [125] to state a few.

In the second part of this Chapter, we aim to extract features directly from the trace of online handwriting, thus alleviating the need for hand-crafted features. In this direction, a convolutional neural network (CNN) [126] is developed to process the data. It may be noted that prior works employing such architecture first convert the online handwritten input to its corresponding bitmap image [22,127]. However in the process of conversion, the utility of important dynamic information such as pen-up/pen-down status, stroke order/directions are likely to get ignored. In order to circumvent this issue, our CNN architecture operates on the online handwriting data directly, thereby eliminating the need of converting it to an offline image.

The first convolution layer accepts the sequence of (x, y) coordinates along the trace of the basic unit as an input and outputs a convolved filtered signal. Thereafter, via alternating steps of convolution and Rectified Linear Unit (ReLU) layers, in a hierarchical fashion, we obtain a set of deep features that can be employed for classification. To the best of our knowledge, this is the first work of its kind that applies a CNN directly on the (x, y) coordinates of the online handwriting data. The results when evaluated for the basic unit and word recognition tasks demonstrate the efficacy of the proposed CNN features over the point-based features.

For the recognition of online handwritten words using the GMM framework, the models are trained on the basic units, that are obtained from segmenting the word samples. To obtain the posterior features, first, a sliding window technique is used to extract the frames from the word samples. Thereafter, the pattern associated with each frame is pre-processed and a d-dimensional point-based feature vector is extracted at each (x, y) point. These feature vectors are then passed through the set of GMMs to obtain the posterior representation. Once the sequence of GMM based feature vectors are available, the HMMs are built for each of the words in the lexicon by process of the concatenation of their constituent basic units.

Further to the above, we also build a large vocabulary word recognition framework with the CNN framework. Here again, the frame patterns associated with the online handwritten word are extracted and passed through the network. By employing the resultant sequence of feature vectors, the word HMMs are built for every entry in the lexicon.

Based on the preceding discussions, the following may be regarded as the research highlights of this chapter.

- Proposal of GMM posterior features for online handwriting recognition.
- Demonstration of the utility of the proposed GMM features in minimizing intra-class variability in the feature space while at the same time improving the separability between the classes.
- Proposal of deep CNN architecture for online handwriting recognition.
- Extraction of CNN features directly from the trace of online handwritten data.
- Development of a basic unit and word recognition systems using the aforementioned features.

The rest of the chapter is organized as follows. The proposed GMM posterior features is described in Section 5.2. This is followed by an overview of the GMM feature based basic unit recognition system

in Section 5.3. In order to demonstrate the utility of the proposed features, we present a visualization of the same in Section 5.4. Subsequent to this, we provide the details of the word recognition systems developed by using the sequence of GMM features in Section 5.5. The experimental results and discussion are provided in Section 5.6.

With regard to the second part of the chapter, we elucidate in Section 5.7, the CNN features for online handwriting. The experimental results are discussed in Section 5.8. Finally, the Chapter is concluded with a summary in Section 5.9.

5.2 Proposed GMM features

A GMM is a parametric probability density function that can model any continuous distribution by a weighted sum of Gaussian components [128]. These models have been shown to capture the underlying statistical variability of the point-based features used to represent the online handwriting data [123]. A GMM with $\mathcal{M}$ Gaussian components is given by:

$$P(\mathbf{o}_i) = \sum_{j=1}^{\mathcal{M}} w_j \; N(\mathbf{o}_i|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \tag{5.1}$$

where $N(\mathbf{o}_i|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ is the d-dimensional Gaussian distribution that takes the form

$$N(\mathbf{o}_i|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_j|^{1/2}} e^{-\frac{1}{2}(\mathbf{o}_i-\boldsymbol{\mu}_j)'\boldsymbol{\Sigma}_j^{-1}(\mathbf{o}_i-\boldsymbol{\mu}_j)}$$

The following may be noted with regard to the mathematical representation of the GMM:

- $\mathbf{o}_i$ is a d-dimensional point-based feature vector
- w_j, $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ denote the weight, mean vector and covariance matrix of the j^{th} Gaussian component respectively

Let $\{\gamma_1, ..., \gamma_C\}$ denote a set of GMMs corresponding to the basic unit classes c_p with $p \in [1, 2, ..., C]$. Our proposed GMM posterior feature vector $\mathbf{v}_i$ is created by stacking the posterior probabilities of $\mathbf{o}_i$ from each of the GMMs and is given by:

$$\mathbf{v}_i = [P(\gamma_1|\mathbf{o}_i) \; P(\gamma_2|\mathbf{o}_i) \; ... \; P(\gamma_C|\mathbf{o}_i)]^T \tag{5.2}$$

More specifically, the posterior probabilities $P(\gamma_p/\mathbf{o}_i)$ are computed through the likelihood estimation

of the GMM, by considering Bayes rule:

$$P(\gamma_p|\mathbf{o}_i) = \frac{P(\mathbf{o}_i|\gamma_p)P(\gamma_p)}{P(\mathbf{o}_i)} \approx P(\mathbf{o}_i|\gamma_p) \tag{5.3}$$

In the above, $P(\mathbf{o}_i)$ is ignored since it appears as a normalization constant that is the same for all classes. Each of the priors $P(\gamma_p)$ are set to $\frac{1}{C}$, thus implying that all the GMMs are equally likely to be selected. The first term $P(\mathbf{o}_i|\gamma_p)$ can be computed from Equation (5.1) by considering the parameters of the GMM γ_p.

Using the preceding approximation, we can rewrite Equation (5.2) for the GMM posterior feature vector [1] as

$$\begin{aligned} \mathbf{v}_i &= [P(\mathbf{o}_i|\gamma_1) \;\; P(\mathbf{o}_i|\gamma_2) \; ... \;\; P(\mathbf{o}_i|\gamma_C)]^T \\ &= [\mathcal{S}_{i1} \;\; \mathcal{S}_{i2} \; ... \;\; \mathcal{S}_{iC}]^T \end{aligned} \tag{5.4}$$

where $\mathcal{S}_{ip}$ is the log-likelihood score of $\mathbf{o}_i$ obtained from the normal density γ_p in the GMM. The symbol T indicates the transpose operation of the row vector. The vector $\mathbf{v}_i$ in Equation (5.4) denotes the GMM posterior feature vector corresponding to the point-based feature vector $\mathbf{o}_i$.

5.3 Proposed basic unit recognition system

The overall framework of the basic unit recognition system employing the proposed GMM posterior features is depicted in Fig. 5.1. It is to be noted that the GMMs are trained for each basic unit class separately with their corresponding data. To extract the posterior features, first, the input sample is passed through the feature extractor module that extracts the sequence of d-dimensional point-based features from the online trace. The resulting feature vectors are then fed to the GMMs to generate the proposed features. Using these, the HMM and SVM classifiers are separately trained for recognizing the basic units. In the testing phase, we extract the GMM posterior features on the input sample and use the trained classifiers (HMM / SVM) for recognition.

To train the GMMs for each class, we use the Expectation-Maximization (EM) algorithm to learn the model parameters [2]. The EM algorithm iteratively estimates the mean vectors, covariance matrices and weight vectors from the training sequence of d-dimensional point-based feature vectors. Initially,

[1] Owing to the approximation made in Equation (5.3), in the remainder of this Chapter, we use the posterior probability and log-likelihood terms interchangeably.

[2] We rely on the available functions of MATLAB for the implementation of GMM.

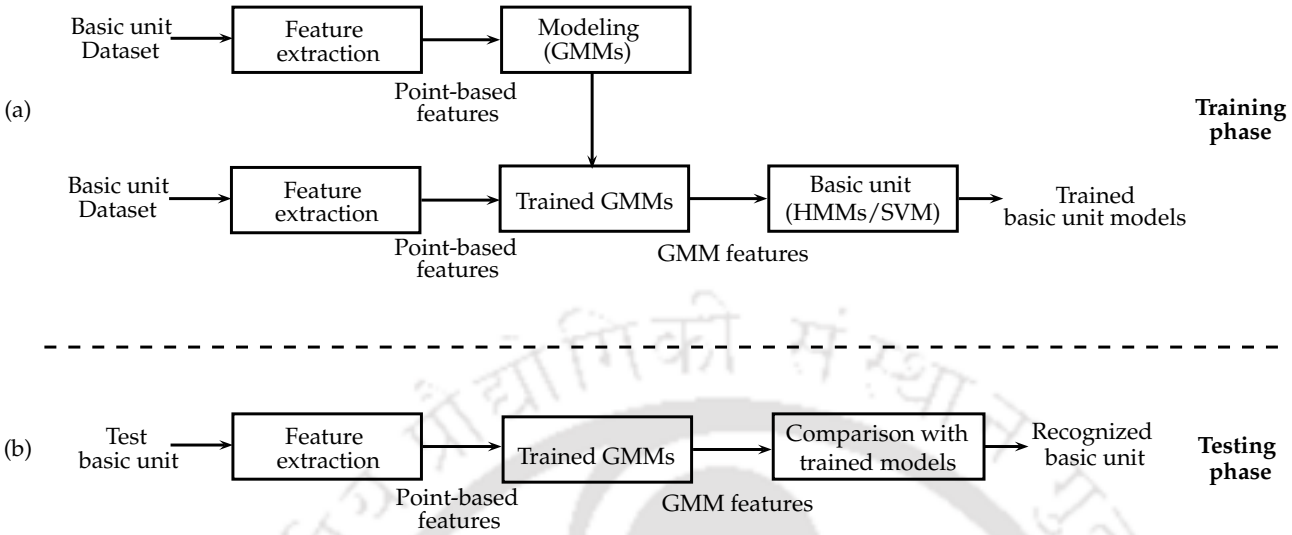


Figure 5.1: The (a) training and (b) testing of the basic unit recognition system employing proposed GMM posterior features.

the model parameters are randomly initialized, and over iterations, they are refined in a way such that the likelihood computed over the feature vectors monotonically increases. The iteration is stopped when a convergence criterion is met. In our implementation, we assume a diagonal structure for the covariance matrix $\mathbf{\Sigma}$ to train the models [128]. Further, to alleviate the problem of ill-conditioned covariance matrix with insufficient training data, we use identical covariance matrices for all the $\mathcal{M}$ Gaussian components of the mixture.

Given a set of q point-based feature vectors, we compute the log-likelihood scores by using Equation (5.1) from each of the GMM models $\{\gamma_1, ..., \gamma_C\}$. This results in transformation of each d-dimensional point-based feature vector to a C-dimensional feature vector of the form given in Equation (5.4). The proposed feature vector sequence corresponding to the basic unit sample can be written as

$$\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_q] \tag{5.5}$$

A pictorial overview summarizing the preceding discussion is given in Fig. 5.2.

5.4 Visualization of GMM features

In this Section, we visualize the intra- and inter-class variability of the feature spaces induced separately by the point-based and GMM feature representations. The goal is to demonstrate the usefulness of the proposed features for the task of handwriting recognition. For the present discussion,

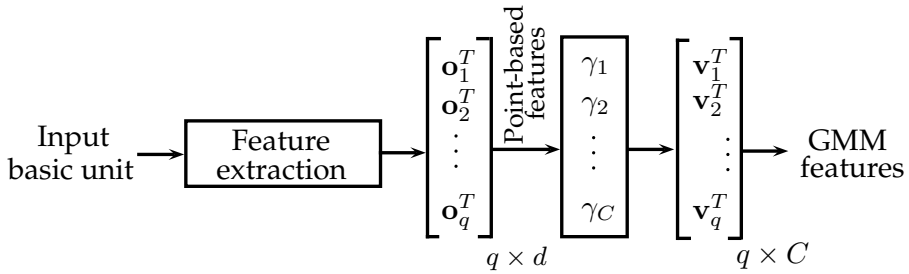


Figure 5.2: Illustration of the GMM feature extraction methodology. A d-dimensional feature vector $\mathbf{o}_i$ is computed at each point of the basic unit having q-points. This forms the point-based feature representation of $[q \times d]$ dimension for the basic unit sample. For the GMM feature extraction, each d-dimension feature vector is transformed to C-dimensional feature vector $\mathbf{v}_i$ by employing GMMs $\{\gamma_1, ..., \gamma_C\}$. This results in a feature representation of $[q \times C]$ dimension for the basic unit sample.

the Assamese digit dataset is considered. A set of 14-dimensional point-based feature vectors and 10-dimensional GMM feature vectors, corresponding to each digit class are extracted. The posterior features are derived from the GMM comprising 32 Gaussian components.

For illustration, we choose two dimensions from each feature set and plot the corresponding feature distributions for the digit ১ (one), ২ (two) and ৩ (three) having class labels c_1, c_2 and c_3. Fig. 5.3 depicts the variation among the three digits by employing the box-plot. Each box being plotted represents the feature distribution of the considered dimension. The chosen point-based features are *cos* α, and y'' while $\mathcal{S}_2$, and $\mathcal{S}_5$ represent two dimensions of GMM posterior features extracted from γ_2 and γ_5, respectively.

For ease of comparison, we normalize the feature values to the $[0, 1]$ range by using the *min-max* normalization approach. It may be observed that the spread in the distributions of GMM posterior features (Fig. 5.3.(b)) is less when compared to that of the conventional point-based features (Fig. 5.3.(a)). This is owing to the fact that the GMM statistically captures the class-specific characteristics by providing feature values that make the intra-class variations compact in the posterior feature space. Further, by comparing the feature distributions across the digits, we can observe a higher inter-class distance between the digits in the proposed feature space. This effect is primarily due to the consideration of all classes in computing the GMM posterior features.

To quantify the inter-class variability, we compute the divergence score [129] for the chosen features. Recall from basic pattern classification theory that a higher divergence value is an indicator of better inter-class separation. The computation steps for the score are as follows:

First the *intra-class* ($\sigma^2_{\mathcal{W}}$) and *inter-class* ($\sigma^2_{\mathcal{B}}$) variances are computed and thereafter the ratio of

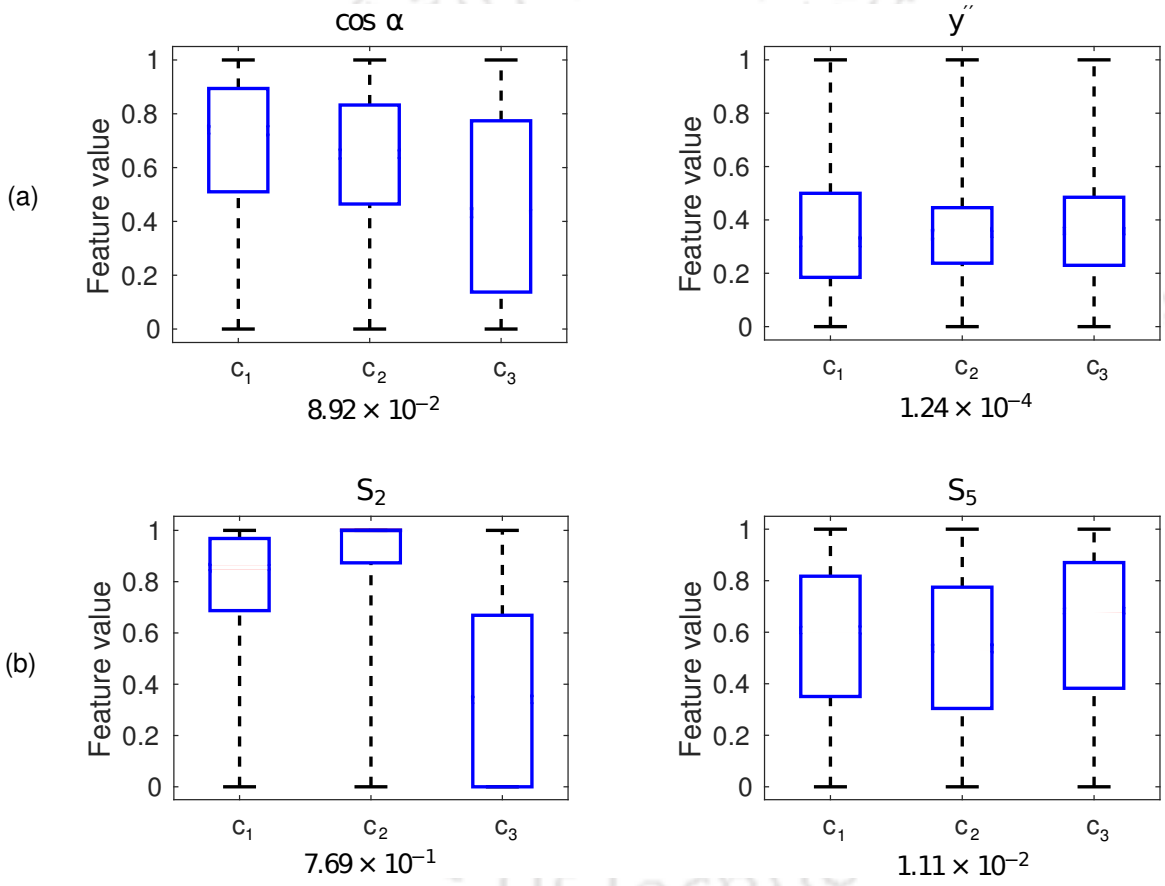


Figure 5.3: (a) Feature distributions of three digits ১ (one), ২ (two) and ৩ (three) that employ the point-based features *cos* α, and y'' and (b) the features $\mathcal{S}_2$, and $\mathcal{S}_5$ derived from the trained class specific GMMs γ_2 and γ_5. The divergence value, measuring the discrimination ability of a feature is also provided.

$\sigma^2_{\mathcal{W}}$ and $\sigma^2_{\mathcal{B}}$ is considered as the divergence score. Mathematically, if n_p denotes the number of feature vectors of the p^{th} basic unit class, $p = [1, ..., C]$, and x_{pi} represents the i^{th} feature vector, then $\sigma^2_{\mathcal{W}}$ and $\sigma^2_{\mathcal{B}}$ can be defined by

$$\sigma^2_{\mathcal{W}} = \frac{1}{N_{\text{tot}}} \sum_{p=1}^{C} \sum_{i=1}^{n_p} (x_{pi} - m_p)^2 \tag{5.6}$$

$$\sigma^2_{\mathcal{B}} = \frac{1}{N_{\text{tot}}} \sum_{p=1}^{C} n_p (m_p - m)^2 \tag{5.7}$$

where m_p represents the individual mean feature vector of each basic unit class

$$m_p = \frac{1}{n_p} \sum_{i=1}^{n_p} x_{ip} \tag{5.8}$$

The term m represents the overall mean feature vector computed from the C classes

$$m = \frac{1}{N_{\text{tot}}} \sum_{p=1}^{C} n_p \; m_p \tag{5.9}$$

and N_{tot} represents total number of feature vectors in all classes:

$$N_{\text{tot}} = \sum_{p=1}^{C} n_p \tag{5.10}$$

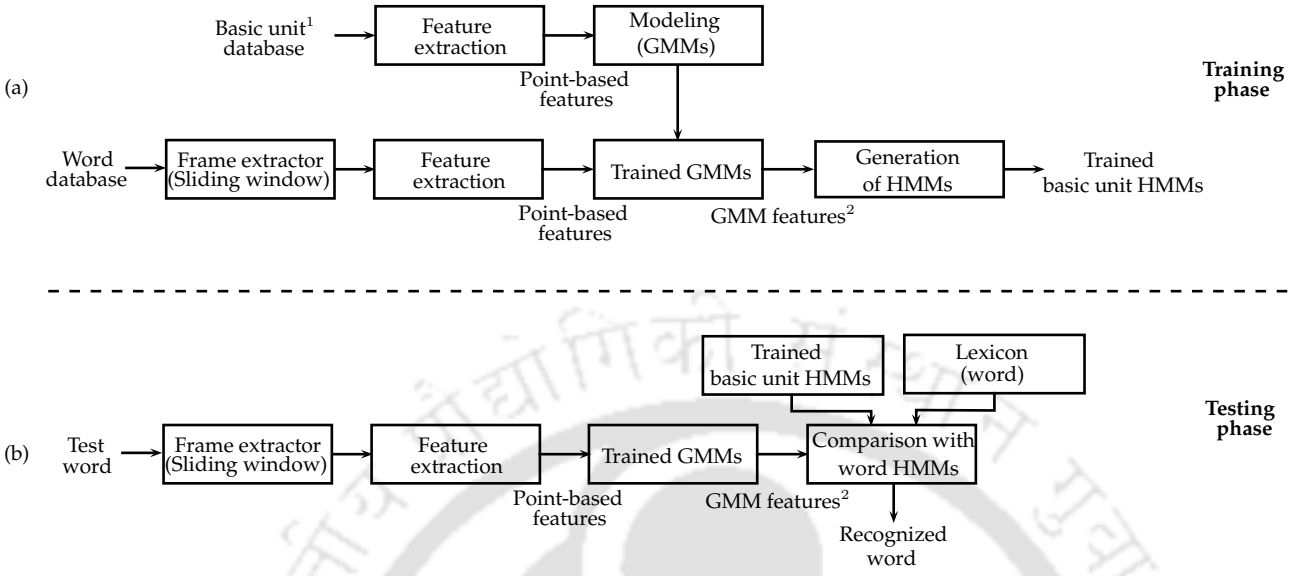
The divergence values for each of the plots are given at the bottom of each panel in Fig. 5.3. It may be noted that higher divergence values are obtained by employing the proposed GMM posterior features. Hence, by utilizing them, a better performance of the online handwriting recognition system can be envisaged.

The GMM posterior features have several advantages over point-based as enumerated below:

- The very essence of feature extraction is to reduce the irrelevant variability while preserving the discriminative information of the data. As demonstrated in the preceding discussion, the estimation of posterior scores through the GMM induces a feature space that alleviates many vulnerabilities of point-based features.

- The proposed features contain conditional probabilities computed from each class. Thus, it measures the degree of assignment of the data to each basic unit model. In other words, it encodes inter-class variability that can serve as a cue to help classify confusing basic unit pairs.

- The GMM features are learned from the training database and therefore can be thought of as



[1] This database is generated by segmenting word data.
[2] There is slight difference in computation of GMM features for word data as given in Section 5.5.3.

Figure 5.4: The (a) training and (b) testing phases of the proposed word recognition system. It is to be noted that the parameters of the HMMs are learned by the sequence of feature vectors that are derived from the trained GMMs.

a data-driven feature extractor. Contrary to it, point-based features are carefully hand-crafted by taking into regard the geometric shape of the characters. Thus, the use of such data-driven features can be exploited to enhance the classification performance.

5.5 GMM feature based word recognition

The overall framework of the proposed word recognition system is depicted in Fig. 5.4. The system is developed in two steps. In the first step, the GMMs are trained on the point-based feature space with the basic unit samples being extracted by employing the baseline HMM that implicitly segments the word samples (Fig. 5.4(a)).

In the next step, we derive a sequence of posterior features from the GMM and utilize the same for building the HMMs (Fig. 5.4(a)). For this, a sliding window technique is used to extract frames from the input word sample. The pattern associated with each frame is then preprocessed and a d-dimensional point-based feature vector is derived at each (x, y) sample point. These feature vectors are then passed through the GMMs for extraction of the posterior features. Once the sequence of the aforementioned features are available, the HMMs are built for each basic unit as discussed in Section 3.2.3.

During the testing phase, the sequence of GMM posterior feature vectors are extracted from the set of frames. Thereafter, they are fed to the HMM-based word models (where the word models are created by concatenating the basic unit HMMs are per the entries in the lexicon) in order to retrieve the most likely word (Fig. 5.4(b)).

In the following, we provide further details of the above steps.

5.5.1 Training of GMMs

To extract the proposed posterior features from an online word data, we need to train the GMMs $\{\gamma_1, ..., \gamma_C\}$ on the labeled basic unit patterns that are segmented from the word samples. However, such basic unit level segmented data is not usually provided with the word database. Hence, in order to obtain the same, we employ a conventional HMM-based system to segment the words of the training dataset into the basic units.

Each of the C basic unit classes is modeled by a separate GMM. However, a mention needs to be made with regard to the input used in the training process. Without loss of generality, the basic units in a word are written mostly in a cursive way leading to a contextual effect, that is observed in the adjacent basic units. In this work, we take advantage of incorporating the same in the training of the GMMs. Said in other words, we prepare a modified training basic unit set such that each of the samples contains one-third fraction of points from both the previous and succeeding characters. This is illustrated in Fig. 5.5 for the word 'adult'. The modified segmented data with the incorporation of context is presented in sub-figure (c). In order to optimize the parameters of the GMMs, a validation set is created by dividing the above-generated data into a training and validation sets in a 2:1 ratio.

For ease of clarity, we present below a simple pseudo-code for the generation of the modified basic units in a given word.

---

1. Input is an online handwritten word W_i comprising N_i basic units.
2. To generate the modified sample for the first basic unit, we consider the contextual information from succeeding basic unit.
3. For second to $(N_i - 1)$ basic units, we consider contextual information from both the previous and succeeding basic units.
4. For the last basic unit, we consider contextual information from only previous basic unit.

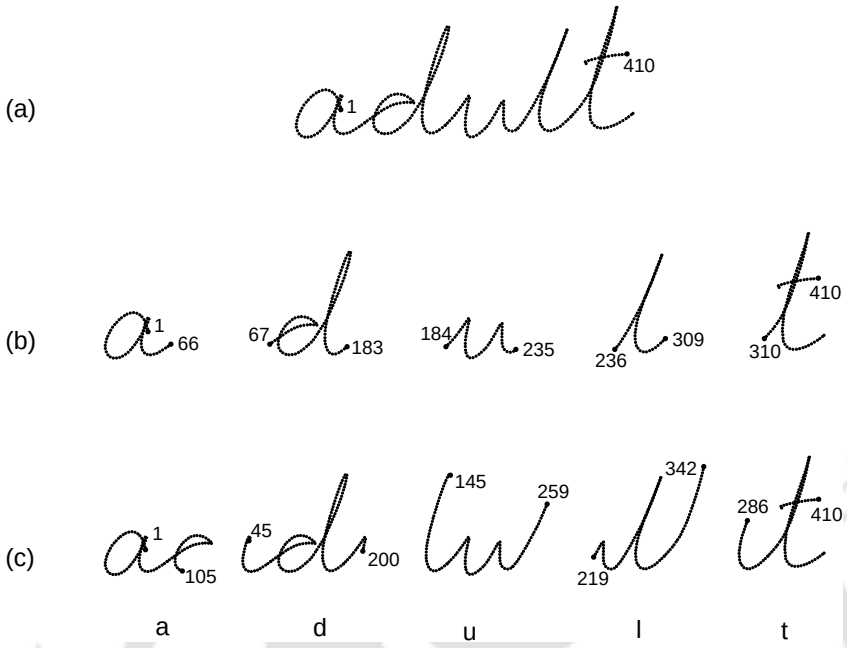


Figure 5.5: (a) An input word 'adult' that comprises 410 (x, y) points. (b) The segmented basic unit boundaries as obtained from HMM system by segmentation. The modified patterns of the basic unit samples used to train the GMMs are shown in sub-figure (c).

5.5.2 Extraction of frames

Given a word sample, we extract a set of frames from the data by using a sliding window technique. The length of the window is determined adaptively for each sample. More specifically, we use window of two different lengths as mentioned below:

$$\text{window length} = \begin{cases} \frac{L^{(x,y)}}{\eta_1 \times N_i}, & \text{for first and last few frames} \\ \frac{L^{(x,y)}}{\eta_2 \times N_i}, & \text{for rest of the frames} \end{cases} \tag{5.11}$$

Here $L^{(x,y)}$ and N_i denotes the number of (x, y) points and basic units present in the word sample W_i. However, during the testing phase, we have to determine the value of N_i heuristically [15] where $N_i > 2$.

The values of η_1 and η_2 determine the length of the window, with a higher value indicating a smaller size for frame extraction. Typically, since there is only a one-sided context available at the begin and end parts of the word, we set a higher value to η_1. Accordingly, we get a smaller window for the first and last few frames. Since, the middle part of the word presents both left and right contexts, a lower value for η_2 is chosen so as to obtain to a larger size window. It may be noted that while implementing the sliding window technique, the overlap of the online trace from adjacent windows are

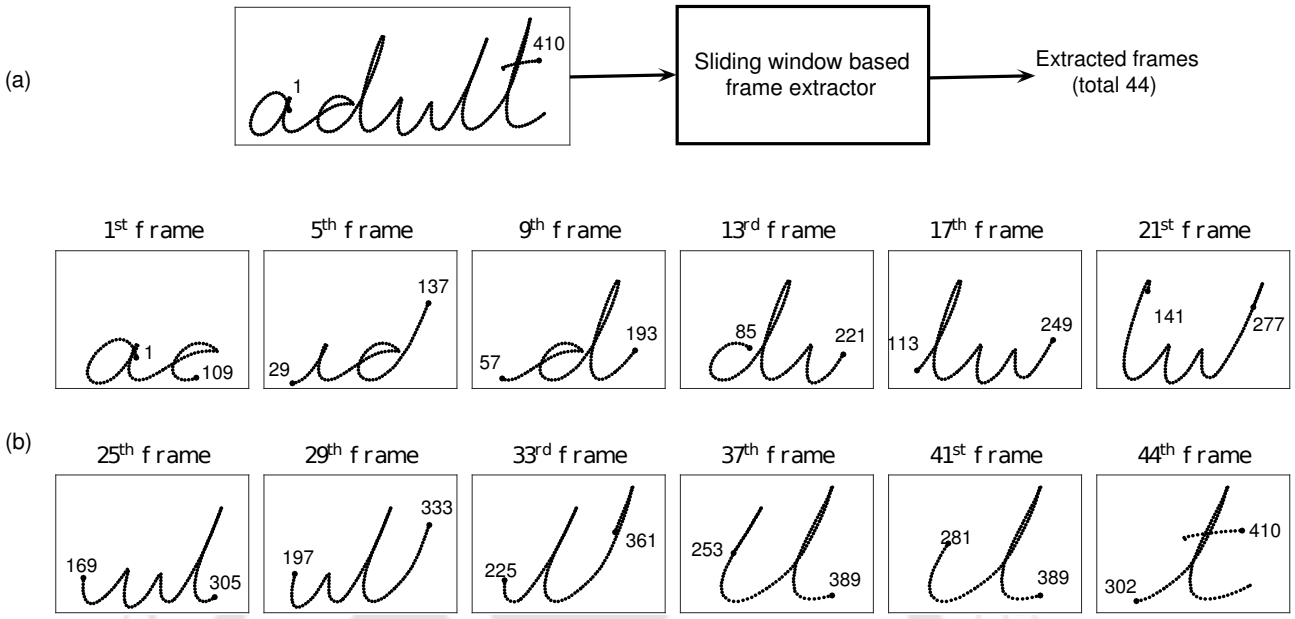


Figure 5.6: (a) The frame extraction procedure for a word 'adult' that contains 410 (x, y) points. The sliding window based technique extracts 44 frames from this word. (b) Patterns of few frames along with their starting and ending point indices. The frame indices are also given above each panel.

considered so that sufficient frames are available for further processing.

Fig. 5.6 illustrates the frame extraction procedure for a 5 letter word 'adult' comprising 410 (x, y) points. In particular, we keep a window length of 109 points for the begin/end and 137 points for the middle part of the word. Each of the adjacent frames has an overlap of 7 points. This in total leads to a set of 44 frames, which are then individually passed through the GMMs for posterior feature extraction.

5.5.3 Feature representation

The pattern associated with each frame is preprocessed and a d-dimensional point-based feature vector is extracted at each (x, y) point. This generates a sequence of q point-based feature vectors corresponding to a frame.

In our approach, we represent each frame by a C-dimensional vector. Essentially, we compute the posterior score of a frame from the GMM γ_p by summing the individual scores from the q feature

vectors [3]. Mathematically, the representation $\widehat{\mathbf{v}}_i$ for the i^{th} frame can be written as:

$$\begin{aligned}\widehat{\mathbf{v}}_i &= \left[\sum_{j=1}^{q} P(\mathbf{o}_{ji}|\gamma_1) \sum_{j=1}^{q} P(\mathbf{o}_{ji}|\gamma_2) ... \sum_{j=1}^{q} P(\mathbf{o}_{ji}|\gamma_C)\right]^T \\ &= \left[\hat{\mathcal{S}}_{i1}\ \hat{\mathcal{S}}_{i2}\ ...\ \hat{\mathcal{S}}_{iC}\right]^T \end{aligned} \tag{5.12}$$

where $\hat{\mathcal{S}}_{ip}$ is the posterior score of the i^{th} frame obtained from GMM γ_p. The notation $\mathbf{o}_{ji}$ denotes the j^{th} point-based feature vector from the i^{th} frame.

Following the above procedure, the posterior feature vector sequence corresponding to a word sample with n-frames can be denoted as:

$$\widehat{\mathbf{V}} = [\widehat{\mathbf{v}}_1, \widehat{\mathbf{v}}_2, ..., \widehat{\mathbf{v}}_n] \tag{5.13}$$

The feature space represented by $\widehat{\mathbf{V}}$ is used to train the HMM-based word recognition system by following the methodology discussed in Section 3.2.3.

5.6 Result of GMM feature based system

The efficacy of the proposed GMM features is demonstrated for basic unit and word recognition tasks. In the following, we present the various experiments conducted and provide the results.

5.6.1 Basic unit recognition

As a first experiment, we present the classification performance of the proposed GMM posterior features by varying the number of Gaussian components $\mathcal{M}$ from 8 to 256 in powers of 2. The results on the validation set are tabulated in Tables 5.1 and 5.2 for the HMM and SVM classifiers respectively. For comparison, we also provide the performance of the systems with the 14-dimensional point-based features of Section 3.3.2.

One can infer that with few Gaussian components, the GMMs do not capture all the variability of the data adequately. This in turn explains the higher error rate. When the value of $\mathcal{M}$ is increased starting from 8, we can see an improvement across both the SVM and HMM classifiers. However, with a large value of $\mathcal{M}$, the error rate again increases owing to the probable over-fitting problem, that arises due to the lack of sufficient training samples. Based on the entries, we see that the lowest

[3]Note that we could have concatenated the sequence of posterior feature vectors corresponding to a frame. However this results in a qC dimensional feature vector, that leads to the curse of dimensionality. Hence, to alleviate the same, we suggest on summing up the scores instead.

Table 5.1: Error rate (in %) of the HMM-based system with GMM features for varying number of Gaussian components. The performance for point-based features is also reported for comparison. The **validation sets** are considered for the experiment. The minimum error rates are denoted in **bold**.

Features	# of Gaussian ($\mathcal{M}$)	Assamese			English		
		Digit	Basic character	Modified character	Digit	Uppercase	Lowercase
Proposed features	8	1.48	4.44	4.57	1.50	4.41	6.94
	16	0.99	3.08	3.47	1.20	3.60	5.92
	32	**0.49**	**2.33**	**2.52**	1.25	3.45	5.81
	64	0.74	2.45	2.73	1.10	3.24	5.34
	128	0.74	3.36	3.36	**0.80**	**2.91**	**4.65**
	256	1.23	3.82	3.54	0.80	3.06	4.73
Point-based	-	1.23	3.65	3.76	1.25	3.90	6.40

Table 5.2: Error rate (in %) of the SVM-based system with GMM features for varying number of Gaussian components. The performance for point-based features is also reported for comparison. The **validation sets** are considered for the experiment. The minimum error rates are denoted in **bold**.

Features	# of Gaussian ($\mathcal{M}$)	Assamese			English		
		Digit	Basic character	Modified character	Digit	Uppercase	Lowercase
Proposed features	8	1.23	6.10	8.57	1.50	3.24	4.00
	16	0.74	4.22	6.88	1.40	2.91	3.25
	32	**0.49**	**3.13**	**4.57**	1.20	2.82	3.16
	64	0.49	3.19	4.91	0.95	2.61	3.08
	128	0.74	3.82	5.54	**0.95**	**2.55**	**2.98**
	256	0.99	4.44	6.33	1.00	2.55	2.98
Point-based	-	0.99	4.50	7.94	1.35	3.30	4.59

error rate is obtained with 32 and 128 mixtures for the Assamese and English databases for both the classifiers. At the same time, it is important to note that the error rate of the proposed system with the GMM features using the optimized $\mathcal{M}$ is consistently better than those of the point-based features.

Table 5.3 presents the performance of the basic unit recognition task on the test sets. Here again, a reduction in error rate is achieved compared to point-based features on all of the datasets, thereby demonstrating the utility of the proposed GMM features.

We believe that the proposed GMM feature representation is quite generic as it captures the underlying statistical variability of point-based features corresponding to the basic unit classes. As a demonstration of this claim, we train the proposed basic unit recognition system on subsets of features from Section 3.2.2. The abbreviation and definition of these features are given in Table 5.4, with their performances being reported in Tables 5.5 and 5.6 on the HMM and SVM classifiers, respectively. The corresponding number of mixture components $\mathcal{M}$ is also provided while reporting the minimum error rates. From the tables, one can observe that there is an improvement while employing the proposed

Table 5.3: Error rate (in %) of the baseline and proposed GMM feature based basic unit recognition systems on the **test sets**.

Dataset	HMM		SVM	
	Point-based features	Proposed features	Point-based features	Proposed features
Assamese digit	1.13	0.56	0.75	0.46
Assamese basic character	3.87	2.51	4.60	2.84
Assamese modified character	4.00	2.70	6.39	3.00
English digit	1.13	0.73	1.21	0.73
Uppercase letter	3.67	2.85	3.09	2.33
Lowercase letter	6.82	5.86	5.05	3.64

Table 5.4: Overview of various feature subsets used for evaluating the GMM feature based system.

Notation	Feature subset	Dimension
F1	$[x, y]$	2
F2	$[x, y, x', y', x'', y'']$	6
F3	$[x, y, x', y', x'', y'', cos\ \theta_w, sin\ \theta_w, cos\ \theta_c, sin\ \theta_c]$	10
F4	$[x, y, x', y', x'', y'', cos\ \theta_w, sin\ \theta_w, cos\ \theta_c, sin\ \theta_c, AR, LN, SL, B_f]$	14

GMM features across the four considered feature subsets.

5.6.2 Word recognition

In this subsection, we evaluate the performance of the proposed HMM-based word recognition system described in Section 5.5. We use GMMs of size 32 and 128 for Assamese and English word datasets, respectively for the extraction of the proposed features. Fig. 5.7 shows the error rate for varying number of HMM states across lexicon sizes of 5000, 10000 and 20000 respectively. The optimized HMM states leading to the minimum error rate are found to be 5 and 4 for Assamese and English datasets, respectively. Further, the number of GMMs used to model each state in the HMM is set to 10.

Another point to be made here is that we have used fewer HMM states in the proposed system as compared to the baseline. This is because the number of GMM feature vectors extracted from a word sample are far less when compared to the number of (x, y) points present in its trace. As an example, it may be recalled that the word sample shown in Fig. 5.6 contains 410 (x, y) points while we extract 44 feature vectors for its representation (one feature vector per frame). This in turn explains the need of having only a few HMM states to adequately model the word data with the proposed GMM feature representation.

Table 5.5: Performance (in %) of the HMM-based system with GMM and point-based features on the different feature subsets, evaluated on **test sets**. The optimized number of Gaussians ($\mathcal{M}$) used for feature extraction in each case is also reported.

Feature subset	System	Assamese			English		
		Digit	Basic character	Modified character	Digit	Uppercase	Lowercase
F1	Point-based features	3.10	14.20	17.34	4.03	7.27	18.12
	GMM features	2.91	13.58	16.57	2.80	6.98	16.50
	(# of $\mathcal{M}$)	(8)	(8)	(8)	(16)	(16)	(16)
F2	Point-based features	1.22	6.30	6.03	1.58	5.12	8.10
	GMM features	0.66	3.83	4.32	0.92	3.95	6.14
	(# of $\mathcal{M}$)	(8)	(8)	(8)	(32)	(32)	(32)
F3	Point-based features	1.13	5.13	4.61	1.43	4.38	7.75
	GMM features	0.56	3.06	3.17	0.86	3.51	6.09
	(# of $\mathcal{M}$)	(16)	(16)	(16)	(64)	(64)	(64)
F4	Point-based features	1.13	3.87	4.00	1.13	3.67	6.82
	GMM features	0.56	2.51	2.70	0.73	2.85	5.86
	(# of $\mathcal{M}$)	(32)	(32)	(32)	(128)	(128)	(128)

Table 5.6: Performance (in %) of SVM-based system with GMM and point-based features on different feature subsets, evaluated on the **test sets**. The optimized number of Gaussians ($\mathcal{M}$) used for feature extraction in each case is also reported.

Feature subset	System	Assamese			English		
		Digit	Basic character	Modified character	Digit	Uppercase	Lowercase
F1	Point-based features	1.97	8.13	11.35	3.25	5.15	8.45
	GMM features	1.69	7.65	10.36	1.35	4.15	6.67
	(# of $\mathcal{M}$)	(8)	(8)	(8)	(16)	(16)	(16)
F2	Point-based features	0.84	6.04	8.40	1.25	3.71	5.73
	GMM features	0.56	4.92	7.19	0.94	2.51	4.08
	(# of $\mathcal{M}$)	(8)	(8)	(8)	(32)	(32)	(32)
F3	Point-based features	0.75	5.65	7.93	1.21	3.11	5.52
	GMM features	0.56	3.13	4.96	0.73	2.35	4.08
	(# of $\mathcal{M}$)	(16)	(16)	(16)	(64)	(64)	(64)
F4	Point-based features	0.75	4.60	6.39	1.21	3.09	5.05
	GMM features	0.46	2.84	3.00	0.73	2.33	3.64
	(# of $\mathcal{M}$)	(32)	(32)	(32)	(128)	(128)	(128)

Table 5.7 presents the error rate of the proposed word recognition system on the test sets of Assamese and English databases with varying lexicon sizes. For comparison, the performance of point-based features is also given. It can be seen that a notable improvement is achieved with the proposed GMM features for all the considered lexicons in both the datasets.

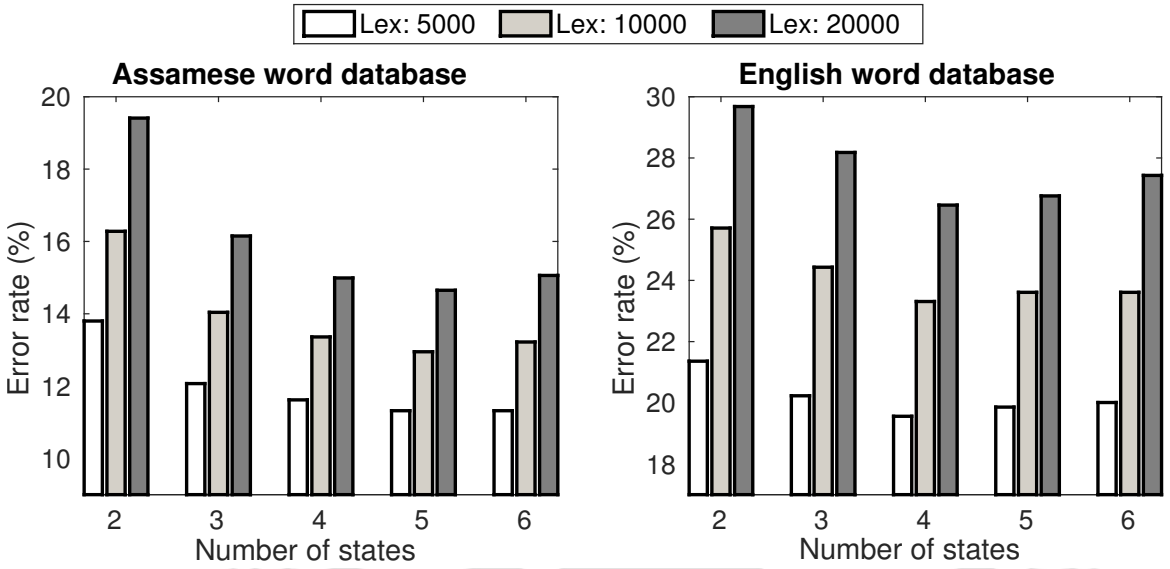


Figure 5.7: Depiction of the error rate (in %) obtained with varying number of HMM states in the GMM feature based word recognition system. The results are evaluated on the **validation sets** of the Assamese and English word datasets for three lexicon sizes.

Table 5.7: Error rate (in %) of the point-based and proposed GMM feature based word recognition systems, evaluated on the **test sets** of the Assamese and English databases.

Lexicon size	Assamese		English	
	Point-based	GMM features	Point-based	GMM features
5000	22.92	19.47	26.27	17.34
10000	26.13	21.70	29.65	20.31
20000	28.86	23.35	33.52	23.71

In the next section, we elucidate the CNN features for online handwriting.

5.7 CNN model for online handwriting

In this part of the chapter, we develop a CNN architecture [126, 130] that can process the online handwriting directly without converting it to an image. Fig. 5.8 depicts a pictorial overview of the same. The input to the CNN is a fixed-size $[q \times 2]$ matrix [4] representing an online handwritten basic unit pattern. This data is first passed through a stack of convolution layers followed by a fully-connected layer and an output layer. At the output layer, a 'softmax' function is employed to classify the data with probabilistic values.

The convolution layers form the main building blocks of the CNN. They comprise a set of trainable filters (or kernels) that perform convolution with the (x, y) points across the temporal sequence of online handwriting, resulting in the generation of feature maps. The feature maps are then stacked together and form the output of the convolution layer. Added to this, each convolution layer is equipped with a Rectified Linear Unit (ReLU) that performs the operation $f(x) = \max(0, x)$ to enforce non-linearity in the network.

In the CNN, the size of the filters plays a key role in determining the relevant discriminative features from the data. A filter with large size may be not capable in capturing low-level features of the data since they are likely to ignore the essential details in the pattern. Contrast to it, a smaller size filter would do better in encapsulating more of the finer information. The number of learning parameters in the network is also associated with the size of the filters employed. However, to achieve optimal performance, the filter length has to be optimized.

We consider a filter of size $[m \times 2]$ in the first convolution layer, where m is the number of points along the temporal sequence of the data. For the higher convolution layers, the corresponding filters have $[m \times 1]$ size, since the data to be operated upon are 1-D signals. Further, the filter length m is kept constant for all the convolution layers. This is owing to recent studies which demonstrate that same size filters in all the convolution layers is found to be better when compared to networks with variable filter lengths [126].

Another parameter associated with the CNN is the stride factor, that specifies how much the filter will move at each step. A larger stride results in a smaller feature map, that is often used for dimensionality reduction. However, in our network, we keep the convolution stride to one point so as to obtain maximum size feature maps for the representation of the online handwriting data.

[4] Here, q represents the number of resampled points in the basic unit sample. In our work, we resample the Assamese and English basic units to 100 and 50 points respectively as discussed in Equation (3.1).

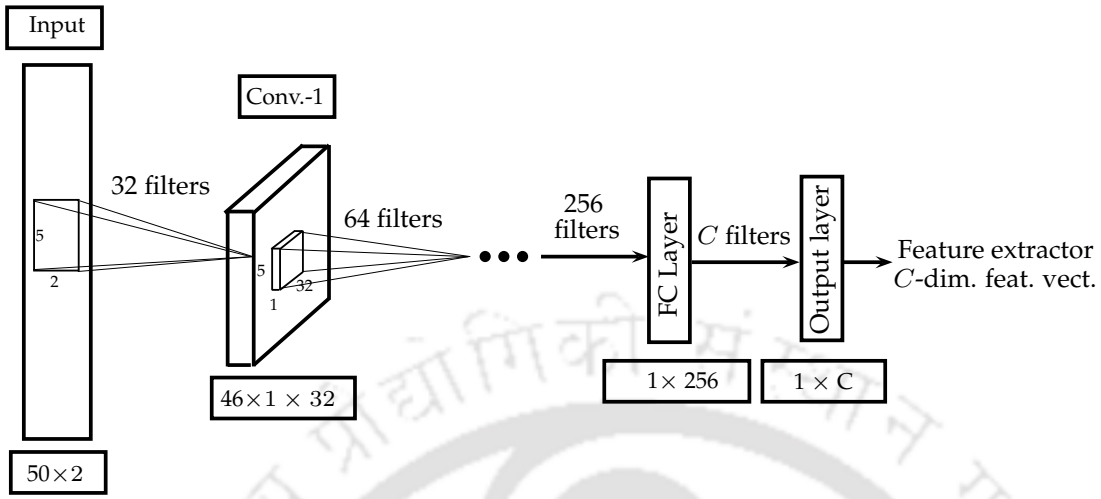


Figure 5.8: Depiction of the CNN architecture employed for modeling the online handwritten sample, that is resampled to 50 points. The feature representation used in our approach corresponds to the output of the last layer, *viz* a C-dimensional vector. Here FC represents the fully connected layer of 256 nodes.

It is to be noted that our CNN architecture also contains one fully connected and one output layer. Without loss of generality, the output of a convolution layer is a 3-D volume while the input of the fully connected layer requires a 1-D vector. For this, we flatten the output of the final convolution layer to a vector by arranging the 3-D volume of numbers into a 1-D vector. In our framework, the fully connected layer has 256-filters while the output layer contains C-filters corresponding to the number of basic unit classes that are to be trained.

The number of filters in the convolution layers are optimized and set in between 32 to 64. The weights of the network are trained under the cross-entropy objective function by employing the back-propagation algorithm with stochastic gradient descent.

Typically, in literature, a CNN architecture operating on image data considers a pooling layer after the convolution operation to reduce the dimensionality of the data. However, in our framework, we rely solely on the convolution layers to construct the features for online handwriting. The motivation behind our choice is as follows:

- Recent studies show that a CNN comprising convolution layers with occasionally dimension reduction by using a stride of 2 can lead to better performance [126].
- The total number of (x, y) coordinates present in the trace of the online handwritten data is much less when compared to the pixels of the image data. Hence, the need for performing dimensionality reduction by adding pooling layers may not arise.

Table 5.8: An overview of the various CNN architectures considered in our work. The depth of the CNN is increased from left (*Net-A*) to right (*Net-D*), by adding more convolution layers to the network. As an abbreviation, the term 'conv5-32' in *Net-A* denotes that the CNN has 32 filters each of length 5. A similar interpretation can be made with respect to the other abbreviations used.

Net-A	*Net-B*	*Net-C*	*Net-D*
3 Layers	4 Layers	5 Layers	6 Layers
conv5-32	conv5-32	conv5-32 conv5-32	conv5-32 conv5-32
-	conv5-64	conv5-64	conv5-64 conv5-64
Fully Connected layer-256			
Output layer- #*C*			
soft-max			

5.7.1 Architectures adopted

In this subsection, we outline the various CNN architectures used in this work. For clarity, these networks have been abbreviated as *Net-A*, *Net-B*, *Net-C* and *Net-D* in Table 5.8.

- *Net-A*: Consists of three layers having 1 convolution, 1 fully connected and 1 output layer.
- *Net-B*: Consists of four layers having 2 convolution, 1 fully connected and 1 output layer.
- *Net-C*: Consists of five layers having 3 convolution, 1 fully connected and 1 output layer.
- *Net-D*: Consists of six layers having 4 convolution, 1 fully connected and 1 output layer.

In each of these networks, the depth is steadily increased by adding more convolution layers. This in a way helps in extracting more discriminative features from the data. Further, all of these networks follow a generic design with the use of $[m \times 2]$ size filters in the first layer and $[m \times 1]$ filters in the higher convolution layers (more than 1). For implementing the above architectures we employ MatConvNet [131], that is available in MATLAB.

5.7.2 CNN features for basic units

We use the above-developed architectures to extract features automatically from the online handwriting. The CNN extracts various characteristic of the data in each of the layers. The weights of the layers are optimized during training in a way that the output layer can minimize the classification error. The neurons in the fully connected layer perform a linear combination with the previous layer outputs and thus we can consider their outputs as being linearly separable. Accordingly, in our work,

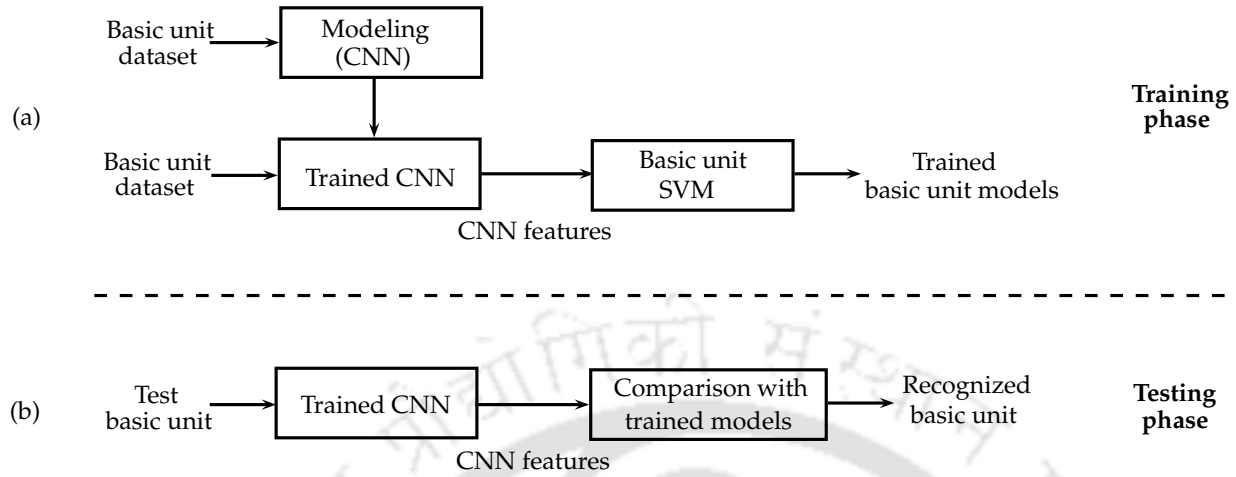


Figure 5.9: The (a) training and (b) testing phases of the basic unit recognition system that employs the features extracted from a CNN.

we have used the output of the last layer as the proposed features. This has a dimension C, that corresponds to the number of classes, whose data is used for training our network (Fig. 5.8).

Accordingly, for a given basic unit sample represented by $[q \times 2]$ matrix, the CNN generates a single C-dimensional feature vector, that can be represented as:

$$\boldsymbol{\mathcal{V}} = [v_1 \; v_2 \; ... \; v_C]^T, \tag{5.14}$$

Here, v_i denotes the score assigned to the i^{th} output node of the CNN, or in other words, the score corresponding to the i^{th} basic unit class.

Fig. 5.9 depicts the overall architecture of the proposed CNN feature based basic unit recognition system. The system is trained in two steps. In the first step, one of the CNN architectures described in Table 5.8 are trained to model the basic unit classes. Thereafter, the features obtained from the output layer are extracted by employing this model. Following this, an SVM classifier is trained to develop the basic unit recognition system.

During the testing phase, the CNN features of a sample are extracted from the learned architecture, which is then used for classification.

5.7.3 Word recognition system

The overall framework of the CNN feature based word recognition system is depicted in Fig. 5.10. In the first step, we need to train the CNN on the labeled basic unit patterns that are segmented from the word samples. However, such basic unit level segmented data is not usually provided with

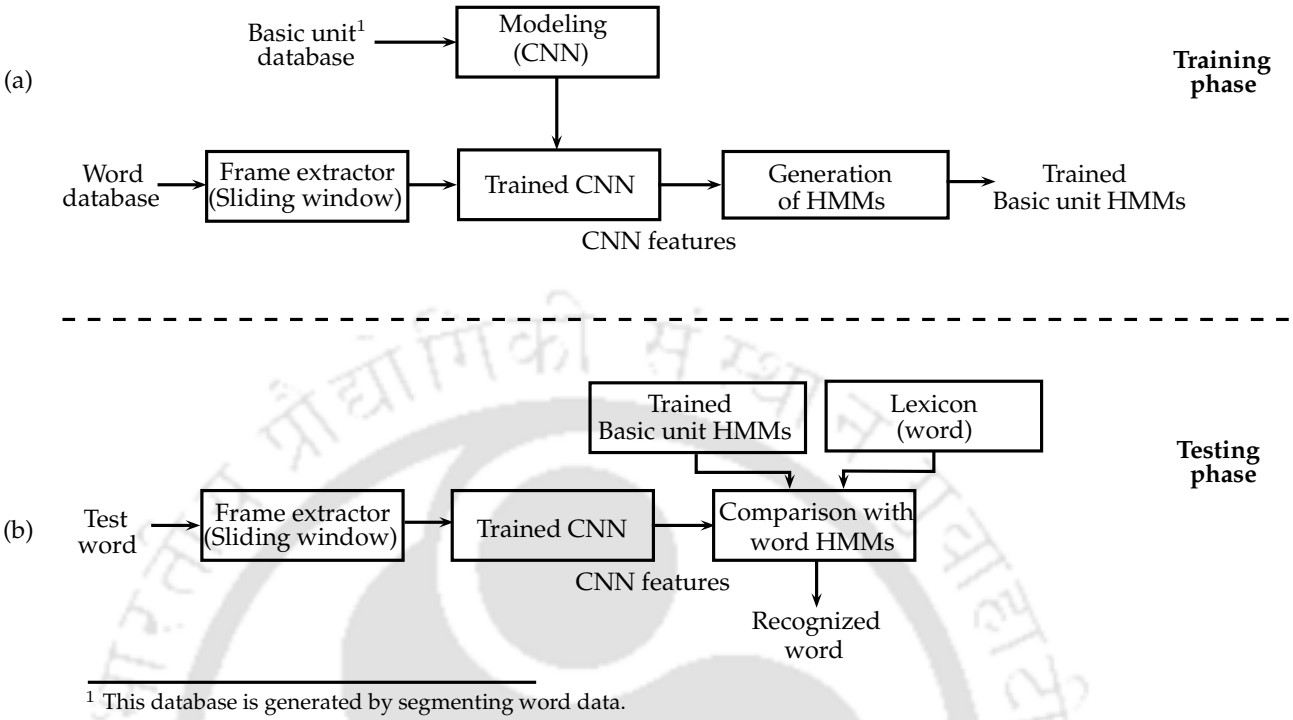


Figure 5.10: The (a) training and (b) testing phases of the proposed word recognition system. It is to be note that the parameters of the HMMs are learned by the sequence of feature vectors that are derived from the CNN.

the word database. In order to obtain the same, we employ a conventional HMM-based system to segment the words of the training dataset into the basic units.

We prepare a modified training basic unit set such that each of the samples contain one-third fraction of points from both the previous and succeeding basic units. This ensures the incorporation of the contextual information into the segmented basic unit samples, prior to training. Further details of the data generation methodology can be found in Section 5.5.1.

In the next step, the HMMs are trained by employing the sequence of CNN feature vectors (Fig. 5.10(a)). For this, we represent a word sample in terms of a set of frames using a sliding window technique, similar to the methodology described in Section 5.5.2. The pattern associated with each frame is then preprocessed to make it a $[q \times 2]$ matrix, which is then passed through the CNN model for feature extraction. The CNN provides a C-dimensional representation for each of the frames. Thus, the feature vector sequence corresponding to a word sample with n-frames can be denoted as:

$$\hat{\boldsymbol{\mathcal{V}}} = [\boldsymbol{\mathcal{V}}_1,\ \boldsymbol{\mathcal{V}}_2, ..., \boldsymbol{\mathcal{V}}_n]^T \tag{5.15}$$

Table 5.9: Error rate (in %) of the SVM-based basic unit recognition system using CNN features. The results are reported on the **validation sets**. The performance of the point-based features with SVM classifier is also given for comparison.

Dataset	Point-based features	CNN model			
		Net-A	*Net-B*	*Net-C*	*Net-D*
Assamese digit	0.99	0.74	0.49	0.49	0.74
Assamese basic character	4.50	3.19	3.08	2.85	3.08
Assamese basic modified character	7.94	5.12	4.57	4.25	4.91
English digit	1.35	1.10	0.95	0.75	0.75
Uppercase letter	3.30	2.91	2.55	2.37	2.37
Lowercase letter	4.59	3.78	2.98	2.64	2.64

where $\boldsymbol{\mathcal{V}}_i = [v_{i1}\ v_{i2}\ ...\ v_{iC}]^T$ is the CNN feature vector corresponding to the i^{th} frame. Once the sequence of CNN features are extracted, the HMMs are built for each basic unit following the methodology discussed in Section 3.2.3.

For a test word, the sequence of CNN feature vectors are extracted in a similar way and fed to the HMM-based word models in order to retrieve the most likely word as shown in Fig. 5.10(b). The word models are created by concatenating the basic unit HMMs as per the entries in the lexicon.

5.8 Result of CNN feature based system

In this Section, we present the results of the basic unit and word recognition using the proposed CNN features.

5.8.1 Basic unit recognition

Table 5.9 presents the performance of the basic unit recognition system with the SVM classifier on the validation set. The employed feature sets are obtained by considering each of the architectures *Net-A* to *Net-D* separately. For comparison, we provide the performance of an SVM system with the hand-crafted point-based features discussed in Section 3.2.2. We have optimized the value of filter length m in the CNN for Assamese and English datasets and found that $m = 5$ is sufficient to extract the discriminative features from the data. Further, it can be observed that, as the number of convolution layers is increased, the error rates are also reduced. The lowest error rate corresponds to the architecture *Net-C* that comprises 3 convolution and 1 fully connected layer.

Table 5.10 reports the result of the basic unit recognition task on the test samples of each of the datasets. It can be seen that the performance of CNN features is notably higher than that of the

Table 5.10: Error rate (in %) of the SVM-based system using CNN features (with architecture: *Net-C*). The results are reported on the **test sets**. The performance of the point-based features with SVM classifier is also given for comparison.

Dataset	Point-based features	CNN features: *Net-C*
Assamese digit	0.75	0.46
Assamese basic character	4.60	2.29
Assamese modified character	6.39	2.45
English digit	1.21	0.63
Uppercase letter	3.09	2.16
Lowercase letter	5.05	3.45

Table 5.11: Some encountered confusion pairs and their frequency of occurrence in the test set employing the baseline and proposed systems. The HMM classifier is considered for the Assamese character recognition system, whereas the SVM classifier is used to build the English character recognition system. The % of improvement achieved by the proposed system is also reported.

Confusion pair	# of test samples	# of confusions of baseline sys.	# of confusions of GMM feat. sys.	% of improv.	# of confusions of CNN feat. sys.	% of improv.
(৩,৬)	216	4	1	75.0	-	-
(অ,আ)	163	12	5	58.3	-	-
(মূ,মৃ)	260	17	8	52.9	-	-
(ম,স)	143	19	10	47.3	-	-
(ক,ফ)	170	13	7	46.1	-	-
(খূ,খৃ)	239	13	7	46.1	-	-
(গী,ণী)	174	18	10	44.4	-	-
(খ,থ)	161	10	6	40.0	-	-
(কা,ক্য)	228	14	9	35.7	-	-
(g,y)	896	21	10	52.3	7	66.6
(U,V)	538	12	7	41.6	5	58.3
(O,Q)	740	14	8	42.8	6	57.1
(A,H)	635	6	4	33.3	3	50.0
(a,u)	1612	26	17	34.6	13	50.0
(r,n)	1832	29	19	34.4	15	48.2
(4,9)	994	7	4	42.8	4	42.8
(r,v)	1142	30	20	33.3	18	40.0
(K,R)	633	8	5	37.5	5	37.5

point-based features, thus inferring that they are more discriminative in nature. The approach also leads to reducing the ambiguity between similar looking confusion basic units, as can be inferred by the entries of Table 5.11

Last but not least, we also see how the performance of the proposed GMM and CNN feature based systems fare with the prior works reported on the Assamese and UNIPEN character dataset in Table 5.12. From the entries, it can be seen that a noticeable improvement in character recognition performance is achieved with our novel feature representation approaches.

Table 5.12: Performance comparison (% error rate) with prior works reported on (a) Assamese character and (b) English UNIPEN character dataset. The results of the GMM feature based system correspond to the entries mentioned in Table 5.3.

(a)

Method	Digit	Basic character	Modified character
Two-stage system [110]	1.20	-	4.30
HMM and SVM combination [25]	1.70	-	-
Proposed GMM feature with HMM classifier	0.56	2.51	2.70
Proposed GMM feature with SVM classifier	0.46	2.84	3.00
Proposed CNN feature with SVM classifier	0.46	2.29	2.45

(b)

Method	Digit	Uppercase	Lowercase
DTW [32]	2.90	7.20	9.30
OnSNT [27]	1.10	4.30	7.90
ANN [1]	0.80	3.10	5.10
HMM [8]	1.73	-	-
Proposed GMM feature with HMM classifier	0.73	2.85	5.86
Proposed GMM feature with SVM classifier	0.73	2.33	3.64
Proposed CNN feature with SVM classifier	0.63	2.16	3.45

5.8.2 Word recognition

We now evaluate the performance of the HMM-based word recognition system shown in Fig. 5.10. The procedure mentioned in Section 5.7.3 is used to extract the CNN features from the given word sample. However, the number of extracted CNN feature vectors in the sequence used to train the HMMs are far less, when compared to the handcrafted feature vectors derived from the (x, y) sample points in the word sample. Therefore, it required to tune the HMM states for this system as well. Fig. 5.11 depicts the error rate obtained for varying number of states in the HMMs across the lexicon sizes of 5000, 10000 and 20000 words. For this experiment, the CNN architecture *Net-C* is used to extract the features. From the plots it can be seen that the best performance is achieved with HMM states of 5 and 4 for the Assamese and English datasets. Further, the number of GMMs in each state is optimized to 10 for both datasets.

Table 5.13 presents the performance of CNN feature based word recognition system with varying lexicon size, evaluated on the test sets of the Assamese and English datasets. All the four CNN architectures are considered separately to extract the features. The performance of point-based features is also given for comparison. From the entries, it can be noted that, the consideration of CNN features improves the system performance for all the considered lexicons in both the datasets. The CNN

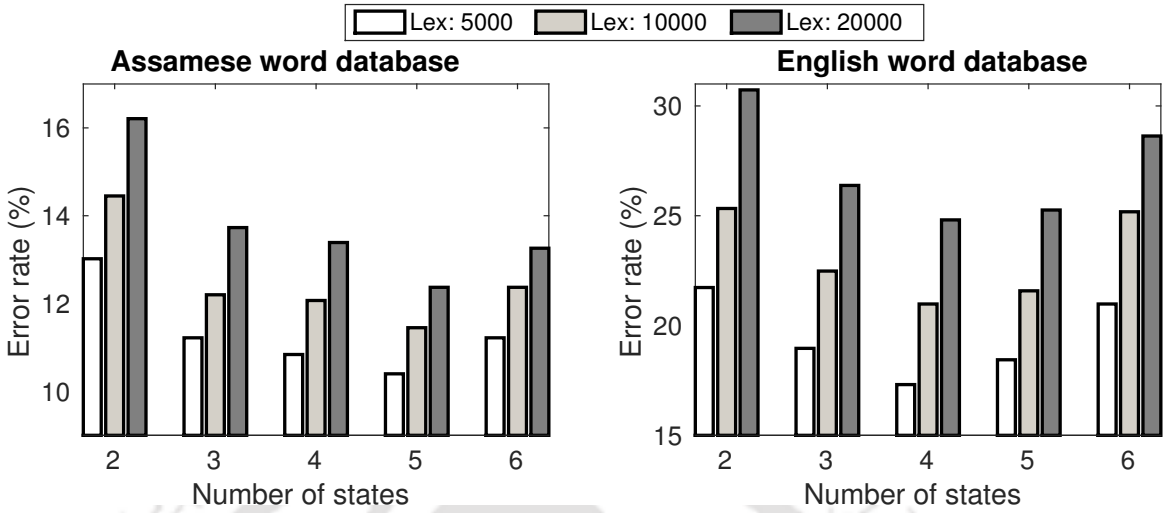


Figure 5.11: Depiction of the error rate (in %) with varying number of HMM states in the CNN feature based word recognition system. The results are shown with regard to the **validation set** of the Assamese and English database for three lexicon sizes.

Table 5.13: Error rate (in %) of the CNN feature based word recognition system, evaluated on the **test sets** of both Assamese and English databases. For comparison, the performance of point-based features is also provided.

Dataset	Lexicon size	Point-based features	CNN features			
			Net-A	*Net-B*	*Net-C*	*Net-D*
Assamese	5000	22.92	21.85	20.91	19.05	19.05
	10000	26.13	24.91	22.65	20.91	21.04
	20000	28.86	26.08	24.56	22.65	22.81
English	5000	26.27	16.30	14.93	14.35	14.35
	10000	29.65	19.22	17.14	16.84	17.14
	20000	33.52	22.36	19.57	19.22	19.35

architecture *Net-C* results in the minimum error rate.

Last but not least, in Table 5.14, we see how our proposed GMM and CNN feature based word recognition systems perform with respect to the work reported in the literature for the English dataset. The results are mentioned across different sizes of the lexicon. It can be seen that both our systems outperform this work. Further, among the two proposed systems, the improvement with the use of CNN features is comparatively higher.

5.9 Summary

In this Chapter, we propose a set of probabilistic features for online handwriting recognition that are derived from Gaussian mixture models (GMMs). The GMMs, being a generative model

Table 5.14: Performance comparison of the proposed GMM and CNN feature based systems with the literature reported work on the English word database. The results of the GMM feature based system correspond to the entries mentioned in Table 5.7.

Method	Lexicon		
	5000	10000	20000
SVM [111]	29.16	32.98	37.41
Proposed GMM feature based system	17.34	20.31	23.71
Proposed CNN feature based system	14.35	16.84	19.22

are intended to capture the class dependent characteristics. We show that the so derived posterior features aid in minimizing intra-class variability in the feature space while at the same time improving the separability between classes.

In the second part of this chapter, we extract features directly from the trace of online handwriting. In this direction, a convolution neural network (CNN) is developed to process the online handwriting data. To the best of our knowledge, this is the first work of its kind that applies CNN directly on the (x, y) coordinates of the online handwriting data.

The efficacy of the proposed GMM posterior and CNN features are demonstrated for the basic unit and word recognition tasks on the Assamese and English databases. The analysis of results indicate that the proposed features possess better discrimination among different classes when compared with the existing point-based features thus improving the recognition performance.

6

DNN-HMMs for Basic Unit Modeling

Contents

6.1 Introduction

The work presented in this chapter explores a hybrid deep neural network - hidden Markov model (DNN-HMM) framework for basic unit modeling. In a conventional HMM-based system, the Gaussian mixture model (GMM) is used to compute the value of the probability distribution of a given observation assigned to a state. Different to it, we investigate the merit of a DNN-HMM framework that is quite popular in the field of speech recognition [23]. In this approach, a deep neural network (DNN) is trained to output the posterior probabilities of the observations. The posteriors are then converted into quasi-likelihoods by dividing them with the prior of the states. These quasi-likelihoods are then used with the HMM during the time of testing of a handwritten basic unit pattern or word.

To the best of our knowledge, the DNN-HMM hybrid structure is hardly explored for online handwriting recognition. A few works have been reported with MLP-HMM framework, where an MLP is used to obtain the HMM state likelihood [118, 132–134].

The DNN-HMM can provide a powerful modeling capability when compared to the GMM-HMM. This is primarily owing to the following reasons.

- The GMMs are generative by nature, while DNNs follow a discriminative modeling paradigm. As such, they can adequately model any kind of non-linear functions of the input and hence do not require any prior assumption of the input distribution.
- Different to the GMMs, the DNNs can process high-dimensional feature vectors (obtained by considering contextual information) to make a better prediction of the input data.
- The GMMs typically make a strong assumption of diagonal covariances while dealing with high-dimensional data. The DNNs, on the other hand, do not make any such assumptions and can well handle such data with the fully connected layers.

The DNNs with many hidden layers are capable of modeling very complex and highly non-linear relationships between the input and output. To train the DNN, first, a GMM-HMM system is built. Thereafter, each of the observations in the training data are assigned to one of the HMM states. The input to the DNN is a single $[d \times (2L_c+1)]$ dimensional feature vector which is generated by combining $(2L_c+1)$ feature vectors (comprising the center feature vector with a context of L_c feature vectors at each side). The associated target label for this generated feature vector is the assignment of the state index corresponding to the center feature vector (namely, the $(L_c+1)^{th}$ feature vector). The DNN is

trained discriminatively using the backpropagation algorithm with cross-entropy cost function, where the stochastic gradient descent algorithm is used for gradient computation.

In the experimental analysis, we demonstrate the effect of several hidden layers in developing the DNN-HMM system for online handwriting. The results demonstrate that a notable improvement is achieved with the proposed DNN-HMM system when evaluated for both the basic unit and word recognition tasks.

In the final part of this contributing chapter, we combine the explorations proposed in the thesis for the basic unit recognition system. In this framework, the features derived from CNN are employed to characterize the online handwritten data. Thereafter the DNN-HMM is used to model the basic unit patterns. Reevaluation of the output class label is also performed by considering the discriminant region-based strategy of Chapter 4. The performance of the combined system is demonstrated for word recognition of English and Assamese.

Based on the above discussions, the following are the research contributes of this chapter.

- The use of DNN-HMM framework for online basic unit and large vocabulary word recognition tasks.
- Proposal of a word recognition system by integrating the CNN features, the DNN-HMM based modeling and the discriminant region-based strategy applied at the output for reevaluation.

The remainder of the chapter is organized in the following way. An overview of a DNN is first described in Section 6.2. Thereafter, in Section 6.3, we present the details of the proposed DNN-HMM framework for online handwriting recognition. This is followed by the discussion of the training of the recognition system in Section 6.4. The result and discussion of the proposed system are presented in Section 6.5. Lastly, Section 6.6 combines all the various explorations made in this thesis to develop an integrated word recognition system. In Section 6.7, we conclude the Chapter.

6.2 Overview of DNN

A deep neural network (DNN) is a feed-forward neural network that has more than one hidden layer between the input and output layers. Each unit of the hidden layer employs a logistic function to non-linearly map the d-dimensional input $\{x_1, x_2, ...x_d\}$ from the previous layer, which is then passed

to the next layer. Mathematically, we can write

$$a_j = \sum_{i=1}^{d} w_{ji}\ x_i + b$$
$$z_j = h(a_j) \tag{6.1}$$

The term b is the bias, i is the index over units in the previous layer, and w_{ji} is the weight connection between the j^{th} unit of current layer with the i^{th} unit of the previous layer. The quantities a_j are known as *activations*. The activations are transformed by using a nonlinear function $h(.)$, which are then fed to the next layer. In our work, we use the *tanh* function for h, defined as

$$h = \frac{e^{a_j} - e^{-a_j}}{e^{a_j} + e^{-a_j}} \tag{6.2}$$

In the output layer, a softmax function is introduced to obtain the probability values $\{\hat{y}_p\}_{p=1}^{C}$ of each class:

$$\hat{y}_p = \frac{\exp(a_p)}{\sum_j \exp(a_j)} \tag{6.3}$$

The DNN is trained discriminatively by back-propagating the derivative of the cost function that measures the error between the target output and the actual output produced by the network. The cross-entropy cost function $\mathcal{H}$ is considered, where

$$\mathcal{H} = -\sum_{p} y_p\ log\ (\hat{y}_p) \tag{6.4}$$

Here y_p is the target probability and $\hat{y}_p$ is the estimated probability of the p^{th} class. The stochastic gradient descent algorithm is used for the computation of the gradients in the back-propagation algorithm.

The DNNs with many hidden layers are very flexible models with a large number of parameters. They are capable of modeling very complex and highly nonlinear relationships between inputs and outputs. However, optimizing the weights become hard, mainly due to the gradient dilution problem during the back-propagation of errors through several layers. To overcome this, the network is trained in a greedy layer-wise supervised manner [135].

6.3 DNN-HMM system

The proposed DNN-HMM based online handwriting recognition system is depicted in Fig. 6.1. The system takes as input, the sequence of (x, y) points representing the trace of the input sample. It

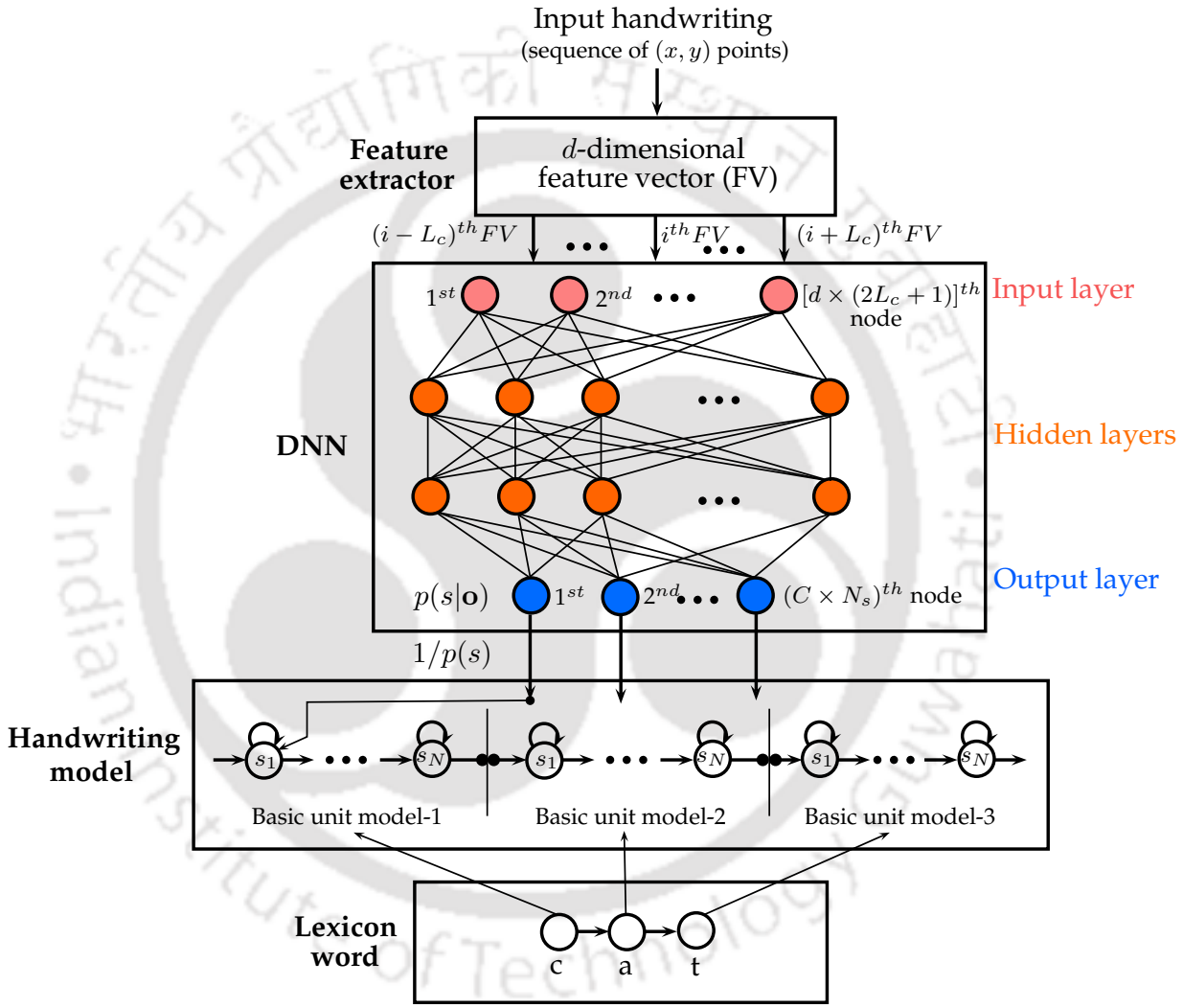


Figure 6.1: Illustration of the proposed DNN-HMM recognition system for an online handwritten word. First, the features are extracted from the trace of the input handwriting. The resulting feature vectors with contextual information are processed by the DNN. The outputs of the DNN are divided by the prior state probability and subsequently used as observation probabilities in the HMM framework. For word recognition, the handwriting model is generated by cascading the basic unit HMMs as per the entries of the lexicon.

then extracts a d-dimensional feature vector at each of the (x, y) points. We consider in total $(2L_c+1)$ feature vectors (comprising the center with a context of L_c feature vectors at each side) as input to the DNN. The network outputs state posterior probabilities which are then converted into likelihood by dividing them with the state priors. For the recognition of a word, the word HMM is created by cascading its constituent basic unit HMMs. Subsequent to it, we compute the posterior probability of each model for the given test sample (represented by $\mathbf{O} = [\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_T]$) and select the class with the highest score.

In the following, we provide details of the computation of posterior probability $P(\lambda_p|\mathbf{O})$ with the HMM framework for the basic unit pattern c_p. To start with, Bayes rule is employed that decomposes $P(\lambda_p|\mathbf{O})$ into a likelihood $P(\mathbf{O}|\lambda_p)$ that represents the contribution of the handwriting model, and a prior probability $P(\lambda_p)$

$$\hat{c} = \arg\max_{1 \leq p \leq C} P(\mathbf{O}|\lambda_p)\ P(\lambda_p). \tag{6.5}$$

In our work, we assume $P(\lambda_p)$ to be the same $\forall p$. The likelihood $P(\mathbf{O}|\lambda_p)$ is obtained by summing it over all possible state paths in λ_p that can generate $\mathbf{O}$

$$\begin{aligned} P(\mathbf{O}|\lambda_p) &= \sum_{\forall S} P(\mathbf{O}|Q, \lambda_p)\ P(Q, \lambda_p). \\ &= \sum_{\forall Q} \left[\prod_{t=1}^{T} P(\mathbf{o}_t|q_t)\ \pi_{q_1} \prod_{t=2}^{T} a_{q_{t-1}\ q_t} \right] \end{aligned} \tag{6.6}$$

where π_{q_1} denotes the probability of the initial state q_1 and $a_{q_{t-1}\ q_t}$ is the transition probability from state q_{t-1} to q_t. The underlying state sequence of λ_p to represent $\mathbf{O}$ is denoted by $S = \{q_1, q_2, ..., q_T\}$ and the total number of states considered for each HMM is N_s. For our implementation, we use the Forward algorithm [19] to compute $P(\mathbf{O}|\lambda_p)$ in Equation (6.6). The choice of the algorithm is made owing to its mathematical tractability and reduced computational burden.

The term $P(\mathbf{o}_t|q_t)$ in Equation (6.6) representing the observation probability is computed from the deep neural network (DNN) in the proposed system. In general, the DNN provides the posterior probability of the form $P(q_t|\mathbf{o}_t)$. To convert them into likelihood, we divide by the prior probability of each state that are estimated via a frequency based approach during the forced alignment of training data. Accordingly, we can now write

$$P(\mathbf{o}_t|q_t) = \frac{P(q_t|\mathbf{o}_t)P(\mathbf{o}_t)}{P(q_t)} \tag{6.7}$$

where $P(q_t|\mathbf{o}_t)$ is the posterior probability obtained from the DNN, $P(q_t)$ is the prior probability of the state q_t. Moreover, $P(\mathbf{o}_t)$ is a normalization independent of the model and hence can be ignored.

6.4 Training

A deep neural network - hidden Markov model (DNN-HMM) is a hybrid framework where the HMM state observation probabilities are computed through the DNN at the time of testing. To train the hybrid system, first we build a conventional HMM-based system where the emission probabilities are computed through the Gaussian mixture model (GMM). It may be recalled from Section 3.2.3 that we build C HMMs denoted by λ_p, $p = [1, 2, ..., C]$ for the basic unit recognition system having C basic unit classes. In the case of word recognition, the HMMs corresponding to the basic units present in the transcription are concatenated to form the word HMM.

For a given HMM, we can find the most likely sequence of states given a sequence of d-dimensional feature vectors $\mathbf{O} = [\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_T]$ by using the Viterbi algorithm. This is also referred to as forced alignment. Said in another way, this technique is used to assign a state label or index to each feature vector $\mathbf{o}_t$ associated with the training dataset. The DNN is trained on these feature vectors by using their corresponding HMM state assignments as target labels.

The input nodes to the DNN correspond to the dimension of handwriting features multiplied with the number of context feature vectors. We consider a context of $(2L_c + 1)$ feature vectors (center, and a context of L_c feature vectors at each side) to train the DNN. Accordingly, the input to the DNN is a single $[d \times (2L_c + 1)]$-dimensional feature vector, which is generated by combining the $(2L_c + 1)$ context feature vectors. The associated target label for this generated feature vector is the state assignment corresponding to the center feature vector (namely, the $(L_c + 1)^{th}$ feature vector).

The number of output nodes in the DNN is determined by the total number of HMM states present in the system. Said in another way, if N_s denote the number of states associated with an HMM and there are C classes present in the system, then the number of output nodes is $(C \times N_s)$. The number of hidden layers and their dimension in the DNN are optimized for optimal performance. The DNN is trained by following the methodology given in the previous section.

It is important to note that, unlike the GMM-HMM training where we realign the transcriptions with the data in each iteration, we train the DNN only once with the aligned data. Hence, the initial probabilities of the states Π and the transition matrix A in the DNN-HMM system are taken directly

Table 6.1: Performance evaluation of the DNN-HMM configuration on the Assamese modified character **validation set**. In particular, we provide the error rates for varying number of hidden layers, nodes in the layer and HMM states. The minimum error rate achieved is denoted in **bold**. For comparison, the performance of the GMM-HMM system is also reported.

HMM state	GMM-HMM	No of nodes in hidden layer	Number of hidden layer (H)					
			1	2	3	4	5	6
13	4.07	400	3.47	2.83	2.52	2.18	2.31	2.52
		500	3.26	2.52	2.18	2.18	2.44	2.70
		600	2.97	2.31	2.18	2.31	2.52	2.83
15	3.76	400	3.26	2.70	2.18	2.18	2.18	2.52
		500	2.97	2.44	**1.94**	2.18	2.31	2.52
		600	2.70	2.18	2.10	2.18	2.44	2.83
17	3.60	400	3.10	2.52	2.31	2.10	2.44	2.70
		500	2.83	2.44	2.18	2.31	2.52	2.83
		600	2.83	2.31	2.18	2.52	2.70	2.97

from the GMM-HMM system.

For a given test sample, the DNN first provides the state posterior probabilities for each of the T observations. Thereafter, these are converted to likelihoods which can then be employed to compute Equation (6.5) across all the HMMs to determine the identity of the class.

6.5 Result and discussion

In this section, we evaluate the efficacy of the proposed DNN-HMM framework on the Assamese and English databases (discussed in Section 1.4). The training set is used to create the DNN-HMM models, whereas the validation and the test sets are used to optimize and evaluate the system respectively. For our implementation of the DNN-HMM architecture, we rely on Kaldi [136] - a speech recognition toolkit.

6.5.1 Basic unit recognition

To begin with, we judge the performance of the proposed DNN-HMM system for varying number of HMM states, hidden layers and the nodes within them [1]. The input nodes of the DNN consists of $(2L_c+1)$ consecutive feature vectors (each of 14 dimensions), with the value L_c determining the contextual information to be used. In this experiment, we set the value of L_c to 4 for the Assamese modified character and English lowercase letter datasets. The results on the validation set are presented in Tables 6.1 and 6.2 respectively.

[1] In the DNN setup, we consider equal number of nodes in each of the hidden layers.

Table 6.2: Performance evaluation of the DNN-HMM configuration on the English lowercase **validation set**. In particular, we provide the error rates for varying number of hidden layers, nodes in the layer and HMM states. The minimum error rate achieved is denoted in **bold**. For comparison, the performance of the GMM-HMM system is also reported.

HMM states	GMM-HMM	No of nodes in hidden layer	Number of hidden layer (H)					
			1	2	3	4	5	6
9	6.97	200	6.10	5.60	5.14	4.85	4.96	5.14
		300	5.60	4.96	4.65	4.34	4.73	5.21
		400	5.14	4.47	4.34	4.26	4.85	5.29
11	6.40	200	5.92	5.39	4.96	4.47	4.73	4.96
		300	5.60	4.85	4.57	**4.14**	4.57	5.14
		400	5.14	4.65	4.47	4.65	4.96	5.34
13	6.32	200	5.81	5.60	5.29	4.96	5.14	5.34
		300	5.60	5.14	4.85	4.57	4.96	5.39
		400	5.21	4.73	4.65	4.96	5.21	5.60

Table 6.3: Performance evaluation of the DNN-HMM configuration on the different **validation sets**. The number of hidden layers, nodes in the layer, L_c value and the HMM states corresponding to the reported minimum error rate are given. For completeness, the performance of the GMM-HMM is also indicated.

Dataset	GMM-HMM	DNN-HMM			
		# of HMM states	(H, node)	L_c	Error rate
Assamese digit	1.23	15	(3,200)	3	0.49
Assamese basic character	3.65	15	(3,300)	4	1.71
Assamese modified character	3.76	15	(3,500)	4	1.94
English digit	1.25	11	(4,200)	3	0.75
Uppercase letter	3.90	11	(4,300)	4	2.61
Lowercase letter	6.40	11	(4,300)	4	4.14

Based on the entries of the tables, the following can be inferred:

- With the increase of number of hidden layers, the error rate decreases across all the HMM states. However, beyond a certain number (3 and 4 for the modified character and lowercase datasets), there is hardly any improvement despite the increase in the learnable weight parameters.
- On the other hand, when the network has fewer hidden layers (e.g. 1 or 2), the addition of nodes in a layer helps in improving the performance.
- The lowest error rate obtained for the framework on the Assamese modified character and English lowercase datasets are 1.94% and 4.14% respectively.

Table 6.3 reports the best performance on the different validation datasets for the basic unit recognition task. For each case, the optimized number of HMM states, number of hidden layers

Table 6.4: Performance comparison of the proposed DNN-HMM based system with the GMM-HMM on the different **test sets**.

Dataset	GMM-HMM	DNN-HMM
Assamese digit	1.13	0.46
Assamese basic character	3.87	1.80
Assamese modified character	4.00	2.03
English digit	1.13	0.63
Uppercase letter	3.63	2.65
Lowercase letter	6.82	4.74

Table 6.5: Performance comparison (error rate in %) with other works reported on (a) Assamese character and (b) English UNIPEN character datasets.

(a)

Method	Digit	Basic character	Modified character
Two-stage system [110]	1.20	-	4.30
HMM and SVM combination [25]	1.70	-	-
Proposed system	0.46	1.80	2.03

(b)

Method	Digit	Uppercase	Lowercase
DTW [32]	2.90	7.20	9.30
OnSNT [27]	1.10	4.30	7.90
ANN [1]	0.80	3.10	5.10
HMM [8]	1.73	-	-
Proposed system	0.63	2.65	4.74

and the nodes in each layer, value of L_c are explicitly specified. Further, the performance of the GMM-HMM system is also provided. An improvement in error rate is observed with the proposed DNN-HMM framework across all the character datasets.

Table 6.4 presents the performance of the proposed DNN-HMM based basic unit recognition system evaluated on the test sets of various character datasets. The parameters learned in the validation set are used for judging the system efficacy on the test data. For comparison, the performance of the baseline GMM-HMM system is also reported. It can be seen that a notable reduction in the error rate is achieved with the proposed method, thereby demonstrating the effectiveness of proposed system.

Finally, a performance comparison of the proposed DNN-HMM based system with prior works reported on Assamese and UNIPEN character dataset is presented in Table 6.5. It can be seen that comparable performance is achieved with the proposed system.

Table 6.6: Error rate (in %) of the proposed DNN-HMM system for the **test sets** with varying lexicon size and hidden layers. The performance of the GMM-HMM is also provided for comparison.

Dataset	Lexicon	GMM-HMM	# of hidden layers in DNN-HMM					
			1	2	3	4	5	6
Assamese	5000	22.92	13.96	12.32	11.38	11.07	11.51	12.19
	10000	26.13	14.74	13.06	12.19	11.76	12.19	12.83
	20000	28.86	15.77	13.96	12.83	12.32	12.97	13.47
English	5000	26.27	15.54	14.17	13.64	13.18	13.64	14.35
	10000	29.65	18.13	16.84	15.95	15.54	16.30	16.84
	20000	33.52	20.16	18.51	18.13	17.72	18.26	18.97

6.5.2 Word recognition

In this subsection, we evaluate the proposed DNN-HMM system for the word recognition task. Table 6.6 presents the error rate obtained for different lexicon size on the test set of Assamese and English datasets. The number of hidden layers are also varied from 1 to 6. For comparison, the performance of the baseline GMM-HMM system is also provided.

In the proposed system, we use $(2L_c + 1)$ consecutive feature vectors as input to the DNN where the value of L_c is set to 4 for both the word datasets. Moreover, the number of nodes are kept constant across the hidden layers with 500 and 300 for the Assamese and English datasets.

From the entries in the table, it can be inferred that the use of more hidden layers results in reduction of the error rate. Said in another way, a sufficient number of trained weights across the layers can better model the handwritten data. However, beyond a certain number of hidden layers, the error rate starts increasing due to the possible problem of over-fitting. It is worth noting that the DNN-HMM outperforms the GMM-HMM system across the datasets with different sizes lexicons. In particular, four hidden layers in the DNN is found to provide the lowest error rate.

Moving further, it can be seen that, with the increasing size of the lexicon, the error rate increases. This can be attributed to the fact that larger lexicons can result in a greater degree of structural similarity among the words. This, in turn, leads to generating models with similar characteristics, that are susceptible to a higher number of mis-classifications, especially with challenging test samples. However, this effect is alleviated by the DNN-HMM system when compared to the GMM-HMM.

Finally, we compare the word recognition result of English dataset with the literature reported performance in Table 6.7. By comparing, it can be seen that, the proposed system has reduced the error rate by 15.98%, 17.44% and 19.69% for the lexicons of 5000, 10000 and 20000 words respectively.

Table 6.7: Performance comparison of the proposed system with the literature reported work on English word dataset.

Method	Lexicon		
	5000	10000	20000
SVM [111]	29.16	32.98	37.41
Proposed system	13.18	15.54	17.72

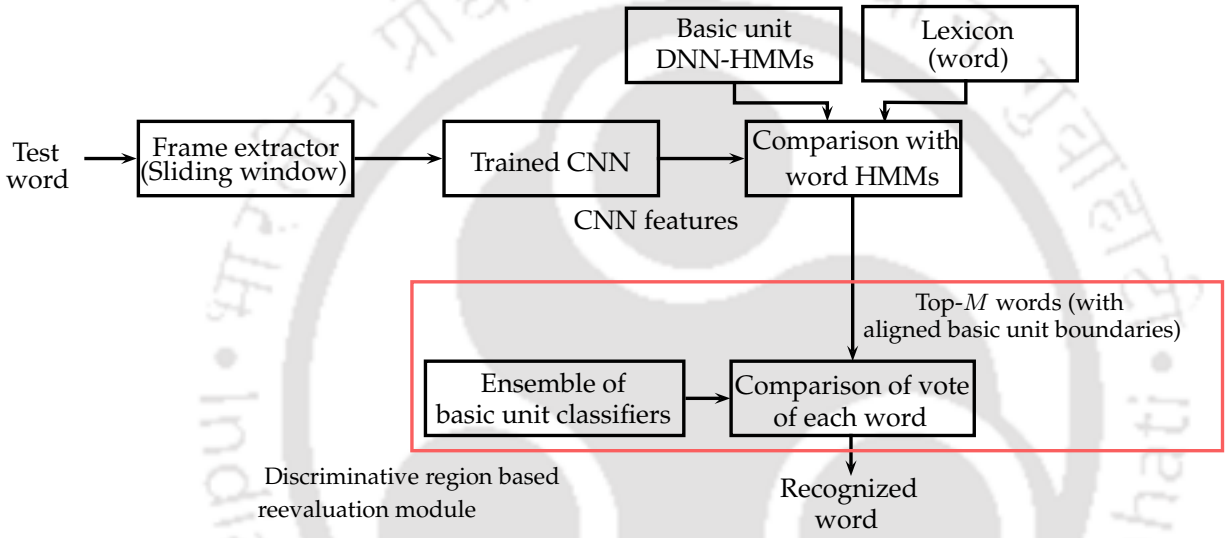


Figure 6.2: Block diagram representing the combined framework to develop a word recognition system.

6.6 Combined framework for word recognition

As a final experiment of the thesis, we integrate all the explorations proposed thus far to develop a word recognition system. Fig. 6.2 shows a block diagram of the framework. For a given online handwritten word, the CNN features are extracted from the sample by employing a set of frames. Thereafter, they are fed to the DNN-HMM framework for recognition. The discriminative region analysis discussed in Chapter 4 is applied at the output of DNN-HMM for revising the decision of the word recognition.

It is to be noted that, we could as well have used other two proposals namely, the GMM posterior features and the discriminative states for reevaluation in the combined system. However, our choice of CNN features with discriminant region technique for reevaluation is owing to their relatively better performance.

Table 6.8: Performance of the combined framework on the **test sets**. Note that the reevaluation scheme corresponds to the discriminant region-based single-stage classifier discussed in Chapter 4 of the thesis.

Dataset	System	Lexicon size		
		5000	10000	20000
Assamese	Point-based features and DNN-HMM classifier [2]	11.07	11.76	12.32
	CNN features + DNN-HMM classifier	9.57	10.32	11.33
	CNN features + DNN-HMM classifier with reevaluation	9.06	9.62	10.57
English	Point-based features and DNN-HMM classifier [2]	13.18	15.54	17.72
	CNN features + DNN-HMM classifier	9.13	11.48	13.82
	CNN features + DNN-HMM classifier with reevaluation	8.31	10.44	12.60

Table 6.8 reports the performance of the combined framework for the Assamese and English word datasets. Further, we also demonstrate the contribution of each of the explored direction in the table. To begin with, we consider only the CNN features and the DNN-HMM classifier to develop the system. The results show that this combination achieves improvement of 1.50%, 1.44%, and 0.99% in the error rate for the lexicon size of 5000, 10000 and 20000 respectively in Assamese dataset. Thereafter, the use of discriminant region-based reevaluation technique for revising the decision of the classifier gives a further improvement of 0.51%, 0.70%, and 0.76% for the respective lexicons. Last but not least, we can also note from the table that the combined system outperforms the DNN-HMM framework based on the 14 dimensional point based features of Section 3.3.2.

Moving now to the English dataset, a similar trend in performance can also be observed. In this case, the improvement achieved with the CNN features and the DNN-HMM classifier is 4.05%, 4.06%, and 3.90% respectively for the lexicon sizes of 5000, 10000 and 20000. Subsequent to the reevaluation of the labels with information from the discriminant region, a further improvement of 0.82%, 1.04%, and 1.22% is obtained.

Lastly, for the English dataset, on comparing the result in Table 6.8 for the combined system with the literature reported performance [111] (Table 6.7), we can see a commendable improvement of 20.85%, 22.54%, and 24.81% for the lexicon sizes of 5000, 10000 and 20000 words respectively.

[2]This system has been described in the first part of Chapter 6. It provides the lowest error rate over our proposals described in the preceding three contributing Chapters.

6.7 Summary

In this chapter, we have presented a hybrid DNN-HMM system for recognizing online handwritten basic unit and words. The key feature of the proposed system is that a DNN is used to estimate the emission probabilities of HMM. The DNNs with many hidden layers are capable of learning very complex and highly non-linear relationships between the input and output. They can also incorporate high-dimensional feature vectors (obtained by considering contextual information) which help the system to make a better prediction of the input data. The proposed DNN-HMM outperformed the baseline GMM-HMM system on the Assamese and UNIPEN English character and word datasets.

In the final part of this contributing chapter, we combine the explorations proposed in the thesis. In this framework, the CNN features are used to develop a DNN-HMM based word recognition system. Thereafter, the discriminant region-based strategy is applied at the DNN-HMM output for reevaluation. The performance of the combined system is demonstrated for word recognition. Our results show that such a scheme is quite promising with regard to reducing the error rate.

7

Summary

Contents

7.1 List of contributions

The different contributions made in this thesis are summarized below.

(i) In an HMM-based online handwriting recognition system, we demonstrate in Chapter 3, that, at times, the consideration of log-likelihood scores pertaining to each of the HMM states may not be effective in capturing the finer nuances of the online trace that discriminate similar shape patterns / basic units.

(ii) Proposal to analyze the HMM states corresponding to the top-2 confusion classes with the objective of identifying a subset of them (referred to as 'discriminative states') for revising the likelihood scores.

(iii) Proposal to reevaluate the decision of an HMM-based handwriting recognition system in a single stage classification framework.

(iv) Demonstration of the efficacy of the aforementioned proposal for basic unit and large vocabulary online handwritten word recognition tasks.

(v) Proposal of a novel strategy in Chapter 4 to detect parts of the trace (referred to as Discriminative Region) that present fine structural differences in similar looking basic units.

(vi) Demonstration of a single-stage classification framework that takes into consideration the discriminative region extracted between the basic units.

(vii) Development of an HMM-based large vocabulary word recognition system by incorporating the aforementioned discriminative region analysis, for revising the word recognition output.

(viii) Proposal of a set of probabilistic features in Chapter 5 (referred to as 'posterior features') that derived from a set of Gaussian mixture models (GMMs). The proposed features can well represent the inter-class variability due to the class-conditional probabilities from the GMMs.

(ix) Demonstration of the proposed GMM features in minimizing the intra-class variability in the feature space, while improving the separation between classes.

(x) Proposal of CNN features to describe the trace of online handwriting data.

(xi) Development of various CNN architectures that can extract features directly from online handwriting (2D sequential data), without the need of conversion to an off-line image.

(xii) Development of a basic unit and a large vocabulary online handwritten word recognition systems with the proposed GMM-based features and CNN features.

(xiii) The utility of DNN-HMM framework for online handwritten basic unit and large vocabulary word recognition tasks. This is discussed in Chapter 6.

(xiv) Development of a word recognition system by integrating the directions explored in this thesis, namely, the CNN features to represent online handwriting, the DNN-HMM framework for classifier modelling and the discriminant-based strategy at the output for reevaluation.

7.2 Summary of results

We now summarize the performance obtained from each of the contributing chapters in Tables 7.1 and 7.2, for the basic unit recognition and word recognition tasks, respectively. For ease of comparison, the results of the baseline classifier for each chapter are also reported.

The performance of each of our proposals is judged in terms of the percentage of wrong classifications (error rate). All our results are reported on two online handwritten databases, namely

- the locally collected Assamese character and word datasets
- the publicly available English UNIPEN character and UNIPEN-ICROW-03 word datasets.

Moreover, for the English database, the performance reported by prior works in the literature are also presented in Table 7.3. The entries are the classification error rates as mentioned from the respective references. It may be noted that the systems being outlined can differ with regard to the employed feature set and classifier. Therefore, it is to be borne in mind that a direct one to one comparison may not be fair. Nevertheless, with regard to these explorations, our proposals in this thesis provide a competent performance at the basic unit and word recognition tasks.

Table 7.1: The sub-table (a) captures the summary of results corresponding to the **test sets** of the proposed basic unit recognition systems from each of the contributing Chapters 3 to 6. Recall in Chapter 4 and 5, that apart from the HMM, we have used the SVM as baseline classifier to build the system and subsequent the proposals. Hence, in a separate sub-table (b), we report the performance for the same.

(a)

Database	Baseline	Chapter 3	Chapter 4	Chapter 5	Chapter 6
Assamese digit	1.13	0.75	0.56	0.56	0.46
Assamese basic character	3.87	3.13	2.89	2.51	1.80
Assamese modified character	4.00	3.15	2.98	2.70	2.03
English digit	1.13	0.86	0.86	0.73	0.63
Uppercase	3.67	3.15	2.96	2.85	2.65
Lowercase	6.82	6.18	5.98	5.86	4.74

(b)

Database	Baseline	Chapter 4	Chapter 5	
			GMM features	CNN features
Assamese digit	0.75	0.56	0.46	0.46
Assamese basic character	4.60	3.71	2.84	2.29
Assamese modified character	6.39	4.76	3.00	2.45
English digit	1.21	0.90	0.73	0.63
Uppercase	3.09	2.49	2.33	2.16
Lowercase	5.05	3.92	3.64	3.45

Table 7.2: Summary of results corresponding to the **test sets** of the proposed word recognition systems of the thesis for different sizes of lexicon.

	System	Assamese			English		
		5000	10000	20000	5000	10000	20000
	Baseline	22.92	26.13	28.86	26.27	29.65	33.52
Chapter 3	Reevaluation	20.57	23.72	25.84	23.89	26.83	29.34
Chapter 4		19.85	23.00	25.32	23.43	26.27	28.98
Chapter 5	GMM features	19.47	21.70	23.35	17.34	20.31	23.71
	CNN features	19.05	20.91	22.65	14.35	16.84	19.22
Chapter 6	DNN-HMM	11.07	11.76	12.32	13.18	15.54	17.72
	Combined framework	9.06	9.62	10.57	8.31	10.44	12.60

7.3 Possible pointers for the future

We conclude the thesis by presenting a number of research directions that can be taken up in the near future.

- Recall from Chapter 3 that the reevaluation of the HMM based online handwriting recognition system is based on the analysis of discriminant HMM states between the confused basic units. However, the log-likelihood score computation of the HMM state can be obtained through the

Table 7.3: Survey of online handwriting recognition systems on the UNIPEN character and ICROW-03 word datasets. The numbers corresponds to the classification error rates (in %) as reported from the respective references.

Dataset	Method	Digit	Uppercase	Lowercase
English character	DTW [32]	2.90	7.20	9.30
	OnSNT [27]	1.10	4.30	7.90
	ANN [1]	0.80	3.10	5.10
	HMM [8]	1.73	-	-
English word		Lexicon size		
		5000	10000	20000
	SVM [111]	29.16	32.98	37.41

DNN as discussed in Chapter 6. Thus, an interesting exploration would be to compute the log-likelihood scores of the discriminative HMM states via a DNN architecture for reevaluation of the classifier decision.

In the present work, all the basic units of a given script are modelled with same number of states. As a future direction, we can explore the selection of discriminative states under the scenario of modeling each basic unit HMM with different number of states. The states may be chosen based on the complexity of the confusion pairs under consideration.

- In Chapter 4 of the thesis, we proposed a discriminative region selection technique that detects parts of the trace that present fine structural differences in similar looking basic units / patterns. It is to be emphasized that the proposed technique assigns contiguous (x, y) points of the trace as the discriminative region. However, there can be multiple discriminative regions present between the confused pair of basic units. Thus, an effort can be made to capture such regions with a goal of improving the classification further.

- The classification framework proposed in Chapter 4 takes into consideration same set of features across the discriminative regions being extracted. An interesting exploration would be to look for features that are specific pertaining to a discriminative region of the confused pairs under question. It is hoped that in doing this, the error rate of the system can be lowered further.

- The proposed GMM / CNN feature description in Chapter 5 have a dimension equal to the number of classes. This can be computationally expensive for the scripts comprising a large number of basic unit classes [14]. As an alleviation to this, we can consider strategies for reducing the dimension of the proposed feature vector prior to classification.

- In Chapter 6, we propose a hybrid DNN-HMM based online handwriting recognition system where the deep neural network (DNN) is used to compute the log-likelihood score of HMM state. It would be interesting to see how the performance of the hybrid system varies with other deep architectures such as Long Short-Term Memory (LSTM), which can be used to model the HMM states.

- The strategies proposed in this thesis consider the case of single-script handwriting recognition where the writers have written in one language only. A possible investigation would be to analyze them in a multi-script handwriting recognition scenario [137].

- Last but not least, some of the ideas explored in this thesis may be attempted for the task of offline handwriting recognition. Since the data to be processed represent the gray-level information corresponding to the pixels occupied by the ink, we would have to extract a different set of features for classification [138].

A

Database Analysis

Contents

A.1 List of Assamese words

1. ব্রহ্মা	2.উজ্জ্বলিত	3. ঔষধ	4. শ্রেষ্ঠ
5. পূজ্য	6. কর্ম্ম	7. হৃষ্ট	8. চম্পা
9. গঙ্গা	10. বিপ্লৱ	11. শান্তি	12. উদ্ভিদ
13. ক্রম	14. নক্সা	15. ৰিক্সা	16. লক্ষণ
17. গৃহস্থ	18. কথ্য	19. জ্ঞানী	20. সূক্ষ্ম
21. বুদ্ধ	22. ভগ্ন	23. খন্থৱা	24. ঘঁহনি
25. বঙলা	26. ঊনপঞ্চাশ	27. ছয়	28. শৃঙ্খল
29. উৎপন্ন	30. আজ্ঞা	31. মন্ত্র	32. সম্ভ্রান্ত
33. বাঞ্ছা	34. আনন্দ	35. ভ্রমণ	36. ফুল
37. সংখ্যা	38. তেঁও	39. ঐশ্বর্য্য	40. তীক্ষ্ণ
41. অনুকৰণ	42. আৰ্জ্জী	43. চঞ্চল	44. ঘূৰণীয়া
45. চন্দ্রমুখ	46. ঊনসত্তৰ	47. ঈশ্বৰ	48. স্ফূর্ত্তি
49. ঘেঁহু	50. জন্মগত	51. ছদ্মবেশ	52. ছত্র
53. ছাত্র	54. ঠৰঙা	55. ঠাই	56. ঠেকেচা
57. দৃঢ়তা	58. হৃষ্টপুষ্ট	59. হৃদয়	60. ভ্রু
61. ধর্মশাস্ত্র	62. ফচল	63. ফুলা	64. ফণি
65. লক্ষ্য	66. ক্ষমতা	67. ভকতি	68. ঈগল
69. ভ্রুক্ষেপ	70. ভ্রম	71. সংস্থাপন	72. সঙ্ঘ
73. সন্ধ্যা	74. উৎসর্গ	75. উল্লেখিত	76. অঞ্চল
77. ডৰে	78. ব্রহ্মাণ্ড	79. ব্রহ্মজ্ঞানী	80. সম্ভাৱনা
81. বিদগ্ধ	82. সম্ভ্রম	83. মন্ত্রিত্ব	84. দাণ্ডি
85. লগ্ন	86. ব্যাপক	87. জিম্মা	88. কীর্ত্তন
89. শুন	90. খং	91. নির্ম্মল	92. শঙ্খ
93. বিশেষ	94. মনোৰঞ্জন	95. সঞ্জীৱনী	96. বাঞ্ছিত

97. মৃত্যুঞ্জয়
98. অনুষ্ঠান
99. ঘণ্টা
100. অর্থাৎ
101. শত্রু
102. বিদ্রুপ
103. ক্রুদ্ধ
104. নাট্য
105. ধ্বংস
106. পর্ব্বত
107. অজ্ঞানী
108. গ্রন্থ
109. বিপ্লব
110. বিজ্ঞান
111. ব্রহ্ম
112. মন্ত্রী
113. স্বাস্থ্য
114. দিওঁ
115. ভক্তি
116. মন্ত্র
117. শাস্ত্র
118. ধর্ম্ম
119. আকৌ
120. ঋষি
121. এক্কণমানি
122. ইন্সপেক্টৰ
123. টেক্বনিক
124. ৰুক্ষক
125. শুক্ল
126. শুচি
127. ভাগ্যবতীয়ে
128. ৰণ্ণন্তি
29. ঝঙ্কাৰ
130. আকাঙ্ক্ষা
131. চ্চষ্টা
132. তুচ্ছ
133. কুজ্ঝটিকা
134. জ্বলি
135. হউগোল
136. আড্ডা
137. কণ্ঠেৰে
138. তত্ত্ব
139. থৃহবষ্ক
140. যন্ন
141. আত্মা
142. উদ্দেশ্য
143. দ্বন্দ্ব
144. গণ্য-মান্য
145. স্ক্রীপ্ট
146. সমাপ্ত
147. স্নপ্ননীলে
148. অন্সৰী
149. শব্দ
150. নিস্তব্ধতা
151. ৰ্লটিং
152. নিম্ন
153. সাম্ফল
154. সম্বন্ধ
155. হিংস্র
156. ছিল্কৰ
157. পল্টন
158. হল্ডঃ
159. কল্পনা
160. সান্ত্বনা
161. ল্মধু
162. নিশ্চয়
163. প্রশ্ন
164. শ্মশান
165. কাবশ্ন
166. তেজষ্ক্রিয়
167. পুষ্পক
168. নিষ্ফল
169. ভীষ্ম
170. স্কুল
171. সমস্তে
172. বস্তু
173. স্পষ্ট
174. বিস্ময়
175. এহুড়
176. তৈহূ
177. কৃষ্ণচ
178. চিঞৰি
179. নির্বিঘ্নে
180. ম্লান
181. স্খলিত
182. সান্ত্বনা

A.2 List of English words

1. a	2. abbandono	3. abdomen	4. abilitate	5. about	6. abstinent	7. accendono
8. access	9. accessori	10. accompagnata	11. accumulare	12. aches	13. acre	14. acustici
15. adeguare	16. adherent	17. adjunct	18. adult	19. advocate	20. affluire	21. afghanistan
22. again	23. aggiungi	24. air	25. aiuto	26. album	27. aldehyde	28. algebra
29. algoritmiche	30. all	31. allegati	32. alluvium	33. alp	34. also	35. alta
36. alto	37. amanuensis	38. ammettere	39. an	40. analizzata	41. analyst	42. and
43. andato	44. anecdote	45. angst	46. anomalia	47. another	48. antecedent	49. aorta
50. aperture	51. apparterranno	52. appendix	53. apprendimento	54. aprire	55. aqua	56. arcsin
57. are	58. aritmetico	59. arrotondati	60. as	61. assegnati	62. asserviti	63. associate
64. at	65. atta	66. attivati	67. attribuito	68. auschwitz	69. ausilio	70. autonoma
71. avanzata	72. avvenire	73. avviati	74. away	75. azzeramenti	76. back	77. backup
78. bad	79. badminton	80. banden	81. Banden	82. bangkok	83. basano	84. batik
85. bauhaus	86. bazaar	87. be	88. beach	89. because	90. been	91. begrip
92. Begrip	93. bell	94. bhagwan	95. bieden	96. Bieden	97. bijouterie	98. bilancio
99. bill	100. bladder	101. bobby	102. bodyguard	103. bolster	104. borax	105. borden
106. Borden	107. bordi	108. bouquet	109. boutique	110. bradford	111. breakdown	112. brisbane
113. brown	114. Brown	115. budget	116. buffet	117. Builen	118. buiten	119. but
120. by	121. byte	122. calante	123. calcium	124. call	125. cambiare	126. camera
127. can	128. capire	129. car	130. caratterizzati	131. cardio	132. Cardio	133. care
134. cassetti	135. catch	136. cede	137. centen	138. Centen	139. centrata	140. cetera
141. charisma	142. checklist	143. cheque	144. chevron	145. chilometro	146. chloride	147. cinquantesima
148. citrus	149. Citrus	150. classificata	151. cockpit	152. cocktail	153. Codeur	154. codeurs
155. coefficienti	156. coincisa	157. colic	158. collegare	159. colonnade	160. comandare	161. comfort
162. commentati	163. compare	164. compilata	165. completamento	166. concatenano	167. concludere	168. concubine
169. condizionare	170. configuratori	171. conjunct	172. connessa	173. consegnato	174. considerato	175. consueto
176. conteggiano	177. contigua	178. contrassegnata	179. controllati	180. convertitori	181. coppie	182.copywriter
183. cornwall	184. corps	185. corretto	186. costituente	187. cowboy	188. crawl	189. creata
190. croquet	191. cursus	192. Cursus	193. cycle	194. czerny	195. dad	196. dai
197. darwin	198. dashboard	199. day	200. dead	201. deadline	202. dealloca	203. debugger
204. decodifica	205. decrementi	206. dejeuner	207. del	208. delhi	209. delinquent	210. enominata
211. deodorant	212. derivato	213. destinare	214. dettagliati	215. diagnose	216. dice	217. diet
218. differenzia	219. digitate	220. dipartimenti	221. direttive	222. disattivato	223. disease	224. disegni
225. disjunct	226. disposti	227. distribuiti	228. diventate	229. dixieland	230. dizzy	231. do
232. doctor	233. documentate	234. dog	235. Dog	236. doh	237. doll	238. don't
239. dove	240. dozen	241. drink	242. durante	243. eat	244. economiche	245. edelweiss
246. effettuata	247. egg	248. elementare	249. elettrostatico	250. elite	251. emergency	252. emetic
253. emettono	254. entertainment	255. entrato	256. equilibrium	257. equipment	258. ereditare	259. esattezza
260. escono	261. esista	262. esperienza	263. espressioni	264. essay	265. estrai	266. eve
267. eventi	268. evoluto	269. excellent	270. exodus	271. export	272. extract	273. exuberant
274. eye	275. face	276. fail	277. falsa	278. fascist	279. fatsoen	280. Fatsoen
281. Fe	282. feedback	283. feel	284. feest	285. Feest	286. ferma	287. fever
288. figuur	289. Figuur	290. fill	291. finland	292. fino	293. fjord	294. flipflop
295. fondi	296. for	297. formattati	298. forzare	299. fourage	300. Fourage	301. fox
302. Fox	303. frankfurt	304. from	305. fruit	306. fuchsia	307. funge	308. fysiek
309. Fysiek	310. gain	311. gap	312. generalizza	313. generico	314. genre	315. giacca

316. girl 317. giustificazioni 318. give 319. gladiator 320. god 321. grave 322. great
323. green 324. guyana 325. gymnast 326. habit 327. had 328. hagedis 329. Hagedis
330. halfback 331. halve 332. hamster 333. hanno 334. has 335. have 336. he
337. heer 338. Heer 339. hello 340. her 341. here 342. hernia 343. heroin
344. herpes 345. him 346. hinder 347. Hinder 348. his 349. hoffman 350. hortensia
351. Hortensia 352. hotdog 353. huisje 354. Huisje 355. hulk 356. huxley 357. hyena
358. hypotheses 359. I 360. ibrida 361. ice 362. identificatrici 363. if 364. ill
365. illustrati 366. I'm 367. immigrant 368. immune 369. immutate 370. impedenza 371. impiegato
372. imporre 373. imprevisto 374. in 375. inches 376. inchiostro 377. income 378. incompleto
379. inconvenient 380. incorrere 381. indicate 382. indirizza 383. inerenti 384. inexact 385. informant
386. informativo 387. inhumane 388. iniziano 389. input 390. inserisce 391. integra 392. interazioni
393. interessanti 394. intermittente 395. interrogano 396. interventi 397. interviews 398. into 399. introduzioni
400. invertono 401. iodine 402. is 403. isolano 404. israeli 405. istanbul 406. it
407. iterativa 408. its 409. jacques 410. jitter 411. join 412. jujube 413. jumped
414. Jumped 415. jury 416. just 417. kafka 418. kamchatka 419. katoen 420. Katoen
421. keep 422. keren 423. Keren 424. keyboard 425. kidnapping 426. kien 427. Kien
428. kiwi 429. knee 430. know 431. knowhow 432. kocher 433. korsten 434. Korsten
435. kremlin 436. kuis 437. Kuis 438. lack 439. lamp 440. landcode 441. landing
442. Landing 443. larghezza 444. larynx 445. lazy 446. Lazy 447. left 448. leg
449. lenig 450. Lenig 451. lethal 452. lice 453. life 454. limitate 455. lincoln
456. liste 457. listig 458. Listig 459. loting 460. Loting 461. lunchroom 462. lunga
463. lusten 464. Lusten 465. luxe 466. macbeth 467. magnetolettura 468. magtape 469. major
470. man 471. mandje 472. Mandje 473. manipolano 474. many 475. marcati 476. masker
477. massimi 478. maxwell 479. mazurka 480. me 481. meester 482. Meester 483. megahertz
484. memorandum 485. messo 486. metal 487. microprogrammi 488. minder 489. Minder 490. minimum
491. minor 492. mode 493. moltiplicano 494. montato 495. more 496. mosterd 497. Mosterd
498. motion 499. multiplate 500. multiple 501. munten 502. Munten 503. muscle 504. my
505. mysteries 506. nail 507. native 508. neck 509. nera 510. never 511. new
512. newton 513. nihilist 514. no 515. nona 516. not 517. now 518. numerano
519. nurse 520. oak 521. object 522. obsoleta 523. oceano 524. of 525. offer
526. ohm 527. OK 528. omogenei 529. on 530. one 531. onyx 532. open
533. opportuno 534. optimum 535. or 536. orbit 537. order 538. ordinatori 539. orientata
540. ortografici 541. other 542. otterrebbe 543. our 544. out 545. over 546. Over
547. own 548. oxford 549. pace 550. paese 551. page 552. pain 553. paperback
554. papyrus 555. parcheggio 556. partner 557. partono 558. pass 559. path 560. pay
561. peak 562. pelvis 563. pensare 564. people 565. periferiche 566. persistent 567. persistenti
568. pianificano 569. pigment 570. piste 571. pneumococcus 572. poet 573. police 574. popcorn
575. portate 576. portfolio 577. possiamo 578. potpourri 579. potsdam 580. potute 581. pound
582. precisati 583. predisposto 584. prememorizzato 585. preparate 586. presentate 587. presupposto 588. primaria
589. privati 590. privilegiare 591. prodotta 592. programmato 593. projector 594. pronunciato 595. prospectus
596. proteggano 597. provocare 598. pulizie 599. pulse 600. quadratura 601. quarto 602. query
603. question 604. quick 605. Quick 606. quiet 607. quite 608. quota 609. raccogliere
610. race 611. rage 612. raggiunte 613. rapporti 614. rate 615. read 616. realistic
617. realizzata 618. recommend 619. referenti 620. reflex 621. reimpostato 622. rembrandt 623. repertorio
624. restituzioni 625. revue 626. rhesus 627. riaccende 628. riassuntive 629. ricaricato 630. richeste

631. ricollocato 632. riconfigurati 633. ricordate 634. ricostruiscono 635. ridistribuisce 636. rientra 637. riferire
638. rightly 639. riguardo 640. rilocare 641. rimossa 642. rinfrescato 643. rinvenute 644. ripartire
645. riportate 646. ripristina 647. rischiare 648. risiedono 649. rispettata 650. ristabiliti 651. ritornata
652. riutilizzata 653. rosse 654. said 655. saint 656. salts 657. salva 658. samaritan
659. samovar 660. sandwich 661. sbandierati 662. scambiata 663. scarico 664. schema 665. scherzo
666. schizophrenia 667. science 668. scientifica 669. scorretti 670. sea 671. secondaria 672. secret
673. segnalazioni 674. sei 675. selected 676. semplicissima 677. sent 678. sequenza 679. serum
680. seste 681. sfumature 682. she 683. sheriffs 684. ship 685. showman 686. shown
687. shuttle 688. sightseeing 689. simboliche 690. simpson 691. sincronizzate 692. sir 693. sistema
694. sleep 695. smistato 696. snob 697. so 698. society 699. sodium 700. software
701. sollecitano 702. some 703. soppresso 704. sostituire 705. sottolineano 706. sottolineato 707. sovrappongono
708. specie 709. spento 710. spreco 711. squaw 712. standardizzano 713. stanza 714. stesso
715. stewards 716. stockholm 717. stopwatch 718. strutturato 719. strychnine 720. studio 721. stuttgart
722. such 723. suddette 724. suoi 725. supportano 726. suspicion 727. svolgimento 728. sweatshirt
729. swiftly 730. Swiftly 731. symposium 732. tableau 733. tagliare 734. talus 735. task
736. tea 737. teamwork 738. technology 739. tecnico 740. tenosynovitis 741. tentano 742. terrible
743. terze 744. text 745. than 746. that 747. the 748. The 749. theatre
750. their 751. them 752. then 753. theory 754. there 755. these 756. they
757. thigh 758. this 759. those 760. threat 761. time 762. titolari 763. to
764. tokyo 765. tomahawk 766. tonic 767. too 768. tracciare 769. tragedy 770. tralasciata
771. transfer 772. transmission 773. trapezium 774. trasferimento 775. traslitterata 776. trasportati 777. trekking
778. tributari 779. triplet 780. trouble 781. trovava 782. turf 783. turquoise 784. two
785. ugly 786. ulcer 787. umana 788. unable 789. understood 790. union 791. uniti
792. unpleasant 793. update 794. upgrade 795. urban 796. us 797. usufruiscono 798. vacuum
799. vai 800. value 801. vantaggi 802. varen 803. Varen 804. vast 805. vault
806. vedete 807. veerpont 808. Veerpont 809. vegetable 810. vera 811. verso 812. vertical
813. very 814. vies 815. Vies 816. violence 817. virgin 818. visionare 819. vitae
820. voeten 821. Voeten 822. vogliono 823. voltmeter 824. vuist 825. Vuist 826. waarden
827. Waarden 828. waiter 829. waiting 830. waiver 831. wake 832. wales 833. walrus
834. warm 835. was 836. we 837. weinig 838. Weinig 839. were 840. what
841. when 842. which 843. who 844. widespread 845. wiek 846. Wiek 847. will
848. wish 849. with 850. woman 851. wonderland 852. working 853. workload 854. workshop
855. world 856. worst 857. Worst 858. would 859. wurgen 860. Wurgen 861. wyoming
862. xylophone 863. years 864. yes 865. yoga 866. you 867. young 868. your
869. youth 870. yucca 871. zaadje 872. Zaadje 873. zender 874. Zender 875. zero
876. zien 877. Zien 878. zigzag 879. zonder 880. Zonder 881. zoo 882. zuurtjes
883. Zuurtjes 884. zwei

Table A.1: Frequency count of basic unit in Assamese and English Words.

No of basic unit	No of Assamese word	No of English word
1	0	2
2	17	23
3	49	61
4	58	115
5	28	116
6	19	133
7	7	105
8	4	96
9	0	90
10	0	56
11	0	41
12	0	20
13	0	15
14	0	9
15	0	2

Table A.2: Number of samples corresponding to each of the basic units from Assamese character dataset.

Class ID	Class	No of samples	Class ID	Class	No of samples	Class ID	Class	No of samples
1	১	322	54	হ	279	107	কী	400
2	২	343	55	ক্ষ	321	108	কু	400
3	৩	322	56	য়	221	109	কূ	400
4	৪	371	57	ড়	230	110	কৃ	385
5	৫	342	58	ঢ়	289	111	কে	398
6	৬	287	59	ৎ	225	112	কৈ	400
7	৭	226	60	◌ং	237	113	কো	385
8	৮	328	61	◌ঃ	272	114	কৌ	400
9	৯	312	62	◌ঁ	273	115	র্ক	335
10	০	247	63	কা	293	116	খী	385
11	অ	254	64	খি	129	117	খু	400
12	আ	249	65	গী	274	118	খূ	328
13	ই	221	66	ঘু	264	119	খৃ	334
14	ঈ	212	67	ঙূ	211	120	খে	385
15	উ	244	68	চৃ	260	121	খৈ	355
16	ঊ	252	69	ছূ	170	122	খো	400
17	ঋ	225	70	জৈ	219	123	খৌ	385
18	এ	334	71	ঝো	231	124	গা	385
19	ঐ	314	72	ঞৌ	307	125	গি	400
20	ও	345	73	ট্য	196	126	গু	387
21	ঔ	309	74	ঠ্য	204	127	গূ	398
22	ক	182	75	ডা	269	128	গৃ	383
23	খ	239	76	টী	203	129	গে	383
24	গ	221	77	ণী	262	130	গৈ	400
25	ঘ	139	78	তু	282	131	গো	400
26	ঙ	283	79	থূ	235	132	গৌ	394
27	চ	158	80	দৃ	196	133	র্গ	400
28	ছ	238	81	ধে	137	134	ঘা	400
29	জ	161	82	র্ন	179	135	ঘি	396
30	ঝ	214	83	পো	212	136	ঘী	400
31	ঞ	251	84	ফৌ	192	137	ঘূ	400
32	ট	276	85	ব্য	237	138	ঘৃ	326
33	ঠ	268	86	র্ভ	172	139	ঘে	399
34	ড	269	87	মা	202	140	ঘৈ	215
35	ঢ	341	88	যি	140	141	ঘো	137
36	ণ	290	89	ৰি	282	142	ঘৌ	394
37	ত	274	90	লু	275	143	র্ঘ	400
38	থ	258	91	ৱ্	137	144	মি	395
39	দ	294	92	শৃ	232	145	মী	400
40	ধ	234	93	ষে	221	146	মু	397
41	ন	255	94	সৈ	293	147	মূ	399
42	প	339	95	হো	243	148	মৃ	400
43	ফ	285	96	ক্ষৌ	186	149	মে	399
44	ব	247	97	ড্য	312	150	মৈ	400
45	ভ	295	98	ঢ়্	214	151	মো	400
46	ম	288	99	য়া	235	152	মৌ	400
47	য	240	100	শু	220	153	ৰা	206
48	ৰ	290	101	গু	223	154	ৰী	400
49	ল	240	102	হু	282	155	ৰো	109
50	ৱ	214	103	ৰু	263	156	র্থ	400
51	শ	210	104	ৰূ	400	157	ক্য	400
52	ষ	273	105	কূ	385			
53	স	285	106	কি	385			

A. Database Analysis

Table A.3: Number of samples corresponding to each of the basic units from Assamese word dataset.

Class ID	Class	No of samples	Class ID	Class	No of samples	Class ID	Class	No of samples	Class ID	Class	No of samples
1	অ	1038	45	য়	1029	89	ণ্ড	206	133	ল্ল	65
2	আ	774	46	ড়	64	90	ত্ত্ব	59	134	ল্ল	199
3	ই	257	47	ঢ়	202	91	থ	65	135	ণ্ড	253
4	ঈ	387	48	ৎ	592	92	ত্র	64	136	শ্চ	64
5	উ	1220	49	ং	1058	93	ব্র	212	137	শ্ল	108
6	ঊ	200	50	ঃ	63	94	ত্ম	65	138	শ্ব	384
7	ঋ	66	51	ঁ	780	95	দ্দ	65	139	শ্ম	74
8	এ	1087	52	া	9714	96	দ্ব	450	140	শ্ল	73
9	ঐ	202	53	ি	5196	97	ভ্ভ	202	141	ষ্ক	64
10	ও	1139	54	ী	2214	98	দ্ম	201	142	ষ্ক্ল	67
11	ঔ	205	55	ু	2128	99	ণ্ট	197	143	ষ্ট	718
12	ক	2024	56	ূ	744	100	ন্ত	459	144	ষ্ঠ	578
13	খ	1150	57	ৃ	1038	101	ত্ত্ব	57	145	ষ্প	71
14	গ	1466	58	ে	2572	102	ন্ত্র	581	146	ষ্ফ	62
15	ঘ	773	59	ৈ	64	103	হ্ন	206	147	ষ্ম	71
16	ঙ	579	60	ো	66	104	ন্দ	382	148	ষ্ক	117
17	চ	1095	61	ৌ	2281	105	ন্ধ্র	64	149	স্ত	149
18	ছ	805	62	্	2068	106	ন্ম	202	150	স্ত্ত	59
19	জ	2197	63	ক্ক	66	107	ন্ব	850	151	স্ত্র	396
20	ঝ	65	64	ক্ট	65	108	ন্ম	191	152	হ্ন	796
21	ট	381	65	ক্ত	67	109	ন্ল	65	153	ম্ন	65
22	ঠ	582	66	ক্ত্র	64	110	ন্ষ	65	154	স্প	73
23	ড	180	67	ক্ষ্ণ	389	111	প্ট	63	155	স্ফ	159
24	ঢ	67	68	ক্ল	72	112	প্ত	65	156	স্ব	206
25	ণ	1001	69	গু	59	113	প্ন	64	157	স্ম	66
26	ত	2480	70	গ্ন	390	114	প্প	375	158	হ্ম	391
27	থ	397	71	থ	206	115	প্ল	72	159	হ্ম	64
28	দ	1362	72	গ্ম	65	116	ব্দ	73	160	১৩	63
29	ধ	823	73	গ্ল	74	117	ব্ধ	71	161	ষ্ম	783
30	ন	5206	74	ঙ্ক	81	118	ম্ব	73	162	ক্ষ্ণ	196
31	প	1990	75	ঙ্খ	392	119	ম্ভ্র	770	163	ক্ষ্ণ	197
32	ফ	790	76	ঙ্গ	193	120	ম্ম	71	164	১	568
33	ব	4127	77	ঙ্ঘ	197	121	ম্প	204	165	৮	1863
34	ভ	708	78	ঙ্ক্ষ	65	122	ম্ফ	72	166	৭	2185
35	ম	2719	79	চ্চ	65	123	ম্ব	71	167	৬	440
36	য	256	80	চ্ছ	66	124	ম্ভ	206	168	৫	52
37	ৰ	1711	81	জ্জ	181	125	ম্ব	397	169	৩	2262
38	ল	3065	82	জ্ঝ	206	126	ম্ম	756	170	২	116
39	ৱ	844	83	ঞ্ছ	66	127	ম্র	57	171	ঘ্ন	57
40	শ	2117	84	ঞ্ঝ	64	128	ল্ক	64	172	ম্ন	56
41	ষ	506	85	ঞ্ছ	388	129	ল্ট	64	173	স্খ	57
42	স	2831	86	ট্ট	64	130	ল্ড	64			
43	হ	1546	87	ড্ড	64	131	ল্প	64			
44	ক্ষ	978	88	ণ্ঠ	65	132	ল্ব	73			

Table A.4: Number of samples corresponding to each of the basic units from English character dataset.

Class ID	Class	No of samples	Class ID	Class	No of samples	Class ID	Class	No of samples
1	*1*	1395	22	*L*	1083	43	*g*	1474
2	*2*	1575	23	*M*	930	44	*h*	1988
3	*3*	1514	24	*N*	1049	45	*i*	973
4	*4*	1542	25	*O*	1610	46	*j*	871
5	*5*	1458	26	*P*	1000	47	*k*	1211
6	*5*	1449	27	*Q*	793	48	*l*	1178
7	*7*	1463	28	*R*	1194	49	*m*	1436
8	*8*	1481	29	*S*	1293	50	*n*	2687
9	*9*	1477	30	*T*	1222	51	*o*	3383
10	*10*	1586	31	*U*	1020	52	*p*	1581
11	*A*	1177	32	*V*	730	53	*q*	1063
12	*B*	931	33	*W*	853	54	*r*	2716
13	*C*	1016	34	*X*	702	55	*s*	2615
14	*D*	878	35	*Y*	827	56	*t*	2948
15	*E*	1142	36	*Z*	821	57	*u*	1615
16	*F*	671	37	*a*	3244	58	*v*	753
17	*G*	825	38	*b*	1141	59	*w*	1224
18	*H*	824	39	*c*	1340	60	*x*	761
19	*I*	720	40	*d*	1862	61	*y*	1258
20	*J*	694	41	*e*	4825	62	*z*	957
21	*K*	881	42	*f*	1299			

Table A.5: Number of samples corresponding to each of the basic units from English word dataset.

Class ID	Class	No of samples	Class ID	Class	No of samples	Class ID	Class	No of samples
1	*B*	34	16	*W*	24	31	*n*	6135
2	*C*	22	17	*Z*	25	32	*o*	6364
3	*D*	9	18	*a*	8356	33	*p*	2448
4	*F*	32	19	*b*	2182	34	*q*	532
5	*H*	25	20	*c*	3749	35	*r*	5992
6	*I*	18	21	*d*	3260	36	*s*	4586
7	*J*	9	22	*e*	9048	37	*t*	6973
8	*K*	25	23	*f*	1463	38	*u*	3790
9	*L*	33	24	*g*	1831	39	*v*	849
10	*M*	24	25	*h*	2445	40	*w*	1532
11	*O*	9	26	*i*	6271	41	*x*	753
12	*Q*	9	27	*j*	627	42	*y*	1461
13	*S*	9	28	*k*	1517	43	*z*	851
14	*T*	9	29	*l*	3200	44	’	18
15	*V*	25	30	*m*	2662			

Bibliography

[1] D. Keysers, T. Deselaers, H. A. Rowley, L.-L. Wang, and V. Carbune, "Multi-language online handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1180–1194, 2017.

[2] O. Samanta, U. Bhattacharya, and S. Parui, "Smoothing of HMM parameters for efficient recognition of online handwriting," *Pattern Recognition*, vol. 47, no. 11, pp. 3614 – 3629, 2014.

[3] R. Plamondon and S. Srihari, "Online and offline handwriting recognition: A comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 63–84, Jan 2000.

[4] Y. Chherawala, P. P. Roy, and M. Cheriet, "Feature set evaluation for offline handwriting recognition systems: application to the recurrent neural network model," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2825–2836, 2016.

[5] Y. Wen and L. He, "A classifier for Bangla handwritten numeral recognition," *Expert Systems with Applications*, pp. 948–953, 2012.

[6] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 855–868, 2009.

[7] H. T. Nguyen, C. T. Nguyen, P. T. Bao, and M. Nakagawa, "A database of unconstrained Vietnamese online handwriting and recognition experiments by recurrent neural networks," *Pattern Recognition*, vol. 78, pp. 291–306, 2018.

[8] S. Singh, A. Sharma, and I. Chhabra, "A dominant points-based feature extraction approach to recognize online handwritten strokes," *International Journal on Document Analysis and Recognition*, vol. 20, no. 1, pp. 37–58, 2017.

[9] I. Abdelaziz, S. Abdou, and H. Al-Barhamtoshy, "A large vocabulary system for Arabic online handwriting recognition," *Pattern Analysis and Applications*, vol. 19, no. 4, pp. 1129—1141, 2016.

[10] S. Jaeger, S. Manke, J. Reichert, and A. Waibel, "Online handwriting recognition: the NPen++ recognizer," *International Journal on Document Analysis and Recognition*, vol. 3, no. 3, pp. 169—180, 2001.

[11] A. Bharath and S. Madhvanath, "HMM-based lexicon-driven and lexicon-free word recognition for online handwritten Indic scripts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 670–682, 2012.

[12] S. Sundaram, "Lexicon free recognition strategies for online handwritten Tamil words," Ph.D. dissertation, Electrical Engineering Depratment, Indian Institute of Science, Bangalore, 2011.

[13] J. Hu, S. G. Lim, and M. K. Brown, "Writer independent online handwriting recognition using an HMM approach," *Pattern Recognition*, vol. 33, no. 1, pp. 133—147, 2000.

[14] X.-D. Zhou, D.-H. Wang, F. Tian, C.-L. Liu, and M. Nakagawa, "Handwritten Chinese/Japanese text recognition using semi-Markov conditional random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 10, pp. 2413—2426, 2013.

[15] M. Liwicki and H. Bunke, "HMM-based online recognition of handwritten whiteboard notes," in *Proc. of Int. Workshop on Frontiers in Handwriting Recognition*, 2006, pp. 595–599.

[16] S. Sundaram and A. G. Ramakrishnan, "Performance enhancement of online handwritten Tamil symbol recognition with reevaluation techniques," *Pattern Analysis and Applications*, vol. 17, no. 3, pp. 587–609, 2013.

[17] L. Prevost, L. Oudot, A. Moises, C. Michel-Sendis, and M. Milgram, "Hybrid generative/discriminative classifier for unconstrained character recognition," *Pattern Recognition Letters*, vol. 26, no. 12, pp. 1840—1848, 2005.

[18] L. Vuurpijl, L. Schomaker, and M. Erp, "Architectures for detecting and solving conflicts: two-stage classification and Support Vector Classifiers," *Document Analysis and Recognition*, vol. 5, no. 4, pp. 213–223, 2003.

[19] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[20] B. Logan and A. Salomon, "A music similarity function based on signal analysis," in *IEEE International Conference on Multimedia and Expo*, 2001, pp. 745—748.

[21] A. Alaei, P. Nagabhushan, and U. Pal, "A new two-stage scheme for the recognition of Persian handwritten characters," in *Proc. of Int. Conference on Frontiers in Handwriting Recognition*, 2010, pp. 130—135.

[22] J. Du, J.-F. Zhai, and J.-S. Hu, "Writer adaptation via deeply learned features for online Chinese handwriting recognition," *International Journal on Document Analysis and Recognition*, vol. 20, no. 1, pp. 69—78, 2017.

[23] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[24] S. Ghosh, P. K. Bora, S. Das, and B. B. Chaudhuri, "Development of an Assamese OCR using Bangla OCR," in *Proc. of the Workshop on Document Analysis and Recognition*, no. 6. ACM, 2012, pp. 68–73.

[25] B. Sarma, K. Mehrotra, R. Krishna Naik, S. Prasanna, S. Belhe, and C. Mahanta, "Handwritten Assamese numeral recognizer using HMM and SVM classifiers," in *Proc. of National Conference on Communications*, 2013, pp. 1–5.

[26] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet, "UNIPEN project of online data exchange and recognizer benchmarks," in *Proc. of Int. Conference on Pattern Recognition*, 1994, pp. 29—33.

[27] E. H. Ratzlaff, "Methods, reports and survey for the comparison of diverse isolated character recognition results on the UNIPEN database," in *Proc. of Int. Conference on Document Analysis and Recognition*, 2003, pp. 623—628.

[28] L. Schomaker, "The UNIPEN-ICROW-03 benchmark set," http://www.ai.rug.nl/~lambert/unipen/icdar-03-competition/_README, 2003, accessed: 04.08.17.

[29] Google, "Google books N-grams," http://norvig.com/google-books-common-words.txt, 2012, accessed: 04.08.17.

[30] X.-Y. Zhang, Y. Bengio, and C.-L. Liu, "Online and offline handwritten Chinese character recognition: A comprehensive study and new benchmark," *Pattern Recognition*, vol. 61, pp. 348–360, 2017.

[31] N. Tagougui, M. Kherallah, and A. M. Alimi, "Online Arabic handwriting recognition: a survey," *International Journal on Document Analysis and Recognition*, vol. 16, no. 3, pp. 209—226, 2013.

[32] C. Bahlmann and H. Burkhardt, "The writer independent online handwriting recognition system frog on hand and cluster generative statistical dynamic time warping," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 3, pp. 299–310, March 2004.

[33] S. Bhattacharya, D. S. Maitra, U. Bhattacharya, and S. K. Parui, "An end-to-end system for Bangla online handwriting recognition," in *Proc. of Int. Conference on Frontiers in Handwriting Recognitio*, 2016, pp. 373–378.

[34] K. Shashikiran, K. Prasad, R. Kunwar, and A. Ramakrishnan, "Comparison of HMM and SDTW for Tamil handwritten character recognition," in *Proc. of Int. Conference on Signal Processing and Communications*, 2010, pp. 1–4.

[35] U. Bhattacharya, B. Gupta, and S. Parui, "Direction code based features for recognition of online handwritten characters of Bangla," in *Int. Conference on Document Analysis and Recognition*, vol. 1, Sept 2007, pp. 58–62.

[36] B. Alijla and K. Kwaik, "OIAHCR: online isolated Arabic handwritten character recognition using neural network." *Int. Arab J. Inf. Technol.*, vol. 9, no. 4, pp. 343–351, 2012.

[37] I. Khodadad, M. Sid-Ahmed, and E. Abdel-Raheem, "Online Arabic/Persian character recognition using neural network classifier and DCT features," in *Proc. of Int. Midwest Symposium on Circuits and Systems*, 2011, pp. 1–4.

[38] U. Bhattacharya, A. Nigam, Y. Rawat, and S. Parui, "An analytic scheme for online handwritten Bangla cursive word recognition," in *Proc. of Int. Conference on Frontiers in Handwriting Recognition*, 2008, pp. 320–325.

[39] A. H. Toselli, M. Pastor, and E. Vidal, *Online Handwriting Recognition System for Tamil Handwritten Characters*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 370–377.

[40] N. Mezghani, A. Mitiche, and M. Cheriet, "On-line recognition of handwritten Arabic characters using a kohonen neural network," in *Int. Workshop on Frontiers in Handwriting Recognition*. IEEE, 2002, pp. 490—495.

[41] S. Kubatur, M. Sid-Ahmed, and M. Ahmadi, "A neural network approach to online Devanagari handwritten character recognition," in *Proc. of Int. Conference on High Performance Computing and Simulation*. IEEE, 2012, pp. 209–214.

[42] K. Primekumar and S. Idiculla, "Online Malayalam handwritten character recognition using Wavelet Transform and SFAM," in *Proc. of Int. Conference on Electronics Computer Technology*, 2011, pp. 49—53.

[43] M. Kherallah, L. Haddad, A. M. Alimi, and A. Mitiche, "On-line handwritten digit recognition based on trajectory and velocity modeling," *Pattern Recognition Letters*, vol. 29, no. 5, pp. 580 – 594, 2008.

[44] A. Fischer and R. Plamondon, "Signature verification based on the kinematic theory of rapid human movements," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 2, pp. 169–180, 2017.

[45] H. Choudhury and S. M. Prasanna, "Handwriting recognition using sinusoidal model parameters," *Pattern Recognition Letters*, vol. 121, pp. 87 – 96, 2019.

[46] F. Biadsy, R. Saabni, and J. El-Sana, "Segmentation-free online Arabic handwriting recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 25, no. 07, pp. 1009–1033, 2011.

[47] A. Ramzi and A. Zahary, "Online Arabic handwritten character recognition using online-offline feature extraction and back-propagation neural network," in *Int. Conference on Advanced Technologies for Signal and Image Processing*. IEEE, 2014, pp. 350—355.

[48] S. A. Azeem and H. Ahmed, "Combining online and offline systems for Arabic handwriting recognition," in *Int. Conference on Pattern Recognition*. IEEE, 2012, pp. 3725—3728.

[49] M. Liwicki and H. Bunke, "Handwriting recognition of whiteboard notes—studying the influence of training set size and type," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 21, no. 01, pp. 83–98, 2007.

[50] K. Daifallah, N. Zarka, and H. Jamous, "Recognition-based segmentation algorithm for on-line Arabic handwriting," in *Proc. of Int. Conference on Document Analysis and Recognition*, 2009, pp. 886—890.

[51] U. Pal, A. Belaıd, and C. Choisy, "Touching numeral segmentation using water reservoir concept," *Pattern Recognition Letters*, vol. 24, no. 1-3, pp. 261–272, 2003.

[52] P. S. Mukherjee, B. Chakraborty, U. Bhattacharya, and S. K. Parui, "A hybrid model for end to end online handwriting recognition," in *Proc. of Int. Conference on Document Analysis and Recognition*, vol. 1. IEEE, 2017, pp. 658—663.

[53] W. Yang, L. Jin, Z. Xie, and Z. Feng, "Improved deep convolutional neural network for online handwritten Chinese character recognition using domain-specific knowledge," in *Proc. of Int. Conference on Document Analysis and Recognition*, 2015, pp. 551—555.

[54] J. Du, J.-S. Hu, B. Zhu, S. Wei, and L.-R. Dai, "A study of designing compact classifiers using deep neural networks for online handwritten Chinese character recognition," in *Proc. of Int. Conference on Pattern Recognition*, 2014, pp. 2950–2955.

[55] S. D. Chowdhury, U. Bhattacharya, and S. K. Parui, "Online handwriting recognition using Levenshtein distance metric," in *Proc. of Int. Conference on Document Analysis and Recognition*, 2013, pp. 79–83.

[56] M. Kherallah, F. Bouri, and A. Alimi, "Online Arabic handwriting recognition system based on visual encoding and genetic algorithm," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 1, pp. 153 – 170, 2009.

[57] C.-L. Liu, S. Jaeger, and M. Nakagawa, "Online recognition of Chinese characters: The state-of-the-art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 198—213, 2004.

[58] Y. Liu and J. Tai, "A structural approach to online Chinese character recognition," in *Proc. of Int. Conference on Pattern Recognition*. IEEE, 1988, pp. 808—810.

[59] T. Wakahara, H. Murase, and K. Odaka, "On-line handwriting recognition," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1181–1194, 1992.

[60] Y.-L. Hsu, C.-L. Chu, Y.-J. Tsai, and J.-S. Wang, "An inertial pen with dynamic time warping recognizer for handwriting and gesture recognition," *Sensors Journal, IEEE*, vol. 15, no. 1, pp. 154—163, 2015.

[61] M. Gargouri, S. M. Touj, and N. E. B. Amara, "Towards unsupervised learning and graphical representation for on-line handwriting script," in *Proc. of Int. Multi-Conference on Systems, Signals & Devices*. IEEE, 2015, pp. 1–6.

[62] G. Kour and R. Saabne, "Fast classification of handwritten on-line Arabic characters," in *Proc. of Int. Conference of Soft Computing and Pattern Recognition*. IEEE, 2014, pp. 312–318.

[63] R. Niels and L. Vuurpijl, "Dynamic time warping applied to Tamil character recognition," in *Proc. of Int. Conference on Document Analysis and Recognition*, vol. 2, 2005, pp. 730—734.

[64] V. Vuori, J. Laaksonen, and J. Kangas, "Influence of erroneous learning samples on adaptation in on-line handwriting recognition," *Pattern Recognition*, vol. 35, no. 4, pp. 915—925, 2002.

[65] X. Li and D.-Y. Yeung, "On-line handwritten alphanumeric character recognition using dominant points in strokes," *Pattern recognition*, vol. 30, no. 1, pp. 31–44, 1997.

[66] J. Sternby, J. Morwing, J. Andersson, and C. Friberg, "On-line Arabic handwriting recognition with templates," *Pattern Recognition*, vol. 42, no. 12, pp. 3278 – 3286, 2009.

[67] R. Kunwar, S. K., and A. G. Ramakrishnan, "Online handwritten Kannada word recognizer with unrestricted vocabulary," in *Proc. of Int. Conference on Frontiers in Handwriting Recognition*, 2010, pp. 611—616.

[68] N. Bhattacharya, U. Pal, and P. P. Roy, "Stroke-order normalization for online Bangla handwriting recognition," in *Proc. of Int. Conference on Document Analysis and Recognition*, vol. 1. IEEE, 2017, pp. 206—211.

[69] G. Fink, S. Vajda, U. Bhattacharya, S. Parui, and B. Chaudhuri, "Online Bangla word recognition using sub-stroke level features and hidden Markov models," in *Proc. of Int. Conference on Frontiers in Handwriting Recognition,*, 2010, pp. 393—398.

[70] H. Boubaker, A. Chaabouni, M. Kherallah, A. M. Alimi, and H. E. Abed, "Fuzzy segmentation and graphemes modeling for online Arabic handwriting recognition," in *Proc. of Int. Conference on Frontiers in Handwriting Recognition*, 2010, pp. 695—700.

[71] G. Al-Habian and K. Assaleh, "Online Arabic handwriting recognition using continuous Gaussian mixture HMMS," in *Proc. of Int. Conference on Intelligent and Advanced Systems*, 2007, pp. 1183—1186.

[72] A. Biem., "Minimum classification error training for online handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 7, pp. 1041–1051, 2006.

[73] S. D. Connell and A. K. Jain, "Writer adaptation for online handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 329–346, 2002.

[74] J. Hu, M. Brown, and W. Turin, "HMM based online handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 1039—1045, 1996.

[75] O. Samanta, A. Roy, S. K. Parui, and U. Bhattacharya, "An HMM framework based on spherical-linear features for online cursive handwriting recognition," *Information Sciences*, vol. 441, pp. 133–151, 2018.

[76] M. Nakai, N. Akira, H. Shimodaira, and S. Sagayama, "Substroke approach to HMM-based on-line Kanji handwriting recognition," in *Proc. of Int. Conference on Document Analysis and Recognition*, 2001, pp. 491–495.

[77] M. Nakai, H. Shimodaira, and S. Sagayama, "Generation of hierarchical dictionary for stroke-order free Kanji handwriting recognition based on substroke HMM," in *Proc. Int. Conference on Document Analysis and Recognition*. IEEE, 2003, pp. 514—518.

[78] K. Verma and R. K. Sharma, "Comparison of HMM-and SVM-based stroke classifiers for Gurmukhi script," *Neural Computing and Applications*, vol. 28, no. 1, pp. 51–63, 2017.

[79] A. Bharath and S. Madhvanath, "A framework based on semi-supervised clustering for discovering unique writing styles," in *Proc. of Int. Conference on Document Analysis and Recognition*, 2009, pp. 891—895.

[80] M. Zhu, F. Shen, and Z. Wu, "Online Chinese characters recognition based on force information by HMM," in *International Conference on Human-Computer Interaction*. Springer, 2007, pp. 522–528.

[81] H. Choudhury, S. Mandal, S. Devnath, Prasanna, SRM., and S. Sundaram, "Comparison of Assamese character recognizer using stroke level and character level engines," in *Proc. of National Conference on Communications*, 2015, pp. 1–6.

[82] A. Kumar and S. Bhattacharya, "Online Devanagari isolated character recognition for the iphone using hidden Markov models," in *Proc. of Students' Technology Symposium*. IEEE, 2010, pp. 300–304.

[83] A. Senior and K. Nathan, "Writer adaptation of a HMM handwriting recognition system," in *Proc. of Int. Conference on Acoustics, Speech, and Signal Processing*, vol. 2. IEEE, 1997, pp. 1447—1450.

[84] M. Parizeau, A. Lemieux, and C. Gagne, "Character recognition experiments using Unipen data," in *Proc. of Int. Conference on Document Analysis and Recognition*, 2001, pp. 481—485.

[85] S. D. Chowdhury, U. Bhattacharya, and S. K. Parui, "Online handwriting recognition using Levenshtein distance metric," in *Proc. of Int. Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 79—83.

[86] S. Mohiuddin, U. Bhattacharya, and S. K. Parui, "Unconstrained Bangla online handwriting recognition based on MLP and SVM," in *Proc. of Multilingual OCR and Analytics for Noisy Unstructured Text Data*. ACM, 2011, p. 16.

[87] E. Caillault, C. Viard-Gaudin, and A. Ahmad, "MS-TDNN with global discriminant trainings," in *Proc. of Int. Conference on Document Analysis and Recognition*, vol. 2, 2005, pp. 856–860.

[88] S. Jaeger, C.-L. Liu, and M. Nakagawa, "The state of the art in Japanese online handwriting recognition compared to techniques in western handwriting recognition," *International Journal on Document Analysis and Recognition*, vol. 6, no. 2, pp. 75–88, 2003.

[89] M. Schenkel, I. Guyon, and D. Henderson, "Online cursive script recognition using time delay neural networks and hidden Markov models," *Machine Vision and Applications*, vol. 8, no. 4, pp. 215–223, 1995.

[90] S. Manke and U. Bodenhausen, "A connectionist recognizer for on-line cursive handwriting recognition," in *Proc. of Int. Conference on Acoustics, Speech, and Signal*, vol. 2. IEEE, 1994, pp. II—633.

[91] M. Liwicki, A. Graves, H. Bunke, and J. Schmidhuber, "A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks," in *Proc. of Int. Conference on Document Analysis and Recognition*, 2007.

[92] Z. Xie, Z. Sun, L. Jin, H. Ni, and T. Lyons, "Learning spatial-semantic context with fully convolutional recurrent network for online handwritten Chinese text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 8, pp. 1903—1917, 2018.

[93] B. Chakraborty, P. S. Mukherjee, and U. Bhattacharya, "Bangla online handwriting recognition using recurrent neural network architecture," in *Proc. of Indian Conference on Computer Vision, Graphics and Image Processing*. ACM, 2016, p. 63.

[94] C. J. Burges, "A tutorial on Support Vector Machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[95] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. John Wiley & Sons, 2012.

[96] N. Bhattacharya and U. Pal, "Stroke segmentation and recognition from Bangla online handwritten text," in *Proc. of Int. Conference on Frontiers in Handwriting Recognition*, 2012, pp. 740–745.

[97] B. Zhu, X.-D. Zhou, C.-L. Liu, and M. Nakagawa, "A robust model for on-line handwritten Japanese text recognition," *International Journal on Document Analysis and Recognition*, vol. 13, no. 2, pp. 121—131, 2010.

[98] S. Izadi and C. Suen, "Integration of contextual information in online handwriting representation," in *Image Analysis and Processing*, ser. Lecture Notes in Computer Science, P. Foggia, C. Sansone, and M. Vento, Eds. Springer Berlin Heidelberg, 2009, vol. 5716, pp. 132—142.

[99] H. Swethalakshmi, A. Jayaraman, V. S. Chakravarthy, C. C. Sekhar *et al.*, "Online Handwritten Character Recognition of Devanagari and Telugu Characters using Support Vector Machines," in *Tenth Int. Workshop on Frontiers in Handwriting Recognition*, 2006.

[100] K. Kumara, R. Agrawal, and C. Bhattacharyya, "A large margin approach for writer independent online handwriting classification," *Pattern Recognition Letters*, vol. 29, no. 7, pp. 933 – 937, 2008.

[101] K. Sivaramakrishnan and C. Bhattacharyya, "Time series classification for online Tamil handwritten character recognition – a kernel based approach," in *Neural Information Processing*, ser. Lecture Notes in Computer Science, N. Pal, N. Kasabov, R. Mudi, S. Pal, and S. Parui, Eds. Springer Berlin Heidelberg, 2004, vol. 3316, pp. 800—805.

[102] C. Bahlmann, B. Haasdonk, and H. Burkhardt, "Online handwriting recognition with Support Vector Machines - a kernel approach," in *Proc. of Int. Workshop on Frontiers in Handwriting Recognition*, 2002, pp. 49—54.

[103] C. Zanchettin, B. Bezerra, and W. Azevedo, "A KNN-SVM hybrid model for cursive handwriting recognition," in *Proc. of Int. Joint Conference on Neural Networks*, 2012, pp. 1–8.

[104] T. Bhowmik, P. Ghanty, A. Roy, and S. Parui, "SVM-based hierarchical architectures for handwritten Bangla character recognition," *International Journal on Document Analysis and Recognition*, vol. 12, no. 2, pp. 97—108, 2009.

[105] J. Milgram, R. Sabourin, and M. Cheriet, "Two-stage classification system combining model-based and discriminative approaches," in *Proc. of Int. Conference on Pattern Recognition*, vol. 1, Aug 2004, pp. 152–155 Vol.1.

[106] A. Bellili, M. Gilloux, and P. Gallinari, "An MLP-SVM combination architecture for offline handwritten digit recognition," *International Journal on Document Analysis and Recognition*, vol. 5, no. 4, pp. 244—252, 2003.

[107] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern recognition*, vol. 38, no. 12, pp. 2270–2285, 2005.

[108] P. Woodland, C. J. Leggetter, J. Odell, V. Valtchev, and S. Young, "The 1994 HTK large vocabulary speech recognition system," 06 1995, pp. 73–76 vol.1.

[109] Y. Rubner, C. Tomasi, and L. J. Guibas, "The Earth Mover's Distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99—121, 2000.

[110] S. Mandal, H. Choudhury, S. M. Prasanna, and S. Sundaram, "Frequency count based two stage classification for online handwritten character recognition," in *Proc. of Int. Conference on Signal Processing and Communications*, 2016, pp. 1–5.

[111] S. Prum, M. Visani, A. Fischer, and J. M. Ogier, "A discriminative approach to online handwriting recognition using bi-character models," in *Proc. of Int. Conference on Document Analysis and Recognition*, 2013, pp. 364—368.

[112] A. F. R. Rahman and M. C. Fairhurst, "Selective partition algorithm for finding regions of maximum pairwise dissimilarity among statistical class models," *Pattern Recognition Letters*, vol. 18, no. 7, pp. 605–611, 1997.

[113] B. Xu, K. Huang, and C.-L. Liu, "Similar handwritten Chinese characters recognition by critical region selection based on average symmetric uncertainty," in *Proc. of Int. Conference on Frontiers in Handwriting Recognition*. IEEE, 2010, pp. 527—532.

[114] K. Leung and C. H. Leung, "Recognition of handwritten Chinese characters by critical region analysis," *Pattern Recognition*, vol. 43, no. 3, pp. 949—961, 2010.

[115] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[116] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for Support Vector Machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, May 2011.

[117] G. Pradhan and S. R. M. Prasanna, "Speaker verification by vowel and nonvowel like segmentation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 4, pp. 854–867, 2013.

[118] J. Schenk and G. Rigoll, "Novel hybrid NN/HMM modelling techniques for online handwriting recognition," in *Proc. of Int. Workshop on Frontiers in Handwriting Recognition*, 2006.

[119] X. Chen, X. Liu, and Y. Jia, "Discriminative structure selection method of Gaussian Mixture Models with its application to handwritten digit recognition," *Neurocomputing*, vol. 74, no. 6, pp. 954–961, 2011.

[120] V. Bharathi and M. K. Geetha, "Performance evaluation of GMM and SVM for recognition of hierarchical clustering character," in *Advanced Computing, Networking and Informatics-Volume 1*, 2014, pp. 161–169.

[121] A. Mezghani, F. Slimane, S. Kanoun, and V. Märgner, "Identification of Arabic/French handwritten/printed words using GMM-based system." in *Proceedings of CIFED*, 2014, pp. 371–374.

[122] M. Liwicki, A. Schlapbach, P. Loretan, and H. Bunke, "Automatic detection of gender and handedness from on-line handwriting," in *Proc. of the Graphonomics Society*, 2007, pp. 179—183.

[123] A. Sharma and S. Sundaram, "A novel online signature verification system based on GMM features in a DTW framework," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 3, pp. 705–718, 2017.

[124] G. J. Zapata-Zapata, J. D. Arias-Londoño, J. F. Vargas-Bonilla, and J. R. Orozco-Arroyave, "On-line signature verification using Gaussian Mixture Models and small-sample learning strategies," *Revista Facultad de Ingeniería Universidad de Antioquia*, no. 79, pp. 86–97, 2016.

[125] A. Schlapbach, M. Liwicki, and H. Bunke, "A writer identification system for on-line whiteboard data," *Pattern Recognition*, vol. 41, no. 7, pp. 2381–2397, 2008.

[126] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[127] D. S. Maitra, U. Bhattacharya, and S. K. Parui, "CNN based common approach to handwritten character recognition of multiple scripts," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2015, pp. 1021–1025.

[128] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian Mixture Models," *Digital Signal Processing*, vol. 10, no. 1, pp. 19–41, 2000.

[129] R. Haeb-Umbach, "Investigations on inter-speaker variability in the feature space," in *Proceedings of Int. Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 397–400.

[130] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[131] A. Vedaldi and K. Lenc, "MatConvNet: Convolutional neural networks for MATLAB," in *Proc. of International Conference on Multimedia*. ACM, 2015, pp. 689–692.

[132] N. Tagougui, H. Boubaker, M. Kherallah, and A. Alimi, "A hybrid MLPNN/HMM recognition system for online Arabic handwritten script," in *World Congress on Computer and Information Technology*, 2013, pp. 1–6.

[133] S. Marukatat, T. Artieres, R. Gallinari, and B. Dorizzi, "Sentence recognition through hybrid neuro-Markovian modeling," in *Proc. of Int. Conference on Document Analysis and Recognition*, 2001, pp. 731—735.

[134] S. Garcia-Salicetti, B. Doizzi, P. Gallinari, A. Mellouk, and D. Fanchon, "A hidden Markov model extension of a neural predictive system for on-line character recognition," in *Proc. of Int. Conference on Document Analysis and Recognition*, vol. 1, 1995, pp. 50–53.

[135] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *Proc. of Int. Conference on Acoustics, Speech and Signal Processing*. IEEE, 2014, pp. 215—219.

[136] D. Povey, A. Ghoshal, G. Boulianne, N. Goel, M. Hannemann, Y. Qian, P. Schwarz, and G. Stemmer, "The kaldi speech recognition toolkit," in *In IEEE 2011 workshop*, 2011.

[137] 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*, vol. 31, no. 3, pp. 444–457, 2009.

[138] S. Espana-Boquera, M. J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez, "Improving offline handwritten text recognition with hybrid HMM/ANN models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 767–779, 2011.

List of Publications

Journal Publications

1. Subhasis Mandal, S.R. Mahadeva Prasanna and S. Sundaram, "*An Improved Discriminative Region Selection Methodology for Online Handwriting Recognition*", **International Journal on Document Analysis and Recognition**, Springer, Nov 2018.

2. Subhasis Mandal, S.R. Mahadeva Prasanna and S. Sundaram, "*GMM Posterior Features for Improving Online Handwriting Recognition*" **Expert Systems with Applications**, Elsevier, vol. 97, pp. 421—433, 2018.

Conference Publications

1. Subhasis Mandal, S.R. Mahadeva Prasanna and S. Sundaram, "*Exploration of CNN Features for Online Handwriting Recognition*", Accepted at **International Conference on Document Analysis and Recognition (ICDAR)**, 2019.

2. Subhasis Mandal, H. Choudhury, S. R. M. Prasanna, S. Sundaram, "*Exploring Discriminative HMM States for Improved Recognition of Online Handwriting*", Proc. of **International Conference on Pattern Recognition (ICPR)**, August 2018, pp. 3753—3758.

3. Subhasis Mandal, H. Choudhury, S. R. M. Prasanna, S. Sundaram, "*DNN-HMM based Large Vocabulary Online Handwritten Assamese Word Recognition System*", Proc. of **International Conference on Frontiers in Handwriting Recognition (ICFHR)**, August 2018, pp. 321—326.

4. Subhasis Mandal, S. R. M. Prasanna, S. Sundaram, "*Discriminative Region based Two Stage System for Online Handwritten Character Recognition*", Proc. of **IEEE India Conference (INDICON)**, December 2017, pp. 1—5.