GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Analysis of Handwriting Recognition System

A Project Report of Capstone Project - 2

Submitted By

PRASHANT VERMA

(1613101504/ 16SCSE101247)

In partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

Under the Supervision of

Mr. AMIT KUMAR

Assistant Professor

MAY-2020

GALGOTIAS
UNIVERSITY

SCHOOL OF COMPUTING AND SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report "**ANALYSIS OF HANDWRITING RECOGNITION SYSTEM**" is the bonafide work of "**PRASHANT VERMA (1613101504)**" who carried out the project work under my supervision.

SIGNATURE OF HEAD
Dr. MUNISH SHABARWAL,
PhD (Management), PhD (CS)
Professor & Dean,
School of Computing Science & Engineering

SIGNATURE OF SUPERVISOR
Mr. AMIT KUMAR, M.Tech.
SUPERVISOR
Assistant Professor,
School of Computing Science & Engineering

Abstract

It was a preliminary endeavor to gain first hand understanding of scope, problems, hurdles and challenges to be encountered in the development of a Devanagari handwriting recognition engine. I have learnt about routines for handling and manipulation of digital images and generation of digital signatures for handwritten letters and words. I am encouraged with these results and these preliminary finding help me to develop engine for automating image manipulation, feature extraction and character and symbol matching device.

The thesis is divided into five chapters. Chapter 1 provides introduction to the field of handwriting recognition systems and various steps involved in it. Chapter 2 describes a review of the existing literature on state of current knowledge and practices in the development of handwriting recognition systems. Chapter 3 provides experimental details pertaining to preprocessing of Devanagari hand scripts. It describes various potential and limitations of different image processing steps in the recognition of handwritten Devanagari paragraphs. Chapter 4 provides a brief summary of this work and gives outlines for future work in this area. Chapter 5 includes the relevant references in the field of this work.

Key words: Handwriting, Devanagari, Recognition, Image Preprocessing, Histogram, Moving Window.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

A world without written characters is unimaginable to people today, but such was the case some six millennium ago. All communication used to be spoken. Writing began in many clusters of civilizations, earliest in Sumeria (4000 BCE) and in Indus Valley. Although pictographic and petroglyph representations of latter have still not been deciphered, former produced the Epic of Gilgamesh. Chinese script that originate in 2^{nd} millennium BC is also based on pictographic script. However the subsequent residents of Indian subcontinent also produced masterpieces of thinking and poetry in the forms of Vedas, Upanishads and Sangam literature, most of which remained spoken until 4^{th} century BC.

As humans began to quit nomadic life and developed settlements necessity of clarity in communication of rules and information alike required scripts of languages. It was to overcome the limitations posed by human memory that writing had developed. The ancient scripts of Kharoshthi, Brahmi and Dravidian scripts emerged. Most of the north Indian language scripts are descendants of Brahmi script.

Scripts are pictorial presentations that human brain can learn upon training. However with the rise of modern digital age we require machines to read and interpret written words at high speed which otherwise is unimaginable for the human brain to accomplish. Numerous methodologies and routines have emerged to read and interpret even the hand written words in different languages.

Over time individuals tend to develop a writing style that is unique to them. A possible reason for this may be the physical nature of the writing process involving an integrated use of the mind, muscles, and eye. Writing style also depends upon the mood of the writer. Though the writing of two people may appear similar but their style of writing will generally differ.

From the very beginning the most important input interface to computers has been the keyboard. With increasing diversity in the skills of computer users, need for a more natural interface is increasingly becoming more important. The most likely inputting candidates could be voice or handwriting. Voice and handwriting recognition capabilities built into a computer could simplify data entry, which otherwise required keyboard entries. Handwriting recognition seems to be more practical than speech recognition because of the fact that in crowded rooms or public places one might not wish to speak to his or her computer due to the confidentiality or personal nature of the data. Also that voice dictation would annoy others sharing the same space. The handwriting recognition systems are more amenable to miniaturization and portable applications. However the handwriting based systems would compromise the speed of data entry as compared to voice based systems.

Since the late 1960's, various methodologies have been introduced for handwriting recognition of roman scripts. But so far there is very limited availability recognition technology for Devanagari (Hindi) script. Google handwriting input is one such tool for inputting Devanagari (Hindi) characters, but it is quite slow and often imperfect. Various methods based on extended knowledge with corresponding heuristic strategies, lexical databases and learning methodologies are being actively investigated.

Section 1.2 gives an overview of handwriting recognition systems, followed by architecture of handwriting recognition in section 1.3. Section 1.4. Introduces Devanagari script and the challenges faced during the recognition of handwritten Devanagari characters. In section 1.5 potential methodology to solve the challenges mentioned in 1.4 are discussed.

1.2 HANDWRITING RECOGNITION SYSTEMS

Handwriting recognition system is a software that allows for handwritten content to be converted to ASCII character codes that a machine can understand.

The above definition may sound simple but its implementation is anything but simple. Humans have evolved to be very good pattern recognizers. Our ability to recognize objects given the minimalist of details is remarkable.

Incorporating this ability in a computer has been the aim of computer vision. Optical Character Recognition (OCR) technique has seen remarkable accuracy, but the same cannot be said for handwriting recognition mainly due to the variations in writing styles. Though there are algorithms that would give 94% accuracy on handwriting recognition, their accuracy seems to drop as the writing is varied. The next part presents the common forms of handwriting recognition software followed by a discussion of some of the inherent complexities of recognizing Devanagari script.

1.2.1 TYPES OF HANDWRITING RECOGNITION SYSTEM

Handwriting recognition systems are divided into two categories. This division is based on the temporal relationship between writing and recognition.

Two categories of handwriting recognition system:

1- Online handwriting recognition
2- Offline handwriting recognition

1.2.1.1 ONLINE HANDWRITING RECOGNITION

This category houses those systems that perform recognition in real time, i.e. recognition and writing are done simultaneously. These systems a present very small delay between writing and recognition and so are preferred in places where on time results are needed.

Implementation of online systems is relatively easy when compared with the implementation of offline systems. They have additional information such as the timing of strokes, the latency between strokes and the path traced by these strokes. Using these additional information recognition becomes relatively easy.

As Jim Hyung Kim and Bong-Kee Sin have described "*online handwriting signal is a spatiotemporal signal that is intended to describe a geometrical shape*". Points are represented in a two dimensional plane. When dimension of time is also incorporated, a spatiotemporal trajectory is arrived at (Figure 1).

The advantages presented by online systems come with increased cost. These systems require special writing devices that can record the strokes and send the data for computation. Tough with increasing technology the cost of these devices has considerably reduced, they still are comparatively expensive. The application of these systems is also limited. They cannot be used for processing tasks that require data to be filled before recognition can proceed, examples include processing forms in a bank, or reading the address on envelopes in a post office.

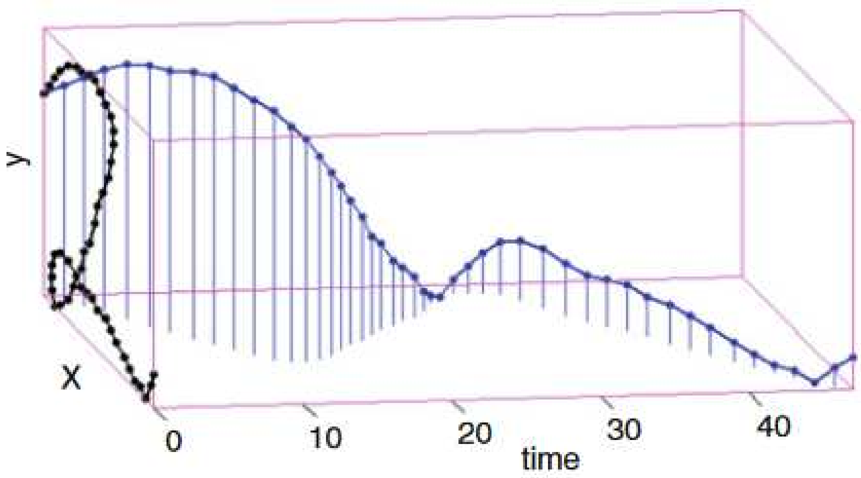


Figure 1: Spatiotemporal trajectory.

1.2.1.2 OFFLINE HANDWRITING RECOGNITION

These systems unlike the online system are not capable of performing simultaneous writing and recognition. Images need to be feed to the system that then analyzes the image and recognizes the handwritten characters.

Implementing such systems is relatively difficult, as dynamic data pertaining to pen movement, stroke direction, and stroke length is not present. To overcome this disability these systems perform rigorous image preprocessing followed by segmentation. Extensive feature extraction is also performed to improve the accuracy of results.

1.3 ARCHITECTURE OVERVIEW

The success of off-line handwriting recognition system depends upon the outcome of the modules that constitute the system.

The arrangement of modules with respect to the data shared among them constitutes the architecture of a system. In this chapter a general architecture is presented that incorporates the modules of existing handwriting recognition system.

1.3.1 GENERAL ARCHITECTURE

Off-line handwriting recognition systems perform exhaustive operations on the input image. The operations are divided into modules namely:

1- Preprocessing module
2- Line Segmentation module
3- Word Segmentation module
4- Feature Extraction
5- Classification

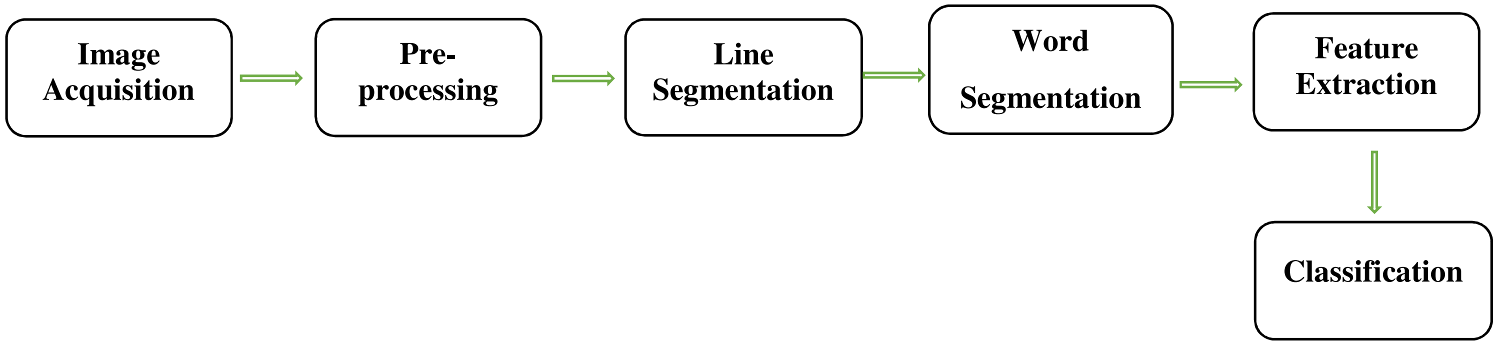


Figure 2: Architecture of recognition system.

1.3.2 PREPROCESSING

During preprocessing a series of operations are performed that enhance and make the image suitable for segmentation. Preprocessing step involve:

1- **Binarization:** Conversion from Grayscale to Binary. Otsu's method for noise removal can be used to calculate the threshold value.
   Filters like median filter, mean filter, min-max filter, and Gaussian filter can also be used to remove noise.
2- **Binary morphological operations:** Opening, closing, thinning, hole filling are used to enhance the visibility and structural information of characters.

3- **Slant Correction:** If document is scanned it may not be perfectly aligned, slant angle correction is performed to remove slant.
4- **Slope Correction:** Words written on a paper may not be in a straight line. This requires slope correction to be performed.
5- **Resizing:** Input document may be resized to improve processing speed. However reducing dimensions below a certain level may remove a few useful features.

1.3.3 LINE SEGMENTATION

The image is scanned horizontally, from left to right and top to bottom. For each row the number of black pixels are recorded. Using the number of black pixels and a threshold value lines can be segmented.

1.3.4 WORD SEGMENTATION

Similar to line segmentation, but instead to scanning pixels row wise they are scanned column wise. The number of black pixel in each column is recorded and using a threshold value the words can be segmented.

1.3.5 FEATURE EXTRACTION

Feature extraction is the heart of any pattern recognition application. Feature extraction techniques like Principle Component Analysis (PCA), Linear Discriminates Analysis (LDA), Independent Component Analysis (ICA), Chain Code (CC), zoning, Gradient based features, Histogram might be applied to extract the features of individual characters. These features are used to train the classifier.

1.4 REQUIREMENTS FOR RECOGNITION OF DEVANAGARI HANDWRITING

The Devanagari script consists of 48 alphabets. 12 of which are vowels and 36 are consonants. Each consonant symbol has a full form and a half form, i.e. without “a”. The consonant forms in combination with vowels are modified to from (36 x 12 =) 432 full consonant-vowel combinations (Figure 3 and Figure 4).

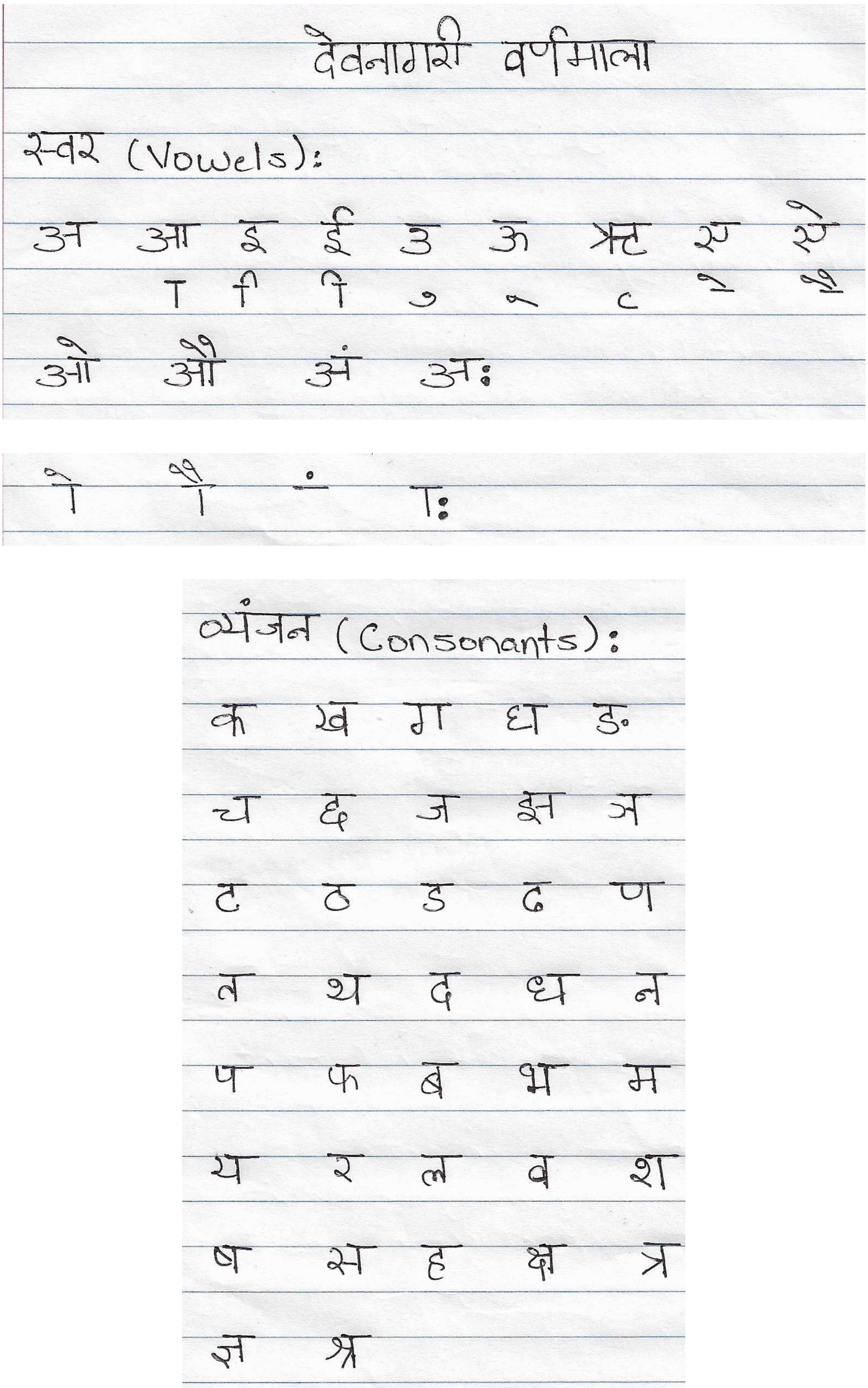


Figure 3: Characters in Devanagari script.

अ	आ	ई	ई	उ	ऊ	ए	ऐ	ओ	औ	अं	अः
क	का	कि	की	कु	कू	के	कै	को	कौ	कं	कः
ख	खा	खि	खी	खु	खू	खे	खै	खो	खौ	खं	खः
ग	गा	गि	गी	गु	गू	गे	गै	गो	गौ	गं	गः
घ	घा	घि	घी	घु	घू	घे	घै	घो	घौ	घं	घः
च	चा	चि	ची	चु	चू	चे	चै	चो	चौ	चं	चः
छ	छा	छि	छी	छु	छू	छे	छै	छो	छौ	छं	छः
ज	जा	जि	जी	जु	जू	जे	जै	जो	जौ	जं	जः
झ	झा	झि	झी	झु	झू	झे	झै	झो	झौ	झं	झः
ट	टा	टि	टी	टु	टू	टे	टै	टो	टौ	टं	टः
ठ	ठा	ठि	ठी	ठु	ठू	ठे	ठै	ठो	ठौ	ठं	ठः
ड	डा	डि	डी	डु	डू	डे	डै	डो	डौ	डं	डः
ढ	ढा	ढि	ढी	ढु	ढू	ढे	ढै	ढो	ढौ	ढं	ढः
ण	णा	णि	णी	णु	णू	णे	णै	णो	णौ	णं	णः
त	ता	ति	ती	तु	तू	ते	तै	तो	तौ	तं	तः
थ	था	थि	थी	थु	थू	थे	थै	थो	थौ	थं	थः
द	दा	दि	दी	दु	दू	दे	दै	दो	दौ	दं	दः
ध	धा	धि	धी	धु	धू	धे	धै	धो	धौ	धं	धः
न	ना	नि	नी	नु	नू	ने	नै	नो	नौ	नं	नः
प	पा	पि	पी	पु	पू	पे	पै	पो	पौ	पं	पः
फ	फा	फि	फी	फु	फू	फे	फै	फो	फौ	फं	फः
ब	बा	बि	बी	बु	बू	बे	बै	बो	बौ	बं	बः
भ	भा	भि	भी	भु	भू	भे	भै	भो	भौ	भं	भः
म	मा	मि	मी	मु	मू	मे	मै	मो	मौ	मं	मः
य	या	यि	यी	यु	यू	ये	यै	यो	यौ	यं	यः
र	रा	रि	री	रु	रू	रे	रै	रो	रौ	रं	रः
ल	ला	लि	ली	लु	लू	ले	लै	लो	लौ	लं	लः
व	वा	वि	वी	वु	वू	वे	वै	वो	वौ	वं	वः
श	शा	शि	शी	शु	शू	शे	शै	शो	शौ	शं	शः
ष	षा	षि	षी	षु	षू	षे	षै	षो	षौ	षं	षः
स	सा	सि	सी	सु	सू	से	सै	सो	सौ	सं	सः
ह	हा	हि	ही	हु	हू	हे	है	हो	हौ	हं	हः
क्ष	क्षा	क्षि	क्षी	क्षु	क्षू	क्षे	क्षै	क्षो	क्षौ	क्षं	क्षः
त्र	त्रा	त्रि	त्री	त्रु	त्रू	त्रे	त्रै	त्रो	त्रौ	त्रं	त्रः
ज्ञ	ज्ञा	ज्ञि	ज्ञी	ज्ञु	ज्ञू	ज्ञे	ज्ञै	ज्ञो	ज्ञौ	ज्ञं	ज्ञः
श्र	श्रा	श्रि	श्री	श्रु	श्रू	श्रे	श्रै	श्रो	श्रौ	श्रं	श्रः

Figure 4: Symbols for Consonant-Vowel combinations.

1.4.1 CHALLENGES IN DEVANAGARI HANDWRITING RECOGNITION

For Devanagari character recognition there might be multiple techniques but it is difficult to achieve 100% accuracy in recognition, particularly for the characters with higher degree of similarity (Figure 5). Higher accuracy can be achieved with proper data collection and preprocessing. Scanned documents may be more suitable than camera captured copies for character recognition.

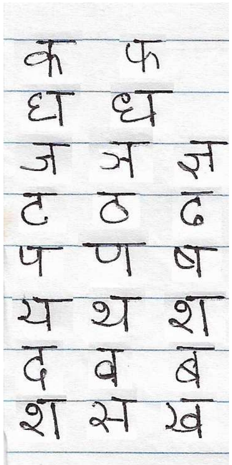


Figure 5: Symbols with high degree of similarity.

1.4.2 SCENE COMPLEXITY

Images acquired with camera may have additional complexities of including other object such as drawings, sketches, etc. that are difficult to distinguish from the characters in the text. Images that include non-textual contents make preprocessing difficult and thereby affect the character recognition process.

1.4.3 CONDITIONS OF UNEVEN LIGHTING

Many times image taken under non uniform lighting include shades and shadows. It makes the demarcation of character segments difficult. Scanned documents are invariably free from such variations. Camera light flash may help with additional lighting but create shadows in images.

1.4.4 DEGREE OF SLANTNESS IN CHARACTERS

While writing on blank paper words tend to become slanted. The degree of slantness varies with individual style. Furthermore the lines may tend to tilt. Slantness and tilting requires proper identification and correction for improving accuracy in recognition.

1.4.5 BLURRING AND DEGRADATION

Often the camera acquired images tend to be blurry due to hand stability during image acquisition. Blurring can lead to imperfect detection of margins and may require enhancing the sharpness and de-blurring during preprocessing.

1.4.6 SHIROREKHA

In Devanagari script a word comprised of multiple characters and modifiers connected through a common Shirorekha. Along with numerals, component characters, consonants, vowels, and vowel modifiers there are many similar shaped characters existing in devanagari script (Figure 6). All these make the handwritten character recognition, a difficult task.

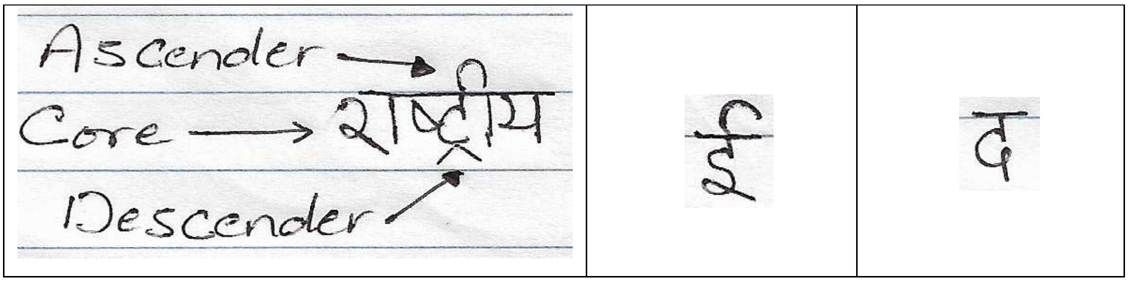


Figure 6: Ambiguity arising from modifiers.

1.4.7 COMPOUND CHARACTERS

In Devanagari script certain consonants are joined together giving rise to new characters or symbols (Figure 7). Due to this structural complexity, recognition of compound and joint characters is a challenging task, more so in the handwritings. Thus the recognition of Devanagari characters is highly complex due to its rich set of conjuncts. There are about 280 compound characters in the Devanagari script.

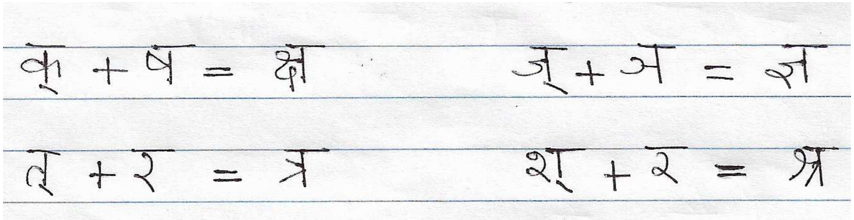


Figure 7: Compound characters.

1.4.8 SIZE, SHAPE AND STYLE

The major obstacles in recognizing handwritten Devanagari character have been the changeable writing conditions and different handwriting styles. There is wide variety of size, shape and style of each character used by individual writers.

1.5 POTENTIAL METHODOLOGY FOR RAPID RECOGNITION OF DEVANAGARI HANDWRITING

Given the complexity as mentioned above, any ab-initio method for recognizing of individual Devanagari characters and symbols would be extremely arduous and slow. However, a combination of heuristic, statistical and lexical database would be the most appropriate method.

1.5.1 HEURISTIC

These are simple rules that do not promise an optimal solution but the solution they provide can be used to improve performance. A simple heuristic to demarcate words can be to examine the shirorekha and stop once a continuous sequence of foreground colors has ended (Figure 8). Another heuristic that can be employed requires counting the loops present in a character, based on the number of loops a lot options can be eliminated.

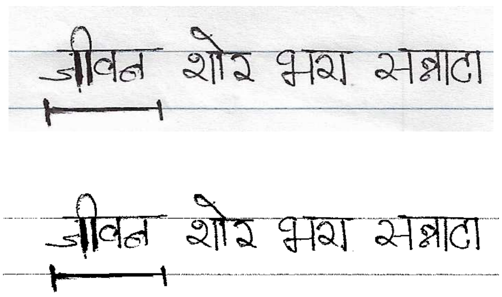


Figure 8: Using shirorekha for demarcation of words.

1.5.2 STATISTICAL

Statistical methods are of two type parametric and non-parametric. In parametric methods the complexity is independent of the size of input due to its fixed number of parameters. Parametric methods are thus applicable where training data is small, and quick learning is required. A few examples of parametric methods are Logistic Regression (LR), Linear Discriminant Analysis (LDA), Hidden Markov Model (HMM) etc.

However, non-parametric methods provide for flexibility in learning concepts but their complexity grows with input size. K Nearest Neighbor (KNN), Decision Trees (DT) are examples of non-parametric techniques.

1.5.2.1 THE NON-PARAMETRIC METHODS

K Nearest Neighbor is the most commonly used non-parametric method when a-priori knowledge of the characters is not available. kNN seeks to find the number of training samples closest to new example based on target function. Based on the result of the target function, the value of output class is inferred. The probability of an unknown sample “X” belonging to class “A” can be calculated as follows:

$$p(A|X) = \frac{\sum W_k \cdot 1_{(k_y = A)}}{\sum W_k}$$

$$W_k = \frac{1}{d(k, X)}$$

Where:

K is the set of nearest neighbors

k_y is the class of k and

d(k, X) the Euclidean distance of k from X, respectively.

The above equation is saying that the probability that “X” belongs to “A” is given by calculating the sum of inverse of distances from “X” to points k belonging to the class “A” and dividing it by the sum of distances from all the points of the nearest neighbors.

1.5.2.2 THE PARAMETRIC METHODS

As mentioned above parametric techniques use finite number of parameters. They assume that sample population or training data can be modeled by a probability distribution that has a fixed set of parameters. Once parameters of the model are learnt, characters can be classified based on rules like maximum likelihood or Bayes probability.

Hidden Markov Model (HMM) is one the most frequently used parametric statistical method used since 2000. In HMM model data that is assumed to be Markov process with hidden states, where in Markov process, probability of one state depends on previous state.

i.e. *X(t)* is a Markov Process if, for arbitrary times $t_1 < t_2 \ldots\ldots < t_k < t_{k+1}$

$$P(X(t_{k+1}) = x_{k+1} | X(t_k) = x_k, \ldots\ldots, X(t_1) = x_1)$$

$$= P(X(t_{k+1}) = x_{k+1} | X(t_k) = x_1)$$

It was first used in speech recognition during 1990s before researchers started using it in recognition of optical characters. It is believed that HMM provides better results even when availability of lexicons is very limited.

1.5.3 LEXICAL DATABASE

Syntactic analysis can be used to find similarities structural primitives using concept of grammar. Benefit of using grammar concepts in finding similarity comes from the fact that this area is well researched and techniques are well developed.

There are different types of grammar based on restriction rules, for example unrestricted grammar, context-free grammar, context-sensitive grammar and regular grammar. Explanation of these grammar and corresponding applied restrictions are out scope of this report.

With well defined grammar, a string is produced that then can be robustly classified to recognize character. A Tree structure can be used to model hierarchical relations among structural primitives. Trees can also be classified by analyzing grammar that defines the tree, thus classifying specific character.

CHAPTER 2

LITERATURE REVIEW

2.1 OVERVIEW

Recognition of handwritten text in different languages constitutes an important and active research area in the field of computer science. These researches aim to convert the hand written text from an image format to an electronic Unicode symbols. Although, the text recognition engine remains the main component of a document processing system, so much so that the success of any document processing system depends on the precise text recognition capability. Various approaches are being tried to deal with handwritten documents for a large vocabulary recognition task. Specifically, the most widely used methods are based on the hidden Markov's models (HMMs) [2,7,13,14,18,26,28,31,35,40,43,44,45], the recurrent neural networks (RNN) [1,6,12,19,20,21,38,41,50,51,62,65] and Support Vector Machine (SVM) [54,58,59,60,61].

Character recognition systems often rely on the internal representations that are produced using the sliding window approach. Features of characters are extracted from the vertical frames of the line art image and output is fed to a trainable classifier. Thus the task of character identification is transformed to a sequence to sequence transduction and encoding of the two-dimensional image using convolutional neural networks [6] or defining the relevant features [10,23] .

Although the character recognition has attracted attention of researchers for past few decades, substantial progress has been made with the development of deep learning methods during the last few years. Advent of deep learning methods facilitated developing systems that can handle the 2D aspect of the input image and the prediction of sequential aspect simultaneously. The multidimensional long short-term memory recurrent neural networks (MDLSTM-RNNs) in conjunction with the Connectionist Temporal Classification (CTC) [19] have been particularly successful in recognizing the characters in different languages [21,65,50]. More recently, the attention-based models, averting the paragraphs to lines segmentation problems have been applied to recognize handwritten texts [5].

2.2 TYPES OF CLASSIFIERS

2.2.1 HIDDEN MARKOV MODELS (HMM)

At the beginning of the past century, the Russian mathematician Andrej Andrejewitsch Markov applied the statistical model for the analysis of character sequences [35]. In Hidden Markov Model (HMM) it is assumed that the system under study is a Markov process with unobserved (hidden) states. A HMM can be considered the simplest dynamic Bayesian network.

In simpler Markov models, e.g. in Markov chain the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a hidden Markov model, the state is not directly visible, but output that depends on the state, is visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states. The state sequence through which the model passes are hidden even if the model parameters are known exactly HMM is especially useful in temporal pattern recognition such as speech, handwriting, gesture recognition, part-of-speech tagging, musical score following, and bioinformatics. HMM can be considered a generalization of a mixture model where the hidden variables (or latent variables), which control the mixture component to be selected for each observation, are related through a Markov process rather than independent of each other.

There are a number or reports Markov-model-based recognition of isolated characters [5, 7, 18, 22, 32, 37]. Actual recognition is either performed for isolated words or for connected words. The latter is the more complicated but more realistic use case since, e.g., full sentences can be treated without relying on prior-successful-segmentation. Consequently, actual document analysis based on text recognition becomes possible.

2.2.2 RECURRENT ARTIFICIAL NEURAL NETWORK (ANN)

Artificial neural network is widely accepted classifier for diverse patterns. Multilayer neural networks have been widely used in pattern recognition applications. The different network models are specified by:

Network topology: the number of neurons and how the neurons are interconnected.

Node characteristics: the type of non-linear transfer function used by the neuron for calculating the output value.

Training rules: specify how the weights are initially set and adjusted to improve performance of the network.

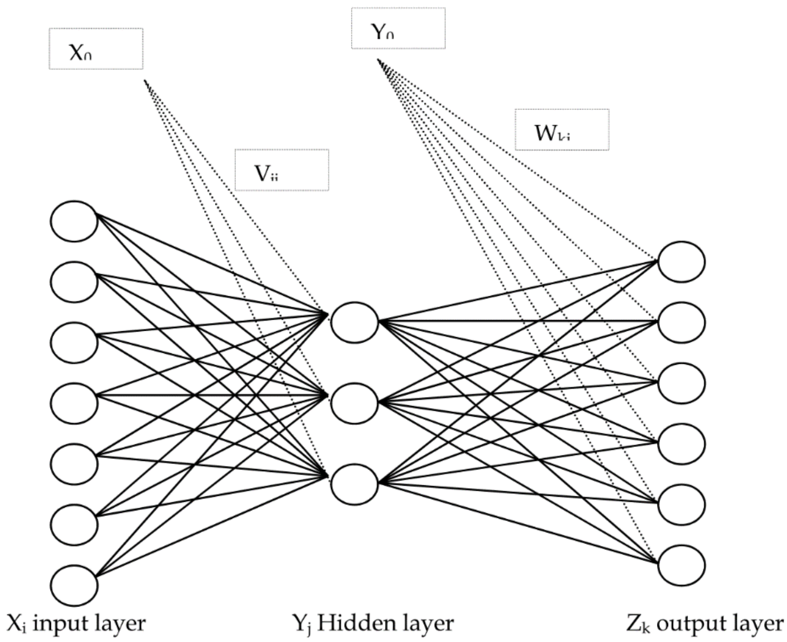


Figure 1: Neural Network.

The architecture of neural network depends on nature and complexity of applications. However multilayer neural network with proper choice of parameter is capable enough to classify almost any pattern. There are so many parameters that control the performance of neural network, like

1- Number of layers
2- Number of neurons in each layer
3- Transfer function used between two layers
4- Learning algorithm
5- Number of epochs.

2.2.3 SUPPORT VECTOR MACHINE (SVM)

Support vector machine is supervised learning tool, which is used for classification and regression. The basic SVM takes a set of input data and predicts, for each given input. Given a set of training examples, each marked as belonging to one of two categories, SVM training algorithm builds a model that assigns new examples into one category or the other. More formally, a support vector machine constructs a hyper plane or set of hyper planes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyper plane that has the largest distance to the nearest training data point of any class since in general the larger the margin the lower the generalization error of the classifier.

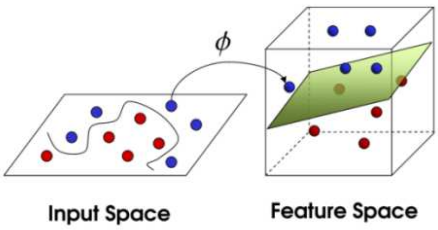


Figure 2: Feature transformation.

Whereas the original problem may be stated in a finite dimensional space, it often happens that the sets to discriminate are not linearly separable in that space. For this reason it was proposed that the original finite-dimensional space be mapped into a much higher-dimensional space, presumably making the separation easier in that space.

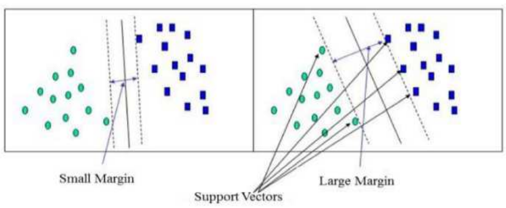


Figure 3: Maximum margin classification by SVM.

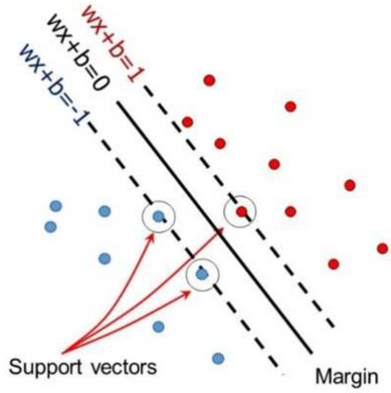


Figure 4: Linear separating hyper planes for the separable case.

2.3 HANDLING THE HANDWRITTEN IMAGE

2.3.1 PREPROCESSING

Scanned documents require preprocessing tasks like image binarization, word segmentation, and noise reduction. In skeleton direction-based features extraction the word skeleton image is virtually enlarged with some blank lines to move the baseline of the word into the middle of the image. The normalized word skeleton graph is subsequently used for the feature extraction process.

The feature extraction process starts by splitting the word image into a set of frames with fixed width in vertical direction. Each frame has an overlap of 50% with its neighbors. Each frame is split horizontally into 4-5 zones with equal height. The choice of 5 zones yielded the best recognition results [11]. The lengths of all lines within a zone of a frame are used as feature. These lengths are calculated in the four directions north, east, south, and west. A normalization of these values with respect to the height of the zone ensures the invariance of the height of the word. The wide variation in value ranges of each feature, normalization of the values are performed [55].

The text line segmentation in handwritten document images deal with the following challenges:

1- Each text line that appears in the document may have an arbitrary skew angle and converse skew angle along the text line

2- Text lines may have different skew directions

3- Accents may be cited either above or below the text line

4- Parts of neighboring text lines may be connected.

Using the binary image a connected component analysis is performed, contour representation of the image is extracted, and noise reduction filtering are performed synchronously. Subsequently language characteristic preprocessing and normalization steps are carried out prior to feature extraction. They are performed in order to reduce writer variability and involve additional tasks of, e.g., slant correction, writing lines normalization, horizontal scaling according to the number of characters, and line thickness normalization. The result of the preprocessing stage is a normalized gray scale representation of the script, which is then fed into the feature extraction stage. Figure 5 shows an example of the preprocessing and normalization steps.

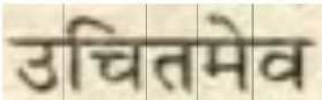
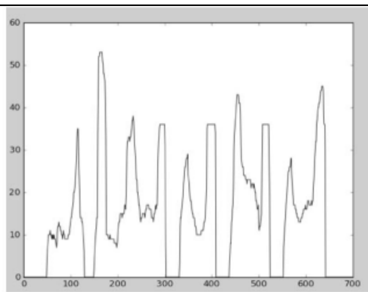
segmented letters	
pixel intensity calculation	

Figure 5: Preprocessing of Image.

2.3.2 WRITING LINES ESTIMATION

Segmentation of a document image into its basic entities namely text lines and words, is a critical stage towards recognition of text in a handwritten document. There arise many problems in the segmentation procedure. For a text line segmentation procedure these include the difference in the skew angle between lines on the page or even along the same text line, overlapping words and adjacent text lines touching each other. They make the segmentation procedure a challenging task.

A variety of text line detection methods for handwritten documents have been reported in the literature. These can be put into three basic categories. Methods lying in the first category make use of the Hough transform ([29, 30, 46]). In these methods, by starting from some points of the initial image, the lines that fit best to these points are extracted. The points considered in the Hough transform are usually either the gravity centers [29] or minima points [46] of the connected components. The second category of methods make use of projections ([3, 4, 8, 34]). In this methodology document image is divided into vertical strips and in these strips the horizontal projections are calculated. The resulting projections are combined in order to extract the final text lines. Histogram of the pixels' intensities at each scan line are calculated. The produced bins are smoothed and the corresponding valleys are identified. These valleys indicate the space between the lines of the text. Finally, in the third category of methods use a kind of smearing [56,63]. Shi and Govindraju used a fuzzy run length is used to segment lines. This measure is calculated for every pixel on the initial image and describes how far one can see when standing at a pixel along horizontal direction. By applying this measure, a new grayscale image is created which is binarized and the lines of text are extracted from the new image.

The text line extraction problem can also be seen from an Artificial Intelligence perspective. The aim is to cluster the connected components of the document into homogeneous sets that correspond to the text lines of the document. Shi et al. [57], make use of the Adaptive Local Connectivity Map. The input to the method is a grayscale image. A new image is calculated by summing the intensities of each pixel's neighbors in the horizontal direction. Since the new image is also a grayscale image, a thresholding technique is applied and the connected components are grouped into location maps by using a grouping method.

There are two distinct approaches determining word segmentation. One by calculating connected components, after taking as input a text line image. The distances between adjacent connected components are measured using a metric such as the Euclidean distance, the bounding box distance or the convex hull metric [33, 36, 52]. Finally, a threshold is defined which is used to classify the calculated distances as either inter-word or inter-characters gaps.

At a certain position of a text line, it has to be decided whether the considered position belongs to a letter of a word, or to a space between two words. For this purpose, three different recognizers based on Hidden Markov Models are designed, and results of writer- dependent as well as writer-independent experiments are reported in the paper. All the above techniques do not deal successfully with the separation of vertically connected text lines which is a crucial aspect towards word recognition. For the word segmentation problem, there is an inefficient discrimination between inter-word and intra-word distances.

The writing lines estimation is an essential step in preprocessing level. Based on these lines, base- and top-line, normalization and the feature extraction is implemented.

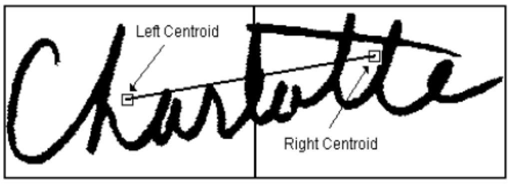


Figure 6: Writing line estimation.

2.3.3 BASE-LINE ESTIMATION

Horizontal and vertical projection histogram (Figure 7, Figure 8) based approaches are widely used for recognition of handwritten text. The projection method is robust and easy to implement but it needs straight lines and long words, which is often not the case for single handwritten words. This approach is often implemented by transforming the binary word image into a Hough parameter space, with a sub-segment maximum detection [52].

Figure 7: Horizontal histogram.

Figure 8: Vertical histogram.

Image may require polygonal processing in order to calculate robust features from the skeleton. These features are used for classifying the connected components into baseline relevant and baseline irrelevant areas. In a subsequent step a regression analysis of points of the relevant objects is done to estimate the final baseline position. Skeleton based approach gives better results.

2.3.4 TOP-LINE ESTIMATION

The horizontal projection histogram is often used to estimate the upper baseline (top-line) too. We again use the Hough space but now we calculate the vertical gradient of the Hough space to determine the top-line by selecting the maximal gradient value within a search area above the position of the baseline. This straight forward method provides good results.

2.4 IMAGE NORMALIZATION

2.4.1 STROKE THICKNESS NORMALIZATION

For estimating normalization parameters and for performing normalization, skeletonization of the word image is needed. The skeletonization is performed on the contour representation resulting directly in a graph representation of the skeleton [16]. The advantages of this method are the following:

1- The data structure is easily accessible,

2- The pen dependent line width is removed

3- The connected components are easily to obtain.

2.4.2 GEOMETRICAL TRANSFORMATIONS

Using the estimated normalization parameters, the skeleton is then normalized with respect to the position of the writing lines, shear angle, height, and width. The normalization involves in detail:

Rotation, resulting in horizontal baseline,

Shearing, resulting in vertical direction

Vertical compression or dilation for height normalization, resulting in a constant vertical distance between the writing lines and constant heights for ascender and descender regions.

Horizontal compression or dilation with linear characteristic, for constant average character width.

2.4.3 RE-THICKENING

Re-thickening is done by Gaussian filtering of the skeleton image, resulting in a gray level image. This gray level image will be used for the feature extraction in the method based on sliding window.

2.5 FEATURE EXTRACTION

Three different feature extraction methods are presented in this section. The Sliding Window with Pixel feature extraction method needs normalized gray level images with the top- and base- lines information. The skeleton direction based feature extraction method is based on the different main zones, which requires normalized skeleton graphs with only baseline information. The Sliding Window with Local feature extraction method uses only the original image and the baseline information, without any others preprocessing.

2.5.1 SLIDING WINDOW WITH PIXEL FEATURE

Feature extraction using a sliding window that uses HMM recognizer and a mixed feature vector [29, 30]. The window width is a constant system parameter, while window height depends on each word image. The sliding window is shifted along the original word image (without any preprocessing) from left to right and a feature vector is calculated for each frame.

The feature extraction method using sliding window is directly based on an image representation of the script using pixel values as basic features. A rectangular window is shifted in respect to the writing direction from left to right (or right to left for Arabic) across the normalized gray level script image and generates a feature vector (frame). This results in a vast amount of features for each frame. In order to reduce the number of features a Karhunen-Lo`eve Transformation is performed on the gray values of each frame. Figure 9 gives an example of the sliding window feature extraction method. The transformation matrices are computed from the training data. The three columns of the sliding window are concatenated to a feature vector. This is done at each position of the window.

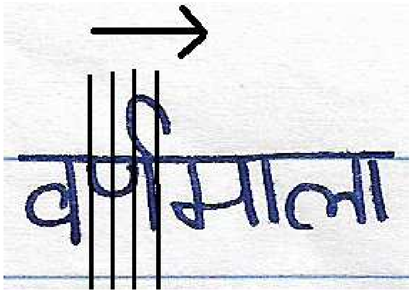


Figure 9: Sliding window feature extraction.

2.6 IN LEXICON AND OUT-OF-LEXICON WORDS

Very often the researchers on handwriting recognition systems rely on linguistic resources, particularly the static dictionaries [15, 17, 18, 24, 25, 28, 64]. The word segments are heuristically matched with lexical listing to arrive at correct recognition in a faster way. However, over the past three decades, several research works have also made provisions for words that do not belong to the used word lexicon [2, 47], i.e. Out-Of-Vocabulary words (OOV words). The OOV words represent an important source of error in handwritten text recognition systems. Several approaches have been proposed to address this issue.

Much progress has been made in detection and recovery of OOV words in Automatic Speech Recognition (ASR) systems. Often the approaches have relied either on OOV detection or lexicon selection. The OOV detection based approaches [9, 49, 66] are aimed at locating the OOV regions in the ASR and searching their matching phoneme sequence in the speech. It uses hybrid language models wherein not only the in-vocabulary word but also their sub-word units are used. Model requires careful selection of sub-word units that often lead to increased chances of error [53].

In handwriting recognition, the methods addressing the OOV problem generally rely on the same techniques used in ASR systems. These methods can be subdivided also into two categories. One-step approaches try to recover OOV words during the recognition process by increasing the vocabulary size, which generally increases the computational complexity and the confusability in the data. An alternative approach is to use sub-word units, either to estimate a full sub-word LM or to generate a hybrid LM that incorporates both words and sub-words. The performance of sub-word modeling approaches will however depend on the language model design and most importantly on the properties of the training corpus compared to those of the test corpus. In addition, they can produce some words that do not belong to the language and consequently their recognition performance drastically drops. The second category is based on two processing steps: OOV detection and OOV recovery [48] .

The detailed study of the existing handwriting recognition research works shows that most of the existing word and text recognition systems integrate only the OOV words recovery without any preliminary detection step. Such detection is essentially based on the comparison of the confidence scores of the recognized words with a heuristic threshold whose value is determined through several experiments.

Considering the OOV words recovery, most of the proposed handwritten text recognition systems rely on the so-called sub lexical units. These systems enrich the word lexicon by decomposing the words into different sub-word lexical units. These units can be letters or syllables for several scripts. They can be, Part of Arabic Words (PAWs) or morphemes, particularly, for the Arabic script.

CHAPTER 3

EXPERIMENTAL WORK, RESULTS AND DISCUSSION

3.1 OVERVIEW

The main objective of this study have been to make original effort to understand basics of approaches so as handwritten devanagari alphabets, words and sentences could be recognized seamlessly. First, I have attempted to create vertical and horizontal signatures of handwritten Devanagari symbols, words and sentences for use in letter recognition engine. Subsequently I tried to evaluate if alphabet signatures are machine readable and really unique so as they can be identified in words and sentences.

GNU Octave Version 4.4.0 was used to carry out the numerical experiments on image handling, image manipulation and feature extraction. It was run on a Pentium III machine in Windows 10 environment. It is a high level interpreted programming language that supports many common C standard library functions, and also certain UNIX system calls and functions. Hand written images were acquired on a HP2130 scanner. Subroutines were written for digitization of image, its binarization, generation of histograms, noise removal and extraction of character features. The binaries for vertical and horizontal histograms for different characters are machine readable and can be combined with lexical heuristics to create fast handwriting recognition systems. The text line hypotheses construction can be carried out relying on a Word Statistical Language Model WSLM, a PAW Statistical Language Model PSLM or a Morpheme Statistical Language Model MSLM.

3.2 LETTER RECOGNITION SYSTEM

Irrespective of whether images required slanting removal to generate before handling. These may be in color, greyscale or line art is immaterial (Figure 1). In this work the original gray scale image was normalized to a fixed height of 96 pixels.

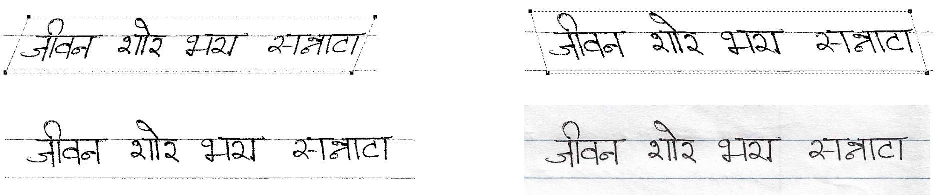


Figure 1: Grey scale, line art and slant images.

3.3 HISTOGRAM GENERATION AND ANALYSIS

Initially the image is digitized and converted to binary data using Octave algorithm. The binaries were subsequently, subjected to algorithms in Figure 2 and Figure 3, to generate horizontal and vertical histograms feature sequences for the input images. Histograms of different alphabets, words, sentences and paragraphs were generated.

```
pkg load image
c = imread("image.jpg")                ▷ Reading the image
c = im2bw(c)                           ▷ Conversion to black and white
c = medfilt2(c)                        ▷ Applying median filter
y = zeros(1, rows(c))                  ▷ New array to store no of black pixels in a row
count ← 0                              ▷ Initialize count to zero
for i = 1 : rows(c) do
    count ← 0
    for j = 1 : columns(c) do
        if c(i, j) == 0 then           ▷ For a row count the number of black pixels
            count ← count + 1
        end if
    end for
    y(i) ← count
end for
stem(y)                                ▷ Create a 2D plot to obtain the histogram
```

Figure 2: Octave routine for generation of horizontal histogram.

```
pkg load image
c = imread("image.jpg")                ▷ Reading the image
c = im2bw(c)                           ▷ Conversion to black and white
c = medfilt2(c)                        ▷ Applying median filter
x = zeros(1, rows(c))                  ▷ New array to store no of black pixels in a column
count ← 0                              ▷ Initialize count to zero
for i = 1 : columns(c) do
    count ← 0
    for j = 1 : rows(c) do
        if c(i, j) == 0 then           ▷ For a column count the number of black pixels
            count ← count + 1
        end if
    end for
    x(i) ← count
end for
stem(x)                                ▷ Create a 2D plot to obtain the histogram
```

Figure 3: Octave routine for generation of vertical histogram.

Histograms for two alphabets are given in Figure 4. As the data for the histograms are numeric one, it carries good potential for computational comparison and pattern matching. Visually it is seen that the horizontal histograms gave the first indication of uniqueness of individual hand written characters (Figure 4). The histograms features of all of the handwritten Devanagari alphabets are uniquely distinguishable. This would facilitate machine based recognition of Devanagari handwriting easier to interpret, when combined with heuristics for character recognition and matching with lexical database.

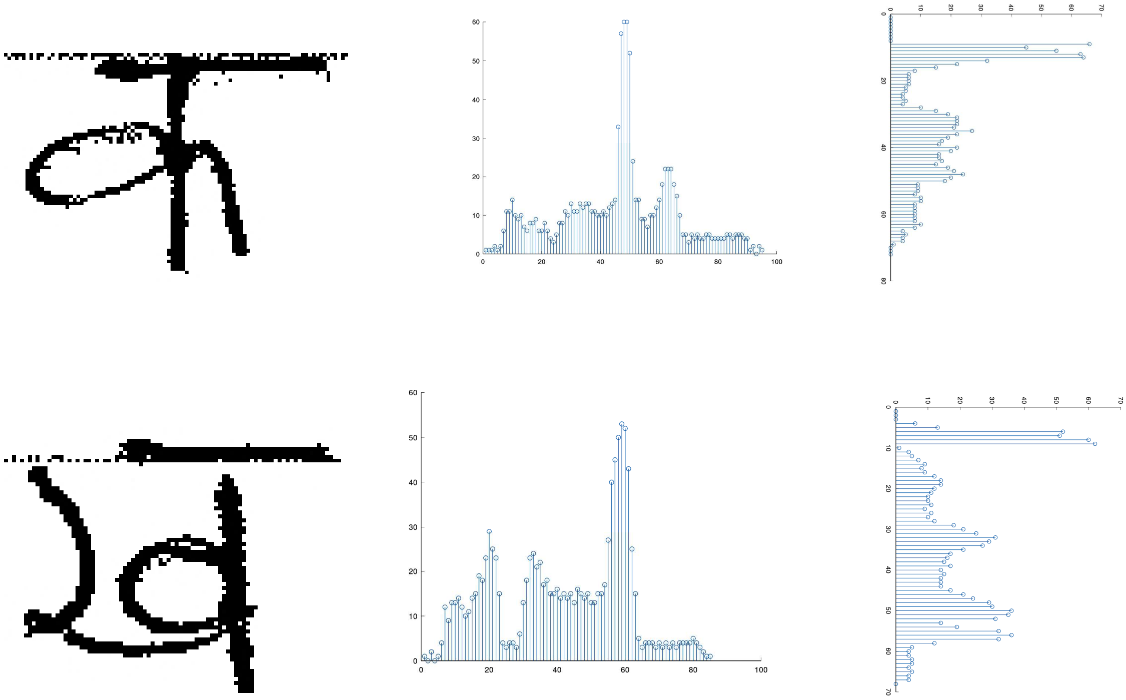


Figure 4: Histograms for the first two hand written Devanagari consonants.

The lines and different alphabets are readily discerned using vertical and horizontal histograms as shown in Figure 5 and Figure 6. Despite of the shirorekha the alphabet columns are clearly discernible in the vertical histogram and the alphabet rows, i.e. lines are visible in the horizontal histogram. Also that the ruled lines on the paper are seen as sharp histograms between the alphabet lines. The horizontal histogram was used to cut the lines. Subsequently the vertical histogram was used to cut and recognize the individual handwritten alphabets.

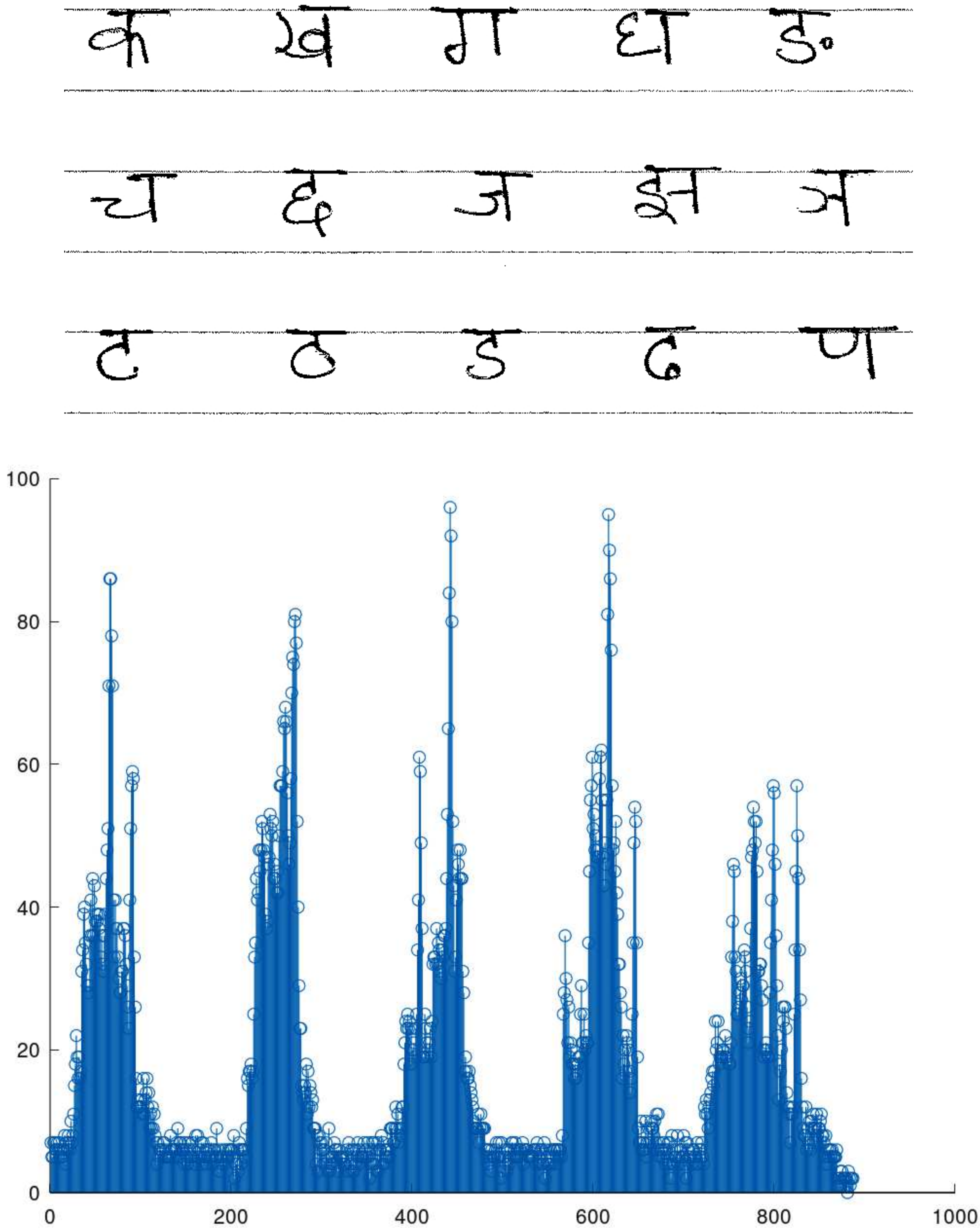


Figure 5: Five columns of handwriting and their vertical histogram.

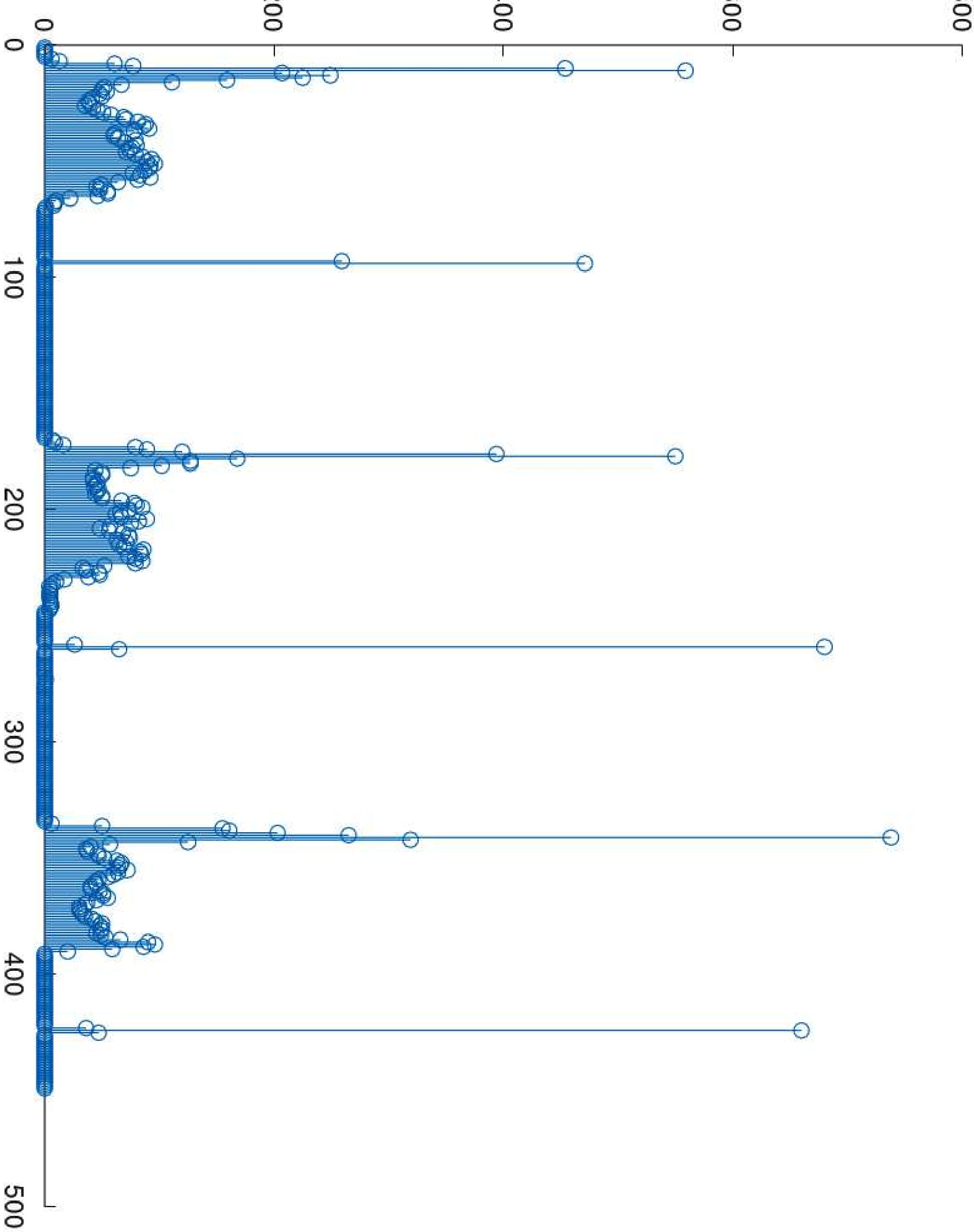


Figure 6: Horizontal histogram for three rows of handwriting.

The handwritten alphabet sequence in a word and their histograms are given in Figure 7. It can be seen that individual alphabets and symbols and their fragments are identifiable in the histograms. The histograms of letter "va" in the word and left half of letter "ka" have close resemblance in the vertical histograms. It could be said that partial absence of preceding shirorekha in "ka" is causing slight variation from the horizontal histogram of "va". However such variations can be eliminated if routine for removal of shirorekha is incorporated.

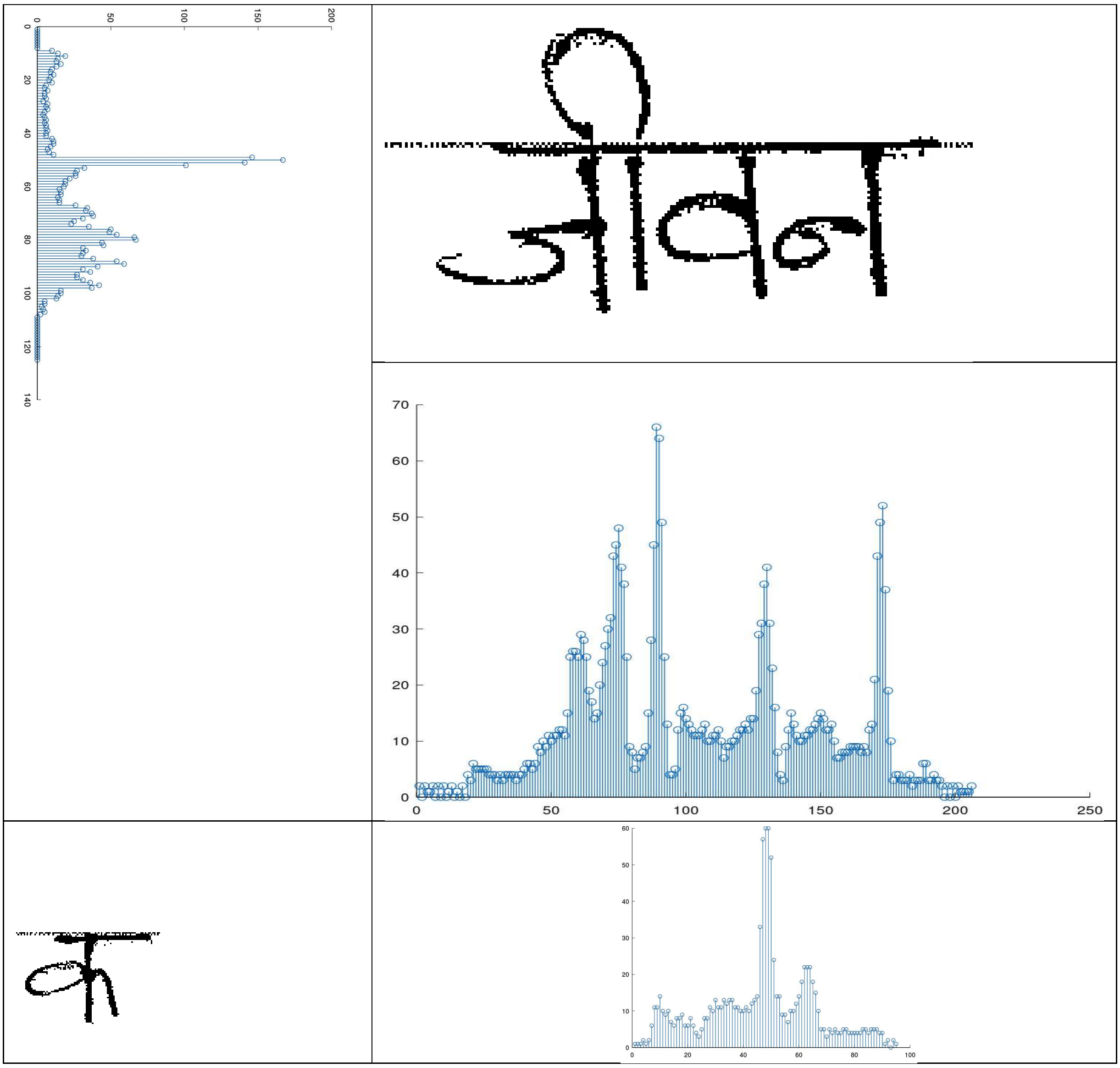


Figure 7: Histogram of a linked word and comparison of letter segment with word fragment.

The handwritten words in Devanagari script give distinct histograms interspersed with blank spaces as seen in the sentence and its histogram in Figure 8. Unlike underlining the shirorekha in the Devanagari handwriting is useful in demarcating words from continuous run of the script. However shirorekha can be readily removed using baseline correction to each line.

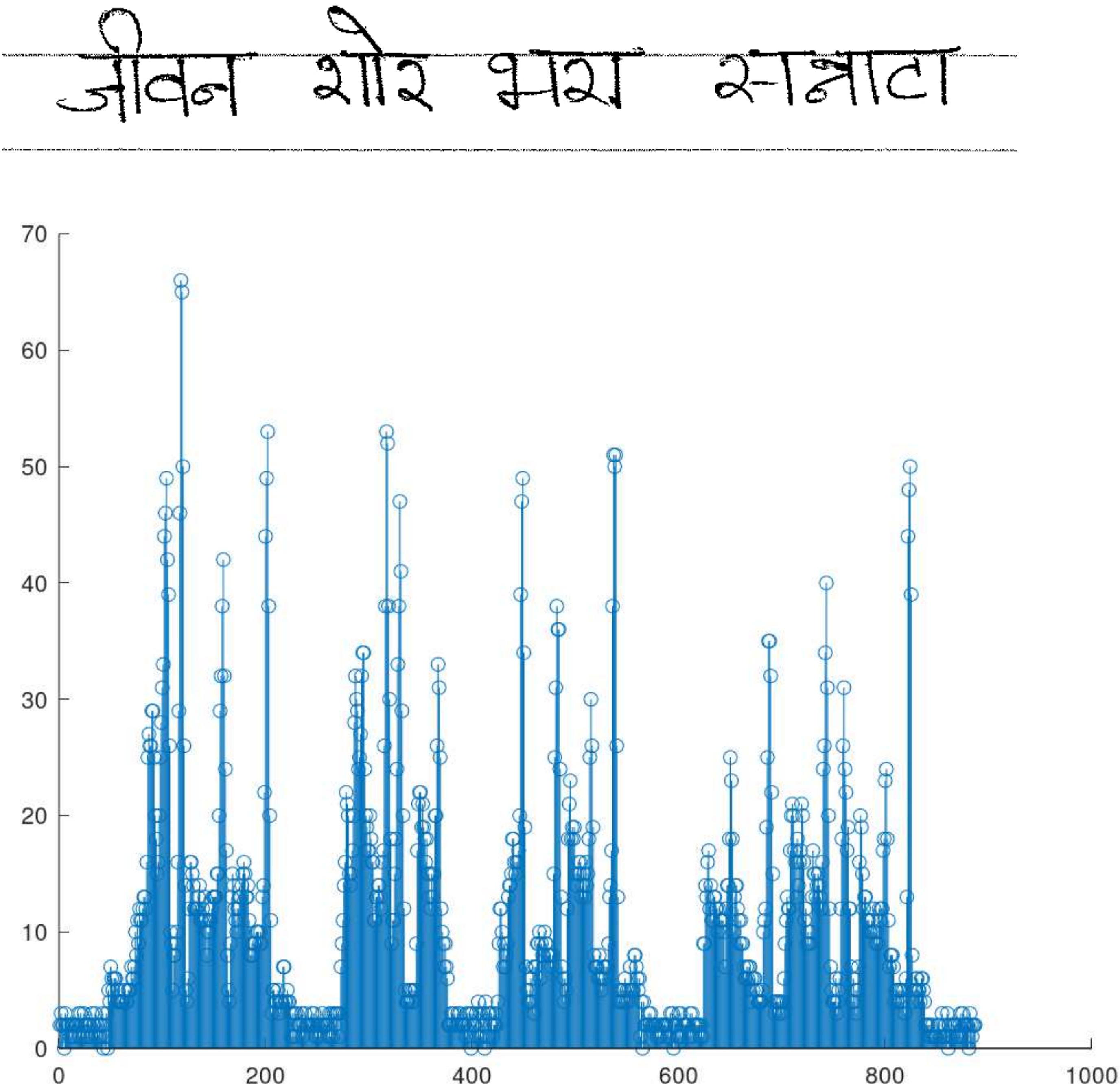


Figure 8: Vertical histogram of a sentence line.

3.4 EXTRACTION OF WORDS AND INDIVIDUAL HANDWRITTEN SYMBOLS

Octave sub-routine as given in Figure 9 was written to expand and split characters in the sentence. 3 and 4 pixels depth were used to locate word and symbol limits. Results are given in Figure 10. As the pen stroke in the shirorekha was used to determine the pixel depth there are some false character demarcation. These appear to be very narrow to be a character and hence, the routine can be modified to eliminate false recognitions.

```
th ← 3                                                        ▷ Define threshold value
for i = 1 : (columns(x) − 3) do
    if x(i) <= th then          ▷ Check if current columns value is less than threshold
        if x(i + 1) <= th&x(i + 2) <= th&x(i + 3) <= th then
            x(i) ← 0
        else
            x(i) ← th + 1
        end if
    else                                      ▷ If current value is more than threshold
        if x(i + 1) <= th&x(i + 2) <= th&x(i + 3) <= th then
            x(i) ← 0
        end if
    end if
end for
result = zeros(size(c))                                  ▷ Create matrix of image size
for k = 1 : columns(c) do                                    ▷ For individual columns
    if x(k) <= th then                     ▷ If value is less than equal to threshold
        for l = 1 : rows(c) do                                  ▷ For individual rows
            result(l, k) ← 0                            ▷ Make the whole column black
        end for
    else
        for l = 1 : rows(c) do
            result(l, k) ← c(l, k)                  ▷ Image value copied to new matrix
        end for
    end if
end for
```

Figure 9: Octave sub-routine for splitting individual words and symbols in the handwritten Sentence.

th<=4pixel

th<=3 pixel

th<=2 pixel

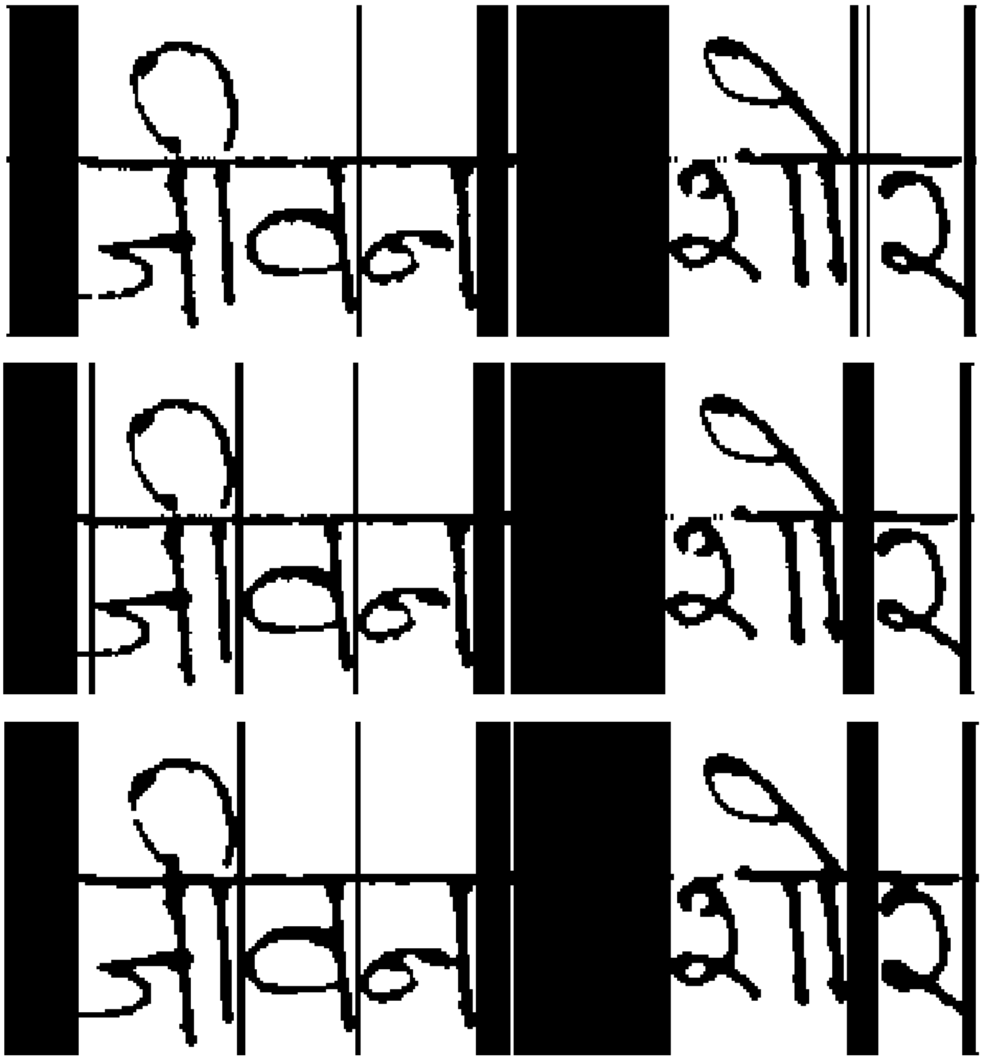


Figure 10: Effect of dilation and pixel depth on character splitting.

The non-character area was recognized implementing a minimum threshold on x axis. It is seen that when threshold was set to 3 or 4 pixels depth in the splitting routine some false, but narrow characters were demarcated. However when image was dilated and threshold was set to ≤ 4 pixels a clear and unambiguous demarcation of individual symbols in the handwriting was obtained.

3.5 USE OF MEDIAN FILTER FOR NOISE REMOVAL

Median filter routine in Octave (Figure 11) was used to enhance image quality and stroke thickness. It could be seen in Figure 12 that machine readability of image was tremendously enhanced after this process. Definitely images subjected to median would facilitate better character recognition.

```
pkg load image
c = imread("image.jpg")        ▷ Reading the image
c = im2bw(c)                   ▷ Conversion to black and white
c = medfilt2(c)                ▷ Applying median filter
```

Figure 11: Median filter routine.

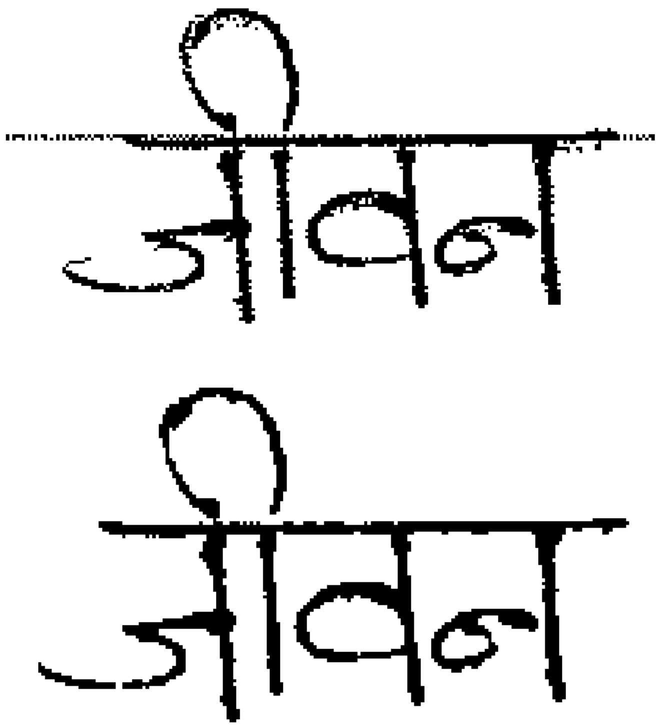


Figure 12: Image quality without (top) and with (below) median filter.

3.6 MOVING WINDOW IMAGE RECOGNITION

Moving window scheme was used for recognizing the character pattern. For the purpose a 5 pixel window was chosen wherein 2+3 arrangement was used. That each new window has 2 pixel overlap with the preceding window. The Octave sub-routine for moving window recognition is given in Figure 13. The outcome of the routine implementation is given in Figure 14. It is seen that image pattern is easily discernible that can be used in HMM to recognize the words. However due to constraints of time and circumstantial pandemic situation programming for implementation of HMM sub-routine was beyond the scope of this work.

```
pkg load image
c = imread("image.jpg")                ▷ Reading the image
c = im2bw(c)                           ▷ Conversion to black and white
c = medfilt2(c)                        ▷ Applying median filter
x = columns(c)

x = x − 2
x = x/3
x = x ∗ 2
x = columns(c) + x
d = zeros(rows(c), x)
count = 0
k = 1
for i = 1 : columns(c) do
    count ← count + 1
    for j = 1 : rows(c) do
        d(j, k) ← c(j, i)
    end for
    k ← k + 1
    if count == 5 then
        count ← 0
        i ← i − 3
        k ← k + 3
    end if
end for
```

Figure 13: Octave sub-routine for moving window splitting of characters.

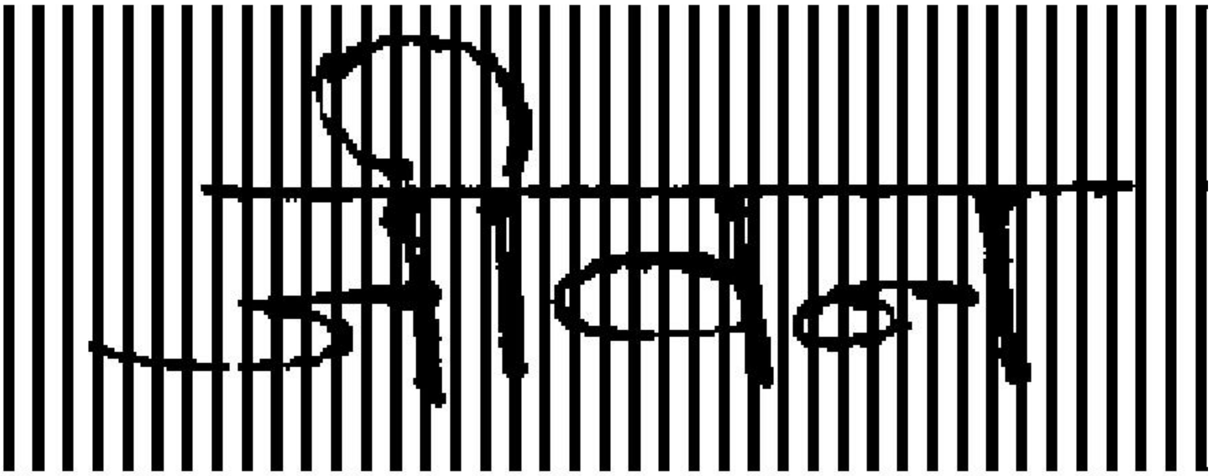


Figure 14: Output of moving window subroutine.

CHAPTER 4

CONCLUSIONS AND FUTURE WORK

It was a preliminary endeavor to gain first hand understanding of scope, problems, hurdles and challenges to be encountered in the development of a Devanagari handwriting recognition engine. This study demonstrates that routines and modules can easily be created for image pre-processing, binarization, noise removal, segmentation of sentences and words and for moving window recognition purposes. It was found that the vertical histograms represent unique signatures of most of the Devanagari characters and symbols. Therefore these binary files can principally be used for character recognition. Unlike the Roman script the vertical histograms for symbols in Devanagari script are feature rich and thus would facilitate accurate identification of the characters and words. The optimized histogram libraries for binaries of different symbols would be developed in future. Modules for Heuristic retrieval of symbols and lexical words would be created. The insights gained through this preliminary endeavor could be used to develop an App to facilitate Devanagari handwriting based inputs for message writing and note taking.

I am encouraged with these results and preliminary findings. It would help me to develop engine for automated image manipulation, feature extraction and character and symbol matching device. The 14 weeks had been very short to learn a new topic, particularly more so in the time of COVID-19 lockdown with debilitating restrictions.

Following work may be undertaken to bring this work to a logical conclusion.

1. Creation of a library of histograms for all handwritten Devanagari alphabets and symbols.
2. Writing subroutines for de-skewing, slanting removal and normalization of text in the images.
3. Inputting moving window sub-routine output to HMM recognition model for recognition of handwritten characters and symbols.
4. Writing Heuristic sub-routines for incorporation of lexical databases into handwriting recognition.
5. Writing sub-routines for in text identification of word fragments for recognition of out-of-lexicon words.

CHAPTER 5

REFERENCES

1. Abandah, G.A., Jamour, F.T. and Qaralleh, E.A. (2014) ‘Recognizing handwritten Arabic words using grapheme segmentation and recurrent neural networks’, Int. J. Document Anal. Recognit., vol.17, pp.275-291.

2. Ahmad, I., Mahmouda, S.A. and Fink, G.A. (2016) ‘Open-vocabulary recognition of machine-printed Arabic text using hidden Markov models’, Pattern Recognit., vol.51, pp.97-111.

3. Arifoğlu, D. and Duygulu, P. (2011) ‘Retrieval of Ottoman Documents’, IEEE 19th Signal Processing and Communications Applications Conference, pp. 526-529.

4. Arivazhagan, M., Srinivasan, H. and Srihari, S.N. (2007) ‘A Statistical App.roach to Handwritten Line Segmentation’, Document Recognition and Retrieval XIV, Proceedings of SPIE, San Jose, USA,pp. 6500T-1-11.

5. Bluche, T., Louradour, J. and Messina, R. (2017) ‘Scan, Attend and read, pp. end-to-end handwritten paragraph recognition with MDLSTM attention’, Proceeding of International Conference on Document Analysis and Recognition, ICDAR, pp. 1050-1055.

6. Bluche, T., Ney, H. and Kermorvant, C. (2013) ‘Feature extraction with convolutional neural networks for handwritten word recognition’, Proceeding of International Conference on Document Analysis and Recognition, ICDAR, pp. 285-289.

7. Britto, A.D.S., Sabourin, R., Bortolozzi, F. and Suen, C.Y. (2001) ‘A two-stage HMM-based system for recognizing handwritten numeral strings’ Proceedings of the International Conference on Document Analysis and Recognition, Seattle, pp.396-400.

8. Bruzzone, E. and Coffetti, M.C. (1999) ‘An Algorithm for Extracting Cursive Text Lines’, Proceedings of the Fifth International Conference on Document Analysis and Recognition, Bangalore, India, pp. 749.

9. Chen, W., Ananthakrishnan, S., Prasad, R. and Natarajan, P. (2013) ‘Variable span out-of-vocabulary named entity detection, Proceedings of Annual Conference of the International Speech Communication Association, pp. 3761-3765.

10. Chherawala, Y. , Roy, P.P. and Cheriet, M. (2017) ‘Combination of context-dependent bidirectional long short-term memory classifiers for robust online handwriting recognition’, Pattern Recognit. Lett., vol.90, pp.58-64.

11. Dehghan, M., Faeza, K., Ahmadi, M. and Shridhar, M. (2001) 'Handwritten farsi (arabic) word recognition, pp. a holistic app.roach using discrete HMM', Pattern Recognition, vol.34, pp.1057-1065.

12. Doetsch, P., Zeyer, A., Voigtlaender, P., Kulikov, I., Schluter, R. and Ney, H. (2017) 'RETURNN, pp. the RWTH extensible training framework for universal recurrent neural networks' Proceeding of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, pp.5345-5349.

13. El-Yacoubi, A., Gilloux, M., Sabourin, R., and Suen, C.Y. (1999) An HMM-based approach for off-line unconstrained handwritten word modeling and recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.21, pp.752-760.

14. Ernard, A.B., Menasri, F., El-Hajj, R., Mokbel, C., Kermorvant, C., Likforman, L. (2011) 'Dynamic and contextual information in HMM Behaviour for handwritten word recognition', IEEE Trans. Pattern Anal. Mach. Intell., vol.99, pp.2066-2080.

15. Esfahani, A., Farahnak, F. and Katanforoush, A. (2015) 'A stroke-level wordnet for Farsi handwriting recognition', Proceeding of Iranian Conference on Machine Vision and Image Processing, MVIP, pp. 232-235.

16. Ferreira, A. and Ubeda, S. (1994) 'Ultra fast parallel countour tracking with application to thinning' Pattern Recog. Vol.27, pp.867-878.

17. Garcia-Salicetti, R., Dorizzi, B., Gallinari, P. and Wimmer, Z. (2001) 'Maximum mutual information training for an online neural predictive handwritten word recognition system', Int. J. Document Anal. Recognit., vol.4, pp.56-68.

18. Ge, Y. and Huo, Q. (2002) 'A study on the use of CDHMM for large vocabulary offline recognition of handwritten Chinese characters, Proceedings of the International Workshop on Frontiers in Hand-writing Recognition, Niagara on the Lake, Canada, pp.334-338.

19. Graves, A. and Gomez, F. (2006) 'Connectionist temporal classification, pp. labelling unsegmented sequence data with recurrent neural networks', Proceeding of Proceedings of the 23rd international conference on Machine learning, ICML, pp.369-376.

20. Graves, A. and Schmidhuber, J. (2008) 'Online handwriting recognition with multidimensional recurrent neural networks', Proceeding of International Conference on Neural Information Processing Systems, NIPS, pp.545-552.

21. Graves, A. and Schmidhuber, J. (2009) 'Online handwriting recognition with multidimensional recurrent neural networks', Proceedings of the Advances in Neural Information Processing Systems, NIPS, pp. 545-552.

22. Hull, J.J. (1996) 'Incorporating language syntax in visual text recognition with a statistical model', IEEE Trans. Pattern Anal. Mach. Intell. Vol.18, pp.1251-1255.

23. Jemni, S.K., Kessentini, Y., Kanoun. S. and Ogier, J. (2018) 'Online Arabic handwriting recognition using BLSTMs combination', Proceeding of IAPR International Workshop on Document Analysis Systems, DAS, pp. 31-36.

24 Kanoun, R., Alimi, A.M. and Lecourtier, Y. (2011) 'Natural language morphology integration in offline Arabic optical text recognition', IEEE Trans. Syst. Man Cybern., vol.41, pp.579-590.

25. Kessentini, Y., Paquet, T. and Hamadou, A.B. (2007) 'A multi-stream app.roach to off-line handwritten word recognition', Proceeding of International Conference on Document Analysis and Recognition, ICDAR, pp. 317-321.

26. Kessentini, Y., Paquet, T. and Hamadou, A.B. (2010) 'Offline handwritten word recognition using multi-stream hidden Markov models', Pattern Recognition Letters, vol.31, pp.60-70.

27. Koerich, A.L., Sabourin, R. and Suen, C.Y. (2003) 'Large vocabulary off-line handwriting recognition: a survey', Pattern Anal. Appl., vol.6, pp.97-121.

28. Koerich, A.L., Sabourin, R. and Suen, C.Y. (2003) 'Lexicon-driven HMM decoding for large vocabulary handwriting recognition with multiple character models', Int. J. Document Anal. Recognit. Vol.6, pp.26-144.

29. Likforman-Sulem, L. , Hanimyan, A. and Faur, C. (1995) 'A Hough Based Algorithm for Extracting Text Lines in Handwritten Documents', Proceedings of the Third International Conference on Document Analysis and Recognition, Montreal, Canada, pp. 774-777.

30. Louloudis, F., Halatsis, K., Gatos, B. and Pratikakis, I. (2006) 'Block-Based Hough Transform Mapp.ing for Text Line Detection in Handwritten Documents', 10th International Workshop on Frontiers in Handwriting Recognition, La Baule, France, pp. 515-520.

31. Luthy, F., Varga, T. and Bunke, H. (2007) 'Using Hidden Markov Models as a Tool for Handwritten Text Line Segmentation', Ninth International Conference on Document Analysis and Recognition, Curitiba, Brazil, pp.8-12.

32. Madhvanath, S. and Govindaraju, V. (2001) 'The role of holistic paradigms in handwritten word recognition', IEEE Trans. Pattern Anal. Mach. Intell., vol.23, pp.149-164.

33. Mahadevan, U. and Nagabushnam, R.C. (1995) 'Gap metrics for word separation in handwritten lines', Third International Conference on Document Analysis and Recognition, Montreal, Canada, pp.124-127.

34. Manmatha, R. and Rothfeder, J.L. (2005) 'A Scale Space Approach for Automatically Segmenting Words from Historical Handwritten Documents', IEEE Trans. Pattern Analysis and Machine Intelligence, vol.27, pp.1212-1225.

35. Markov, A.A. (1913) 'Example of statistical investigations of the text which demonstrates the connection of events in a chain' Bulletin de l'Académie Impériale des Sciences de St.-Pétersbourg, pp.153-162. (in Russian)

36. Marti, U.V. and Bunke, H. (2001) 'Text Line Segmentation and Word Recognition in a System for General Writer Independent Handwriting Recognition', Sixth International Conference on Document Analysis and Recognition, Seattle, WA, USA, pp.159-163.

37. Nopsuwanchai, R., Biem, A. and Clocksin, W.F. (2006) 'Maximization of mutual information for offline Thai handwriting recognition', IEEE Trans. Pattern Anal. Mach. Intell. Vol.28, pp.1347-1351.

38. Oprean, C., Likforman-Sulem, L., Popescu, A. and Mokbel, C. (2015) 'Handwritten word recognition using web resources and recurrent neural networks', Int. J. Document Anal. Recognit., vol.18, pp.287-301.

39. Pechwitz, M. and Maergner, V. (2002) 'Baseline estimation for Arabic handwritten words', Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR), pp.479-484.

40. Pechwitz, M. and Margner, V. (2005) 'HMM based app.roach for handwritten Arabic word recognition using the IFN/ENIT – database', In 7th International Conference on Document Analysis and Recognition (ICDAR), pp.890-894.

41. Pham, V., Bluche, T., Kermorvant, C. and Louradour, J. (2014) 'Dropout improves recurrent neural networks for handwriting recognition', Proceeding of International Conference on Frontiers in Handwriting Recognition, ICFHR, pp.285-290.

42. Pitrelli, J.F. and Roy, A. (2003) 'Creating word-level language models for large-vocabulary handwriting recognition', Int. J. Document Anal. Recognit., vol.5, pp.126-137.

43. Plötz, T. and Fink, G.A. (2009) 'Markov models for offline handwriting recognition: a survey.' IJDAR, vol.12, pp.269-298.

44. Plötz, T. and Fink, G.A. (2009) 'Markov models for online handwriting recognition, pp. a survey', Int. J. Document Anal. Recognit. Vol.12, pp.269-298.

45. Procter, S., Illingworth, J. and Mokhtarian, F. (2000) 'Cursive handwriting recognition using hidden Markov models and a lexicon-driven level building algorithm', Proceeding of IEE Vision, Image and Signal Processing, vol.4, pp.332-339.

46. Pu, Y. and Shi, Z. (1998) 'A Natural Learning Algorithm Based on Hough Transform for Text Lines Extraction in Handwritten Documents', Proceedings of the 6th International Workshop on Frontiers in Handwriting Recognition, Taejon, Korea, pp. 637-646.

47. Puigcerver, G., Toselli, A.H. and Vidal, E. (2016) 'Querying out-of-vocabulary words in lexicon-based keyword spotting', Neural. Comput. Appl., vol.28, pp. 2373-2382.

48 Quiniou, S., Cheriet, M. and Anquetil, E. (2009) 'Handling out-of-vocabulary words and recognition errors based on word linguistic context for handwritten sentence recognition', Proceeding of International Conference on Document Analysis and Recognition, ICDAR, pp. 466-470.

49. Rastrow, A., Sethy, A., Ramabhadran, B. and Jelinek, F. (2009) 'Towards using hybrid word and fragment units for vocabulary independent LVCSR systems', Proceedings of Annual Conference of the International Speech Communication Association, pp. 1931-1934.

50. Rawls, S., Cao, H., Kumar, S. and Natarajan, P. (2017) 'Combining convolutional neural networks and LSTMs for segmentation-free OCR', Proceedings of International Conference on Document Analysis and Recognition, ICDAR, pp. 155-160.

51. Schuster, M. and Paliwal, K.K. (1997) ‘Bidirectional recurrent neural networks’, IEEE Trans. Signal Process., vol.45, pp.2673-2681.

52. Seni, G. and Cohen, E. (1994) ‘External Word Segmentation of Off-line Handwritten Text Lines’, Pattern Recognition, vol.27, pp.41-52.

53. Shaik, M.A.B., Mousa, A.E.D., Hahn, S., Schluter, R. and Ney, H. (2015) Improved strategies for a zero OOV rate LVCSR system, in IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 5048-5052.

54. Sharma, M.K., Dhaka, V.S. (2019) ‘Segmentation of handwritten words using structured support vector machine’, Pattern Anal. Applic. https://doi.org/10.1007/s10044-019-00843-x.

55. Sharma, M.K., Dhaka, V.S. Segmentation of handwritten words using structured support vector machine. Pattern Anal Applic (2019). https://doi.org/10.1007/s10044-019-00843-x.

56. Shi, Z. and Govindaraju, V. (2004) ‘Line Separation for Complex Document Images Using Fuzzy Runlength’, First International Workshop on Document Image Analysis for Libraries, pp. 306.

57. Shi,, Z., Setlur, S. and Govindaraju, V. (2005) ‘Text Extraction from Gray Scale Historical Document Images Using Adaptive Local Connectivity Map’, Eighth International Conference on Document Analysis and Recognition, Seoul, Korea, pp.794-798.

58. Sihwi, S.W., Fikri, K. and Aziz, A. (2019) ‘Dysgraphia Identification from Handwriting with Support Vector Machine Method’, J. Phys.: Conf. Ser. 1201 01205.

59. Tuba, E., Hrosik, R.C., Alihodzic, A., Jovanovic, R. and Tuba, M. (2020) ‘Support Vector Machine Optimized by Fireworks Algorithm for Handwritten Digit Recognition’, in “Modelling and Development of Intelligent Systems”, Simian, D. and Stoica, L.F. (Eds.), Springer Nature Switzerland, pp. 187–199.

60. Tuba, E., Tuba, M. and Beko, M. (2016) ‘Support vector machine parameters optimization by enhanced fireworks algorithm’, in “Advances in Swarm Intelligence”, Tan, Y., Shi, Y., Niu, B. (eds.) Springer, pp. 526-534.

61. Vamvakas, F., Gatos, B. and Perantonis, S.J. (2010) ‘Handwritten character recognition through two-stage foreground sub-sampling’, Pattern Recogn. Vol.43, pp.2807-2816.

62. Voigtlaender, P., Doetsch, P. and Ney, H. (2016) ‘Handwriting recognition with large multidimensional long short-term memory recurrent neural networks’, Proceeding of International Conference on Frontiers in Handwriting Recognition, ICFHR, pp.228-233.

63. Wahl, F.M., Wong, K.Y. and Casey, R.G. (1982) ‘Block Segmentation and Text Extraction in Mixed Text/Image Documents’ Computer Graphics and Image Processing, vol.20, pp.375-390.

64. Xue, G. and Govindaraju, V. (2002) ‘On the dependence of handwritten word recognizers on lexicons’, IEEE Trans. Pattern Anal. Mach. Intell., vol.24, pp.1553-1564.

65. Yan, R., Peng, L., Bin, G., Wang, S. and Cheng, Y. (2017) 'Residual recurrent neural network with sparse training for online Arabic handwriting recognition', Proceedings of International Conference on Document Analysis and Recognition, ICDAR, pp. 1031-1037.

66. Young, S.R. (1994) 'Detecting misrecognitions and out-of-vocabulary words', Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, pp. 21-24.