WRITER IDENTIFICATION SYSTEM

A PROJECT REPORT

Submitted by

ASHOK KRISHNAN.S	**2011115507**
JAYVIZ SELVIN.X	**2011115519**
SANJITH.J.K	**2011115549**

to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

for partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY

COLLEGE OF ENGINEERING GUINDY

ANNA UNIVERSITY

CHENNAI 600 025

APRIL 2015

BONAFIDE CERTIFICATE

Certified that this project report titled **"WRITER IDENTIFICATION SYSTEM"** is the bonafide work of **"Ashok Krishnan. S (2011115507), Jayviz Selvin. X (2011115519) and Sanjith. J.K (2011115549)"** who carried out the project work under my supervision, for the partial fulfillment of the requirements for the award of the degree of the Bachelor of Technology in Information Technology. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion.

PLACE: Chennai

DATE:

Dr. V.UMA MAHESWARI

PROJECT GUIDE

PROFESSOR

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY

COLLEGE OF ENGINEERING

ANNA UNIVERSITY

CHENNAI – 600 025

COUNTERSIGNED

DR. A. KANNAN

HEAD OF THE DEPARTMENT

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY

COLLEGE OF ENGINEERING GUINDY

ANNA UNIVERSITY

ABSTRACT (ENGLISH)

Handwriting recognition is one of the important areas in image processing. Writer recognition proves to be more challenging because it involves identifying the writer, independent of the text written by him / her. Handwriting styles like slant, scale help to identify features that uniquely distinguish a writer. Text Independent Handwriting Recognition is widely used in the field of forensic analysis and document verification.

The Writer Identification System consists of training, enrollment and identification stages. In all the stages, the handwritten document images are segmented into word regions and Scale Invariant Feature Transform (SIFT) is applied and descriptors are obtained. In the Training stage, Codebook is constructed from the SIFT Descriptors(SD) of the training samples of different writers so that the number of features that need to be considered can be reduced for identification. In the Enrollment stage, the handwriting documents of different writers are given as input and two features SD Signature (SDS) and Scale and Orientation Histogram(SOH) are generated for each writer and stored as feature templates to be used in the next stage. In the Identification Stage, the SDS of the given handwritten document is generated using the Codebook and is compared with SDS present in the feature templates and a distance measure is computed. Likewise, the SOH of the given handwriting is generated, matched with those found in the feature templates and a distance measure is computed. The two measures are used in the identification of the writer.

திட்டப்பணிச் சுருக்கம்

கையெழுத்து அங்கீகாரம் பட செயலாக்க முக்கியமான பகுதிகளில் ஒன்றாகும். எழுத்தாளரின் சுதந்திரமான உரை அடையாளம் கான்பதால் எழுத்தாளர் அங்கீகாரம் சவால் ஆகும். கையெழுத்தின் சாய்வு, கையெழுத்தின் அளவுகள் தனிப்பட ஒரு எழுத்தாளர் வேறுபடுத்தி பண்புகளைக் கண்டறிய உதவுகிறது. கையெழுத்து அங்கீகாரம் தடயவியல் ஆய்வு மற்றும் ஆவண சரிபார்ப்பு துறையில் பரவலாக பயன்படுத்தப்படுகிறது.

எழுத்தாளர் அடையாளம் காணும் கருவியில் பயிற்சி, சேர்க்கை மற்றும் அடையாள நிலைகளில் உள்ளன. அனைத்து நிலைகளிலும், கையால் எழுதப்பட்ட ஆவணப் படங்களின் வார்த்தைப் பகுதிகளை பிரித்து, அளவில் மாற்றமில்லாத அம்சம் மாற்றம் (SIFT) பயன்படுத்தி டிஸ்க்ரிப்டர்களைப் பெற்று வருகின்றனர். பரிசீலிக்கப்பட வேண்டும் என்ற அம்சங்கள் பல அடையாளம் குறையக்கூடாது என்பதற்காக பயிற்சி கட்டத்தில் பல்வேறு எழுத்தாளர்கள் பயிற்சி மாதிரிகள் SIFT டிஸ்க்ரிப்டர்களைப் (SD) பயன்படுத்தி Codebook கட்டப்பட்டுகிறது. பதிவு கட்டத்தில், பல்வேறு எழுத்தாளர்கள் கையெழுத்து ஆவணங்களை உள்ளிட்டு இரண்டு அம்சங்கள் SDகையெழுத்து (SDS) மற்றும் அளவு மற்றும் திசை பட்டை வரைபடம் (SOH) ஒவ்வொரு எழுத்தாளர்க்கு உருவாக்கப்படுகின்றன மற்றும் அம்சம் வார்ப்புருக்கள் உருவாக்கப்பட்டு அடுத்த நிலைகளில் பயன்படுத்த சேமிக்கப்பட்டு உள்ளன. அடையாள நிலையில், கொடுக்கப்பட்ட கையால் எழுதப்பட்ட ஆவணம் மற்றும் Codebook கொண்டு SDS உருவாக்கி, அம்சம் வார்ப்புருக்களில் உள்ள SDS உடன் ஒப்பிட்டு வேறுபாடு கணிக்கப்படுகின்றன். அதேபோல், கொடுக்கப்பட்ட கையெழுத்தின் SOH உருவாக்கப்பட்டு அம்சம் வார்ப்புருக்களில் காணப்படும் SOH உடன் வேறுபாடு நடவடிக்கை கணிக்கப்படுகின்றன. இரண்டு வேறுபாட்டுகள் எழுத்தாளர் அடையாளத்திற்கு பயன்படுத்தப்படுகின்றன.

ACKNOWLEDGEMENT

We express our deep gratitude to our guide, **Dr. V. Uma Maheswari** for guiding us through every phase of the project. We appreciate her thoroughness, tolerance and ability to share her knowledge with us. We thank her for being easily approachable and quite thoughtful. Apart from adding her own input, she has encouraged us to think on our own and give form to our thoughts. We owe her for harnessing our potential and bringing out the best in us. Without her immense support through every step of the way, we could never have it to this extent.

We are extremely grateful to **Dr. A. Kannan**, Head of the Department of Information Science and Technology, Anna University, Chennai 600025, for extending the facilities of the Department towards our project and for his unstinting support.

We express our thanks to the panel of reviewers **Dr. S. Swamynathan**, Associate Professor, **Dr. K. Vidya** , Assistant Professor , **Dr. N. Thangaraj** , Assistant Professor, **Ms. K. Arul Deepa** , Assistant Professor for their valuable suggestions and critical reviews throughout the course of our project.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

Ashok Krishnan. S **Jayviz Selvin. X** **Sanjith. J.K**

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT (ENGLISH)	iii
	ABSTRACT (TAMIL)	iv
	LIST OF TABLES	x
	LIST OF FIGURES	xi
	LIST OF ABBREVIATIONS	xiii
1	**INTRODUCTION**	**1**
	1.1 HANDWRITING RECOGNITION	1
	1.1.1 Online Recognition	1
	1.1.2 Offline Recognition	3
	1.2 IMAGE PREPROCESSING	4
	1.2.1 Binarization	4
	1.2.2 Laplacian of Gaussian Filter	5
	1.2.3 Word Segmentation	5
	1.3 FEATURE EXTRACTION TECHNIQUES	6

	1.4 CLASSIFICATION	7
	1.4.1 Supervised learning	7
	1.4.1.1 Neural Networks	7
	1.4.2 Unsupervised learning	7
	1.4.2.1 Clustering	8
	1.4.2.2 Self Organizing Map	8
	1.5 ORGANIZATION OF THESIS	9
2	**RELATED WORKS**	**10**
	2.1 FEATURE EXTRACTION	10
	2.2 WORD SEGMENTATION	13
	2.3 LIMITATIONS IN THE PREVIOUS WORKS	14
	2.4 OBJECTIVES OF THE SYSTEM	14
3	**WRITER IDENTIFICATION SYSTEM**	**15**
	3.1 ARCHITECTURE	15
	3.2 WORD SEGMENTATION	18
	3.3 SCALE INVARIANT FEATURE TRANSFORM	20

	3.3.1	Scale Space Extrema detection	22
	3.3.2	Keypoint Localization	23
	3.3.3	Orientation Assignment	24
	3.3.4	Keypoint Descriptor	24
	3.4	CODEBOOK GENERATION	24
	3.5	FEATURE EXTRACTION	25
	3.5.1	SIFT Descriptor Signature	25
	3.5.2	Scale and Orientation Histogram	27
	3.6	FEATURE MATCHING	28
4		**IMPLEMENTATION**	**30**
	4.1	WORD SEGMENTATION	30
	4.2	FEATURE EXTRACTION	31
	4.2.1	SD Signature Algorithm	31
	4.2.2	Scale and Orientation Histogram Algorithm	32
	4.3	CODEBOOK GENERATION	33
	4.4	FEATURE MATCHING	33

5	**PERFORMANCE EVALUATION**		**35**
	5.1	RESULTS	35
	5.2	ANALYSIS	38
6	**CONCLUSION AND FUTURE WORK**		**42**
	6.1	CONCLUSION	42
	6.2	FUTURE WORK	42
	REFERENCES		**43**

LIST OF TABLES

TABLE NO.	TITLE	PAGE NO.
5.1	Time taken in Training Stage	40
5.2	Identifying handwriting using Different Approaches	41

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
1.1	Sample Handwritten Image	1
1.2	Online Recognition	3
1.3	Binary Image	4
1.4	Word Segmentation	5
3.1	Training Stage	15
3.2	Enrollment Stage	16
3.3	Identification Stage	17
3.4	Original Image	19
3.5	Binary Image	19
3.6	Filtered Image	19
3.7	Filtered Binary Image	20
3.8	Segmented Words	20

3.9	Steps in SIFT	21
3.10	Features Extracted by SIFT	21
3.11	Difference of Gaussian	22
3.12	Maxima and Minima of DoG	23
5.1	SOM Hits	35
5.2	Histogram of SDS Difference of Same Writer and Different Writer	36
5.3	SOH Difference of Same Writer	37
5.4	SOH Difference of Two Different Writers	37
5.5	Two Samples from the Same Writer	38
5.6	Two Samples from the Different Writer	39
5.7	Two Samples from the Same Writer with Different Slants	39
5.8	Two similar samples from the different writers	40

LIST OF ABBREVATIONS

HWR	Handwriting Recognition
PDA	Personal Digital Assistance
LoG	Laplacian of Gaussian
GLCM	Gray-Level Co-occurrence Matrix
ANN	Artificial Neural Network
SOM	Self Organizing Map
ART	Adaptive Resonance Theory
HMT	Hidden Markov Tree
LBP	Local Binary Pattern
FRG	Feature Relation Graph
PDF	Probability Density Function
SIFT	Scale Invariant Feature Transform
SD	SIFT Descriptors
SDS	SIFT Descriptor Signature

SOH	Scale and Orientation Histogram
WR	Word Regions
DoG	Difference of Gaussian

CHAPTER 1

INTRODUCTION

1.1 HANDWRITING RECOGNITION

Handwriting recognition (HWR) is the ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touch-screens and other devices. Recognition is online if the text is obtained using the movements of a pointing device on a Pen-based computer screen surface. Whereas the recognition is said to be offline if the written text image is obtained from a piece of paper by optical scanning. Handwriting recognition principally entails optical character recognition. A complete handwriting recognition system handles formatting, performs correct segmentation into characters and finds the most of the words. HWR is very important for forensic analysis, documents authorization, calligraphic relics identification Signature verification ,etc. Figure 1.1 shows a sample handwritten text.

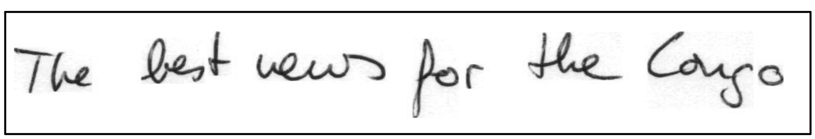


Figure 1.1 Sample Handwritten Image

1.1.1 Online Recognition

Online handwriting recognition involves the automatic conversion of text as it is written on a special digitizer or PDA, where a sensor picks up the pen-tip movements as well as pen-up/pen-down switching. The elements of an on-line handwriting recognition interface typically include:

- a pen or stylus for the user to write with.
- a touch sensitive surface, which may be integrated with, or adjacent to, an output display.
- a software application which interprets the movements of the stylus across the writing surface, translating the resulting strokes into digital text.

The process of online handwriting recognition can be broken down into a few general steps:

- preprocessing
- feature extraction and
- classification.

The purpose of preprocessing is to discard irrelevant information in the input data, that can negatively affect the recognition. This concerns speed and accuracy. Preprocessing usually consists of binarization, normalization, sampling, smoothing and denoising. The second step is feature extraction. Out of the two or more dimensional vector field received from the preprocessing algorithms, higher-dimensional data is extracted. The purpose of this step is to highlight important information for the recognition model. This data may include information like pen pressure, velocity or the changes of writing direction. The last big step is classification. In this step various models are used to map the extracted features to different classes and thus identifying the characters or words the features represent.

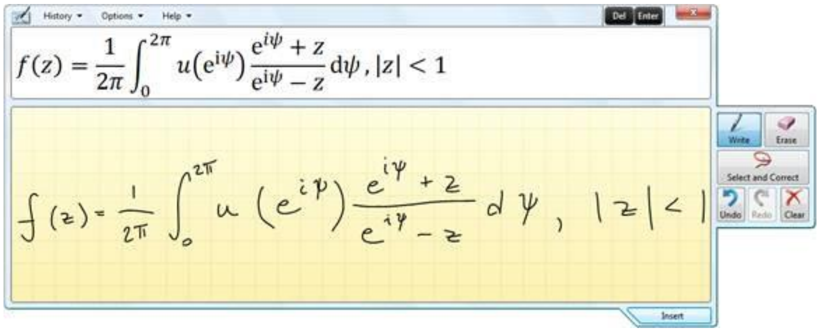


Figure 1.2 Online Recognition

1.1.2 Offline Recognition

Offline handwriting recognition involves the automatic conversion of text in an image into letter codes which are usable within computer and text-processing applications. The data obtained by this form is regarded as a static representation of handwriting. Offline handwriting recognition is comparatively difficult, as different people have different handwriting styles. Off-line character recognition often involves scanning a form or document written sometime in the past. This means the individual characters contained in the scanned image will need to be extracted. After the extraction of individual characters , a recognition engine is used to identify the corresponding computer character. Neural network recognizers learn from an initial image training set. The trained network then makes the character identifications. Each neural network uniquely learns the properties that differentiate training images. It then looks for similar properties in the target image to be identified. Some example properties are important in an handwriting are aspect ratio, number of strokes, average distance from image center. Off-line writer identification is divided into two types - text-dependent and text-independent writer identification. Text-dependent identification matches one or a small group of same characters/words and consequently requires the writer to write the same fixed text in the handwriting documents. Text-independent identification does not use the writing features of some specific characters/words, while instead considers handwriting document layout features, text line features, etc.

1.2 IMAGE PREPROCESSING

Preprocessing is the first and important step in processing a handwritten Image. The purpose of preprocessing is to remove irrelevant information in the input image. In preprocessing the input image is converted into binary image, then filtered using linear filters. Since the image can contain uneven illumination and image contrast variation, the image is binarized to produce good results. Uneven intensity values found in the images are removed using image filters.

1.2.1 Binarization

Image Binarization is usually performed as a first step in preprocessing stage of Image processing. It converts a gray-scale document image into a binary document image. A binary image is a digital image that has only two possible values for each pixel. Typically the two colors used for a binary image are black and white although any two colors can be used. The color used for the object in the image is the foreground color while the rest of the image is the background color. In the document-scanning industry this is often referred to as bi-tonal. Image Binarization converts an image of up to 256 gray levels to a black and white image. Frequently, Binarization is used as a pre-processor before OCR. A very important characteristic of a binary image is the distance transform. This gives the distance of every set pixel from the nearest unset pixel. The distance transform can be efficiently calculated. Figure 1.4 shows example of Binary Image.

Figure 1.3 Binary Image

1.2.2 Laplacian of Gaussian Filter

Handwritten images are often corrupted by random variations in intensity values called noise. One such noise is gaussian noise. Gaussian noise contains variations in intensity that are drawn from a Gaussian or normal distribution. The Laplacian of Gaussian filter (LoG) is used to remove gaussian noise. The Laplacian of Gaussuan essentially acts as a bandpass filter because of its differential and smoothing behavior. Second, the Gaussian is separable, which helps make computation very efficient. The Laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image. This means that in areas where the image has a constant intensity (i.e. where the intensity gradient is zero), the LoG response will be zero. The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection. The Laplacian is often applied to an image that has first been smoothed with something approximating a Gaussian smoothing filter in order to reduce its sensitivity to noise. The operator normally takes a single graylevel image as input and produces another graylevel image as output.

1.2.3 Word Segmentation

Word Segmentation is important step in handwriting recognition because only after the sentences the segmented into words they can be processed, features are extracted and then features can be matched. Word segmentation is the problem of dividing a sentence of written language into its component words. Hough Transform is used to segment word from the input image. Figure 1.2 shown segmented words.

Figure 1.4 Word Segmentation

1.3 FEATURE EXTRACTION TECHINQUES

In Image processing, feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative, non-redundant, facilitating the subsequent learning and generalization steps. Feature extraction is related to dimensionality reduction. The extracted features contain the relevant information from the input data, so that the desired task can be performed by using this reduced representation instead of the complete initial data. In handwriting recognition, features are extracted using texture-based or structure-based approaches. In this section we will discuss different feature extraction techniques.

A statistical method of examining texture that considers the spatial relationship of pixels is the gray-level co-occurrence matrix (GLCM), also known as the gray-level spatial dependence matrix. The GLCM functions characterize the texture of an image by calculating how often pairs of pixel with specific values and in a specified spatial relationship occur in an image, creating a GLCM, and then extracting statistical measures from this matrix.

Grid Microstructure Feature is a histogram based feature extraction technique and it is used in Chinese writer identification system. It is extracted from the edge image of the real handwriting image. Edge Image is seen as a primary representation of handwritten document. The positions of the edge pixel are used to describe the characteristics in a local grid around every edge pixel. After global statistic, the probability density distribution of different pixel pairs is regarded as the feature representing the writer style of the handwriting. Then the similarity of the two handwritings is measured with improved weighted versions of some original metric.

1.4 CLASSIFICATION

Classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. Learning techniques are classified into supervised and unsupervised learning.

1.4.1 Supervised learning

Supervised learning is the machine learning task of inferring a function from labeled training data. The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object and a desired output value. A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. Artificial Neural Network is an example of Supervised learning.

1.4.1.1 Neural Network

Artificial neural networks (ANNs) are learning algorithms used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. Artificial neural networks are generally presented as systems of interconnected neurons which can compute values from inputs, and are capable of machine learning as well as pattern recognition thanks to their adaptive nature. Neural networks are one technique which can be used for image recognition.

1.4.2 Unsupervised learning

The problem of unsupervised learning is that of trying to find hidden structure in unlabeled data. Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution. This distinguishes unsupervised learning from supervised learning and reinforcement learning. Approaches to unsupervised learning are clustering (e.g., k-means, mixture models, hierarchical clustering),Hidden Markov models. Among neural network models, the self-organizing map (SOM) and adaptive resonance theory (ART) are commonly used unsupervised learning algorithms.

1.4.2.1 Clustering

Clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). In Image processing, similar features extracted from the handwriting image form clusters. Clustering limits the features extracted from the handwriting image.

1.4.2.2 Self Organizing Map

A self-organizing map (SOM) or self-organizing feature map (SOFM) is a type of artificial neural network (ANN) that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map. Self-organizing maps are different from other artificial neural networks in the sense that they use a neighborhood function to preserve the topological properties of the input space.

A self-organizing map consists of components called nodes or neurons. Associated with each node are a weight vector of the same dimension as the input data vectors, and a position in the map space. The usual arrangement of nodes is a two-dimensional regular spacing in a hexagonal or rectangular grid. The self-organizing map describes a mapping from a higher-dimensional input space to a lower-dimensional map space. The procedure for placing a vector from data space onto the map is to find the node with the closest (smallest distance metric) weight vector to the data space vector. The SOM algorithm is based on unsupervised, competitive learning. It provides a topology preserving mapping from the high dimensional space to map units. Map units, or neurons, usually form a two-dimensional lattice and thus the mapping is a mapping from high dimensional space onto a plane. The property of topology preserving means that the mapping preserves the relative distance between the points. Points that are near each other in the input space are mapped to nearby map units in the SOM. The SOM can thus serve as a cluster analyzing tool of high-dimensional data. Also, the SOM has the capability to generalize. Generalization capability means that the network can recognize or characterize inputs it has never encountered before.

1.5 ORGANIZATION OF THESIS

In this report, the Chapter 2 gives a brief account of the related works which have been carried out in field of handwriting recognition. In Chapter 3 system architecture design of overall project is explained briefly. Chapter 4 deals with the implementation of the handwriting recognition system and are explained with pseudo codes. Chapter 5 consists of the experimentation and result analysis. Chapter 6 concludes the implemented project and explains the future enhancement for the proposed system.

CHAPTER 2

RELATED WORKS

2.1 FEATURE EXTRACTION

In this section, a general review of previous work on detecting features in handwriting image are discussed.

Offline Handwriting recognition is into text dependent and text independent. Text dependent offline deals with recognizing same text in same handwriting and in text independent approach the writer is recognized even with different text of same handwriting. The proposed work deals with offline text independent handwriting identification.

Since the text of handwriting images are different, text-level features like slant, scale are used to distinguish a writer. The existing approaches for offline text independent writer identification can be divided into texture-based and structure-based approaches. In textural-based approaches special textural features of handwriting texts are extracted and used for identification. Hanusiak et al [2] used a Gray-level occurrence matrix (GLCM) to extract the textural features from handwriting images. Y.Tang et al [3] extracted textural features based on Hidden Markov Tree (HMT) model in wavelet domain for writer identification. HMT model used one representative of the existing methods for off-line, text-independent writer identification, on both identification accuracy and computational efficiency. This accurate model naturally brings about a better identification result. Y.Tang et al [4] also extracted wavelet-based textural feature from handwriting images. It presents a simple and effective feature for off-line, text-independent writer identification, namely wavelet domain local binary patterns (WD-LBP). Based on WD-LBP, a writer identification algorithm is developed. In paper [5], Helli used Gabor and XGabor filter to extract features from handwriting Images and used a feature relation graph (FRG) to represent the extracted features. The extracted features are represented for each person by using a graph that is called FRG (feature relation graph). This graph is constructed using relations between extracted features by employing a fuzzy method. The fuzzy method determines the similarity between features extracted from different handwritten instances of each person.

A text-independent Persian writer identification based on feature relation graph (FRG). In paper [6], Bertolini et al used both Local binary patterns (LBP) and local phase quantization (LPQ) as textural features of handwriting identification. Besides assessing two texture descriptors (local binary patterns and local phase quantization), and also address important issues related to the dissimilarity representation, such as the impact of the number of references used for verification and identification, how the framework performs on the problem of writer identification, and how the dissimilarity-based approach compares to other feature-based strategies.

Structural-based features are found more stable and notable for writer Identification than textural-features of the handwriting image. Most of the structure-based techniques extract features from the contours of handwriting image. Bulacu et al [7] used edge-based directional distribution, edge-hinge distribution, directional co-occurrence PDF to distinguish a writer. A defining property of our methods is that they are designed to be independent of the textual content of the handwritten samples. These methods operate at two levels of analysis: the texture level and the character-shape (allograph) level. At the texture level, it uses contour-based joint directional PDFs that encode orientation and curvature information to give an intimate characterization of individual handwriting style. In the analysis at the allograph level, the writer is considered to be characterized by a stochastic pattern generator of ink-trace fragments, or graphemes. Maaten et al [8] used multi-scale edge-hinge feature for writer Identification. It concludes that multi-scale features may enhance identification performances and that random codebooks are to be preferred over Kohonen-based codebooks. In paper [9], Li et al proposed grid microscopic feature (GMF) by using edge pixel pairs for identification. The feature is extracted from the edge image of the real handwriting image. The positions of edge pixel pairs are used to describe the characteristics in a local grid around every edge pixel. Brink et al [10] used Quill-Hinge feature. The width of ink traces is a powerful source of information for off-line writer identification, particularly if combined with its direction. Such measurements can be computed using simple, fast and accurate methods based on pixel contours, the combination of which forms a powerful feature for writer identification. Schomaker et al [11] and Ghiasi et al [12] used coordinates of points on the normalized and resampled contours of connected components. In paper [12] and [13], features are extracted according to relationship of straight line segments that's fit the connected component of handwriting image. Djeddi et al [14] extracted the run-length features of the binary image of handwriting.

Most of the structural-based techniques are based on allograph fragment or contours of handwriting. Allograph fragments or contours are easily affected by Slant and aspect ratio of characters in handwriting. And these features fail to extract structural features between allograph of the same words which is major drawback of these methods. Structures between allograph in the same word have strong discriminability for different writers. To overcome these Scale Invariant Feature Transform [15] is used. This method extracts the keypoints based structural features at word level from handwriting images. This method is also insensitive to aspect ratio and slant of the characters.

2.2 WORD SEGMENATATION

Word segmentation is one of the important step in handwriting image preprocessing and it is mandatory that handwriting images are segmented into word level to extract the features of the handwriting.

In paper [16], text line segmentation is achieved by applying Hough transform on a subset of the document image connected components. The distances between adjacent overlapped components in a text line are calculated using the combination of two distance metrics and each of them is categorized either as an inter- or an intra-word distance in a Gaussian mixture modeling framework. The word segmentation procedure is divided into two steps. The first step deals with the computation of the distances of adjacent components in the text line image and the second step concerns the classification of the previously computed distances as either inter-word gaps or inter-character gaps. For the first step, the average of two different metrics: the Euclidean distance metric and the convex hull-based metric. The classification of the computed distances is performed using a well-known methodology from the area of unsupervised clustering techniques, the Gaussian mixtures.

In Paper [17], first text lines are segmented based on non-overlapped vertical zones and words are located by adopting SVM-based metric in each line. The line segmentation algorithm is based on locating the optimal succession of text and gap areas within vertical zones by applying Viterbi algorithm. Then, a text-line separator drawing technique is applied and finally the connected components are assigned to text lines. Word segmentation is based on a gap metric that exploits the objective function of a soft-margin linear SVM that separates successive connected components.

These text line segmentation methods fails to segment word on skew handwriting images in which text lines are not horizontal and hence isotropic LoG filter is used to segment words from handwriting images.

2.3 LIMITATIONS OF PREVIOUS WORKS

Experiments show that features like allograph fragments or contours are not identified. Allograph fragments or contours features are important in writer identification. They are easily affected by slant and aspect ratio of characters in handwriting and most of the structural-based techniques fail to extract structural features between allograph of the same words. Most of the word segmentation methods fail to segment word on skew handwriting.

2.4 OBJECTIVES OF THE SYSTEM

The system proposed by Xiangqian et al [1] uses Scale Invariant Feature technique (SIFT) to extract features from handwriting images. SIFT is used because it is insensitive to slant and aspect ratio. Text Independent Writer Recognition System is composed of training, enrollment and identification stages. In all stages the images are segmented into word regions and Scale Invariant Feature Transform (SIFT) is applied and descriptors are obtained. In Training stage Codebook is constructed from SIFT Descriptors(SD)s of training samples of different writers. In Enrollment stage ,the handwriting of different writer are enrolled and two features SD Signature (SDS) and Scale and Orientation Histogram (SOH) are generated and stored as feature templates for Identification. In Identification Stage the SDS and SOH of the input handwriting are matched with feature templates based on nearest match is found.

CHAPTER 3

WRITER IDENTIFICATION SYSTEM

3.1 ARCHITECTURE

This project discusses and implements the paper [1] used for identifying text independent offline handwriting.

Overall architecture is divided into three stages. They are training stage, enrollment stage, identification stage. The different stages are discussed below:

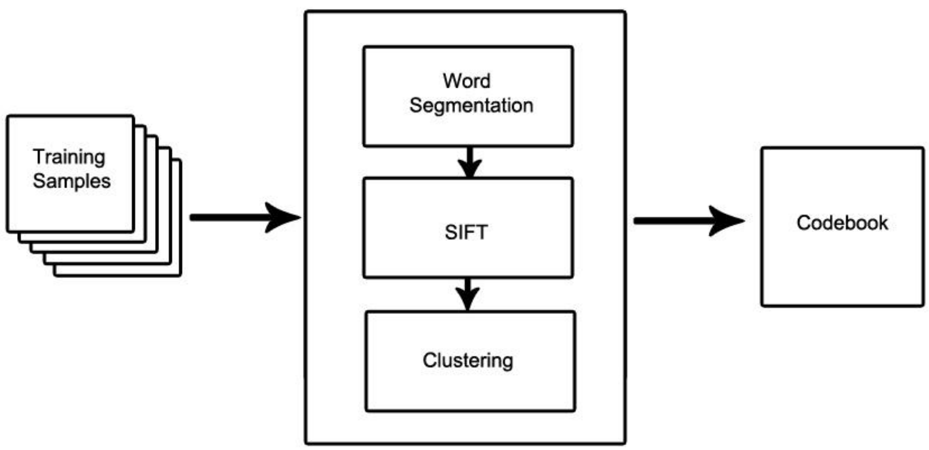


Figure 3.1 Training Stage

Training Stage :

In the training stage, the training sample are segmented into word regions. For each word region, SIFT is used to detect the number of keypoints and extract the descriptors, scales and orientations. Since there are more number of keypoints and descriptors, it is very difficult to keep all the descriptors for identification. In order to increase efficiency, the SDs of the keypoints are clustered to form Codebook. Each value in the Codebook represents a descriptor in the center of each cluster. The hierarchical Kohonen SOM Clustering is used for Codebook generation.

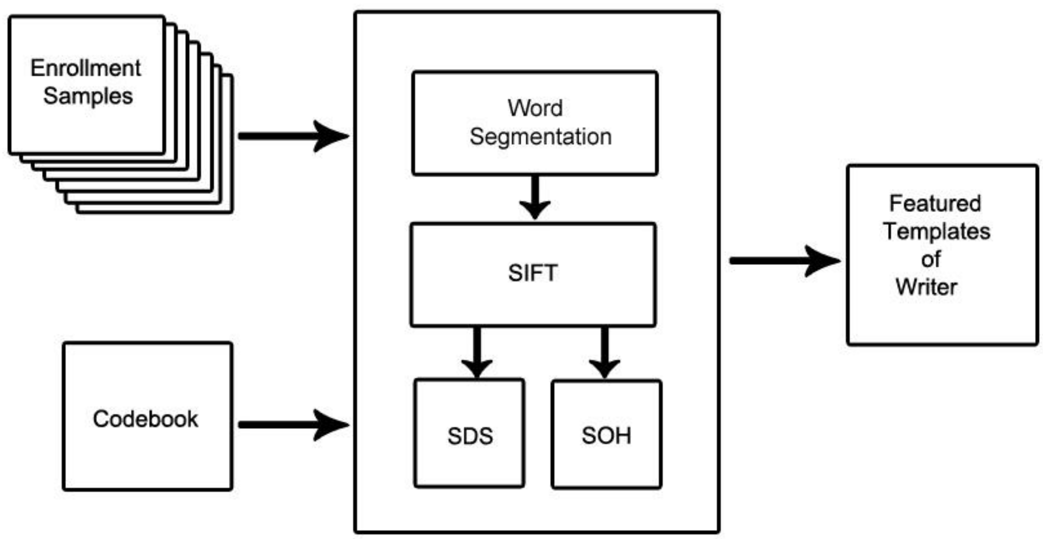


Figure 3.2 Enrollment Stage

Enrollment Stage :

In the Enrollment Stage, handwriting image of different writers to be enrolled are segmented into word regions. Features (descriptors) of each word region are extracted by applying SIFT. Using the extracted Codebook and descriptors, SIFT Descriptor Signature (SDS) and Scale-Orientation Histogram (SOH) for each writer are generated and are stored as feature templates to be used in identification stage.

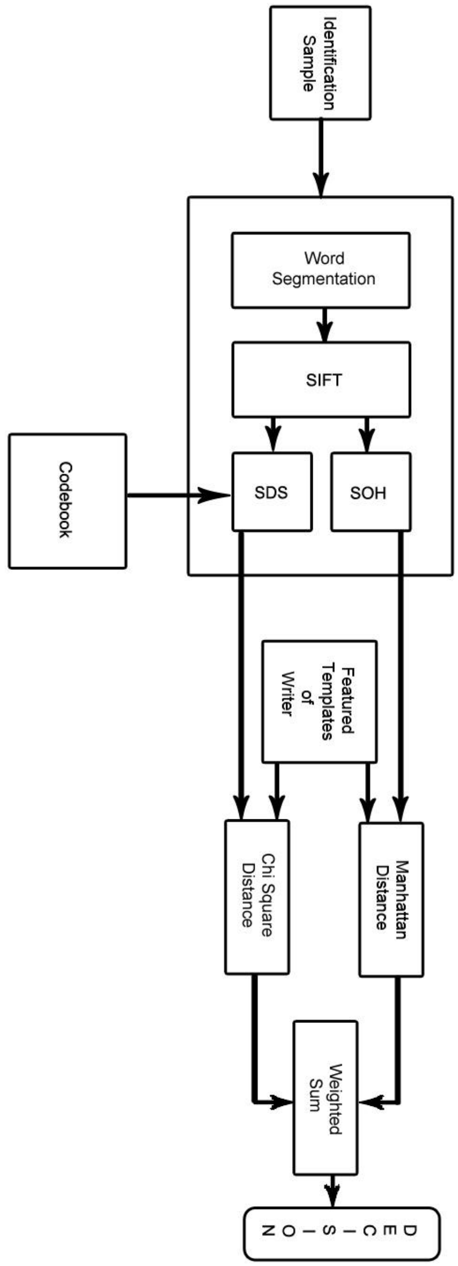


Figure 3.3 Identification Stage

Identification Stage :

In the Identification Stage, the SDS and SOH of the given input handwriting image are matched with feature templates and the writer is matched based on nearest match.

3.2 WORD SEGMENTATION

In order to extract the word-level features of handwriting image, given handwriting image is segmented into word regions (WRs). An isotropic LoG filter is used to avoid text line segmentation and to reduce the effect of the direction of text lines. Following are the steps to segment words from handwriting images.

Step 1:

Given input handwriting image is converted to binary image (I_b) by using Otsu algorithm. Otsu's method is used to automatically perform the reduction of a graylevel image to a binary image. Binary image formed consists of 1's and 0's..

Step 2:

Connected components of the binary image(I_b) are found and then average height (h_a) of the connected components is calculated.

Step 3:

Binary image(I_b) is filtered using Laplacian of Gaussian (LoG) filter to get filtered image (I_f) . Since the space between same word are much less than the space between between different words LoG filter is suitable to concatenate connected components of same words and separate different words. Variance of the LoG filter is taken as 2.5 x h_a .

Step 4:

The filtered image (I_f) is binarized using threshold obtained by Otsu algorithm used in step 1 to get filtered binary image (I_{fb}).

Step 5:

Each connected component in I_b is assigned to nearest connected component in I_{fb} to form semi word regions (SWR) .

Step 6:

SWR are merged to get Word Regions according to the adjacent distances.

Step7:

Overlapping connected components are splitted into separate connected components.

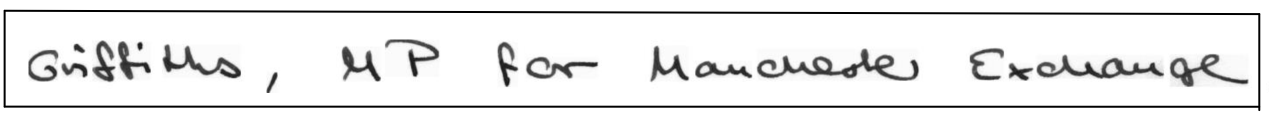


Figure 3.4 Original Image

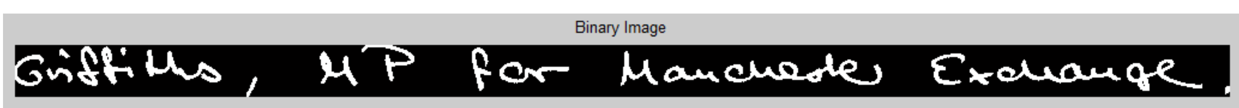


Figure 3.5 Binary Image

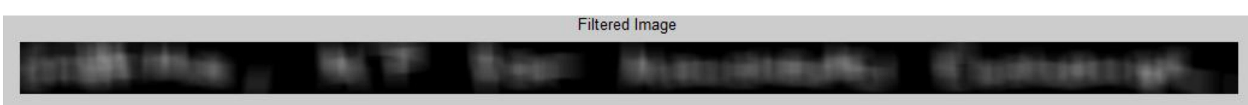


Figure 3.6 Filtered Image

Figure 3.7 Filtered Binary Image

Figure 3.8 Segmented Words

Once all the steps are done word segments are used in extracting features. Feature extraction is performed using SIFT technique.

3.3 SCALE INVARIANT FEATURE TRANSFORM

Scale invariant feature transform (SIFT) for distinctive scale-invariant features extraction from images, has been widely and successfully applied in many fields.

The SIFT algorithm has four major stages of computation:

(1) scale-space construction,

(2) key point localization,

(3) orientation assignment,

(4) key point descriptor extraction.

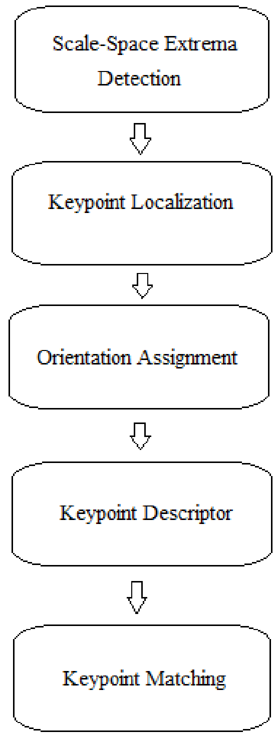


Figure 3.9 Steps in SIFT

Keypoint descriptors typically uses a set of 16 histograms, aligned in a 4x4 grid, each with 8 orientation bins, one for each of the main compass directions and one for each of the mid-points of these directions. This results in a feature vector containing 128 elements.

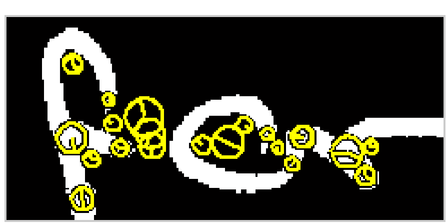


Figure 3.10 Features Extracted by SIFT

3.3.1 Scale Space Extrema Detection

In Scale Space Extrema Detection, Laplacian of Gaussian is found for the image with various values. LoG acts as a blob detector which detects blobs in various sizes due to change in. In short, acts as a scaling parameter. For eg, in the above image, gaussian kernel with low gives high value for small corner while guassian kernel with high fits well for larger corner. So, we can find the local maxima across the scale and space which gives us a list of values which means there is a potential keypoint at (x,y) at scale. But this LoG is a little costly, so SIFT algorithm uses Difference of Gaussians which is an approximation of LoG. Difference of Gaussian is obtained as the difference of Gaussian blurring of an image with two different. This process is done for different octaves of the image in Gaussian Pyramid. It is represented in image below:

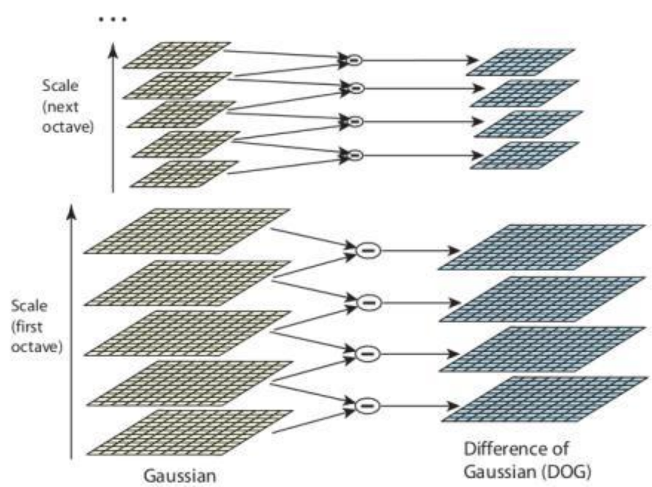


Figure 3.11 Difference of Gaussian

Once this DoG are found, images are searched for local extrema over scale and space. For eg, one pixel in an image is compared with its 8 neighbours as well as 9 pixels in next scale and 9 pixels in previous scales. If it is a local extrema, it is a potential keypoint. It basically means that keypoint is best represented in that scale. It is shown in image below:

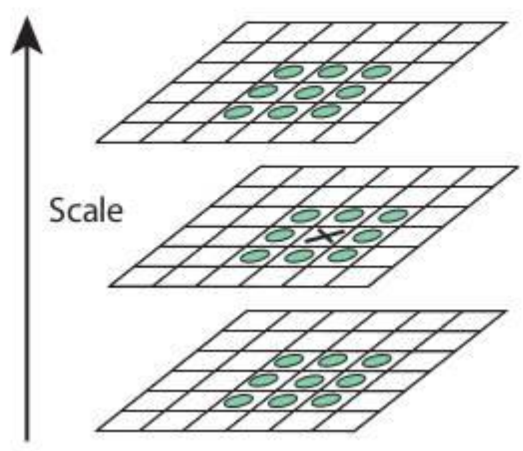


Figure 3.12 Maxima and Minima of DoG

3.3.2 Keypoint Localization

Once potential keypoints locations are found, they have to be refined to get more accurate results. They used Taylor series expansion of scale space to get more accurate location of extrema, and if the intensity at this extrema is less than a threshold value (0.03 as per the paper), it is rejected. This threshold is called contrastThreshold.

DoG has higher response for edges, so edges also need to be removed. For this, a concept similar to Harris corner detector is used. They used a 2x2 Hessian matrix (H) to compute the pricipal curvature.

So it eliminates any low-contrast keypoints and edge keypoints and what remains is strong interest points.

3.3.3 Orientation Assignment

Now an orientation is assigned to each keypoint to achieve invariance to image rotation. A neigbourhood is taken around the keypoint location depending on the scale, and the gradient magnitude and direction is calculated in that region. An orientation histogram with 36 bins covering 360 degrees is created. (It is weighted by gradient magnitude and gaussian-weighted circular window with \sigma equal to 1.5 times the scale of keypoint. The highest peak in the histogram is taken and any peak above 80% of it is also considered to calculate the orientation. It creates keypoints with same location and scale, but different directions. It contribute to stability of matching.

3.3.4 Keypoint Descriptor

Now keypoint descriptor is created. A 16x16 neighbourhood around the keypoint is taken. It is divided into 16 sub-blocks of 4x4 size. For each sub-block, 8 bin orientation histogram is created. So a total of 128 bin values are available. It is represented as a vector to form keypoint descriptor. In addition to this, several measures are taken to achieve robustness against illumination changes, rotation etc.

In this work we use SIFT to extract descriptors of keypoints identified in handwriting image to generate SDS and SOH features. Before generating these features the codebook is constructed. Codebook is generated by clustering the descriptors of training samples. Codebook generation is discussed in next section. Two features SDS and SOH features are used to uniquely distinguish a writer.

3.4 CODEBOOK GENERATION

Handwriting image is segmented into word regions. For every word region descriptors of keypoints are extracted using Scale Invariant Feature Transform technique. Extracted descriptors of keypoints are large in number. Number of descriptors vary depending on the number of words and number of letters in each word, in the given samples. So the dimensionality is too large and it is infeasible to use all of them. Hence clustering technique is used to bring together features that are very similar. SDs of keypoints found in training samples are clustered into N categories and each category is represented by its center, called as code. All of the N codes form a codebook of size N. And using codebook, SDS features are computed that help in the identification of the writer.

The Hierarchical Kohonen Self Organizing Map (SOM) proposed in paper [1] performs better clustering compared to other clustering techniques. Hence it is used in this work for clustering. For every cluster generated, the center of each cluster is taken and Codebook is generated. The size of Codebook is empirically chosen to be 300.

3.5 FEATURE EXTRACTION

Since the system deals with text independent handwriting recognition text in the enrolled handwriting document in offline text-independent writer identification system, the layout of the key points may be totally different in the different handwriting images, even if they are from same writer, so position of descriptors are of no use here. So the frequency of each SD and SO occurrences in a handwriting image is taken into account. Two algorithms proposed in paper [1] are used to generate two features SDS and SOH. Features SDS and SOH have strong influence for identification of writer.

3.5.1 SIFT Descriptor Signature

In this algorithm, Euclidean distance is calculated for every descriptor of handwriting image I with every values in Codebook C with respect to 128 dimensions. Then the distances are sorted in ascending order and their top t index values are stored. The SDS vector of those indexes are incremented with constant value. The process is repeated until all the descriptors of handwriting image I are processed. Then SDS vector is updated by dividing their value by SDS sum.

SD = $\{d_1, d_2, \ldots, d_n\}$ denotes the SDs extracted from an offline handwriting image I, where n is the size of SD vector. And $C = \{c_1, c_2, \ldots, c_N\}$ denote a codebook of size N.

The SD Signature algorithm is presented below:

1. SDS vector of size N is initialized with zeros.

$$SDS=(0 , 0. . , 0) \tag{3.1}$$

2. Using each descriptor $d_i \in SD$, Euclidean distance is computed with every code word $c_j \in C$ and stored in new EDV vector.

$$ED_{ij} = \sqrt{\sum_{K=1}^{L} (d_{ik} - c_{ij})} \tag{3.2}$$

where L=128 size of SIFT descriptor. and Euclidean distance vector is obtained is shown below

$$EDV = (ED_{i1}, ED_{i2},..., ED_{iN}) \tag{3.3}$$

4. EDV vector is then sorted in ascending order and top t index are stored in IDX vector

$$IDX=\{ idx_1 , idx_2 , \ldots , idx_t \} \tag{3.4}$$

5. SDS vector is added with non-increasing function δ(x) for each values in IDX vector where idx ∈ IDX

$$SDS_{idx} = SDS_{idx} +\delta(EDV_{idx}) \tag{3.5}$$

6. Steps 2 to 4 are repeated until all SDs are processed.

7. Values in SDS vector is divided by their total sum.

$$SDS_i = \frac{SDS_i}{\sum_{j=1}^{N} SDS_j} \tag{3.6}$$

3.5.2 Scale and Orientation Histogram

The images are decomposed into *X* octaves and *Y* sub-levels in each octave, i.e. Z(X × Y) scales, by using SIFT. Let S = { s_1, s_2, . . . , s_n} denote *n* SIFT keypoints scales, 1 ≤ s_i ≤ *Z*, and let O = { o_1, o_2, . . . , o_n} denote the corresponding orientations of these SIFT Keypoints Give an angle step φ, the orientation [0, 360] can be quantized to Obin = $\left\lceil \frac{360}{\varphi} \right\rceil$ intervals, where ⌈x⌉ is an operator to get the nearest integer which is greater than or equal to x.

The SOH feature generation algorithm is presented below.

1. SOH vector of size M = Z× O_{bin} is initialized with zeros.

$$SOH = (0, 0, . . , 0) \tag{3.7}$$

2. Index idx is calculated for each key point's scale and orientation, where $s_i \in S$ and $o_i \in O$.

And bin is calculated by taking the ceil of o_i by φ .

$$bin = \left\lceil \frac{o_i}{\varphi} \right\rceil \tag{3.8}$$

$$idx = O_{bin} \times (s_i - 1) + bin \tag{3.9}$$

3. SOH vector is incremented by 1 in the index idx .

$$SOH_{idx} = SOH_{idx} + 1 \tag{3.10}$$

4. Steps 2 and 3 are repeated until all key points are processed.

5. Every values in SOH vector is divided by their total sum.

$$SOH_i = \frac{SOH_i}{\sum_{j=1}^{n} SOH_j} \tag{3.11}$$

where n is the size of the SOH vector.
The parameter *X* and *Y* are empirically selected as 8 and 3.

3.6 FEATURE MATCHING

Once SDS and SOH are calculated feature matching is done based on the weighted sum. Let I_1 and I_2 denote two handwriting images, and let u = ($u_1, u_2,.., u_N$) and v = ($v_1, v_2,.., v_N$) denote their SDSs, and x = ($x_1, x_2,.., x_M$) and y = ($y_1, y_2,.., y_M$) denote their SOHs of two images.

The Manhattan distance is used to find the dissimilarity between two SDSs u and v:

$$D1(u, v) = \sum_{i=1}^{N} | u_i - v_i | \tag{3.12}$$

The Chi-Square distance is adopted to find the dissimilarity between two SOHs x and y

$$D2(x, y) = \sum_{j=1}^{M} (x_j + y_j)^2 \; / \; (x_j - y_j) \tag{3.13}$$

Distance D1 and D2 are fused to form a new distance to measure the dissimilarity between I_1 and I_2:

$$D(I_1, I_2) = w \times D1\,(u, v) + (1 - w) \times D2\,(x, y) \qquad (3.14)$$

where $0 \leq w \leq 1$ is a weight.

Input writer is matched with enrolled image based on the nearest match found.

CHAPTER 4

IMPLEMENTATION

4.1 WORD SEGMENTATION

Handwriting images are segmented into word regions before applying Scale Invariant Feature Technique. In the process of word segmentation the handwriting image is converted to binary image and then filtered with LoG filter and connected components are identified and labeled and they are segmented. Pseudo code for word segmentation is shown below.

Word Segmentation Algorithm

INPUT : Handwrting Image

OUTPUT **:** Segmented Words

Load the Handwriting Image I using imread

Convert Color Image I to Grayscale Image I using rgb2gray

Calculate Threshold t of Image is found using Otsu algorithm

Convert Grayscale image I is to Binary Image I_b using im2bw

Find all the Connected Components in I_b cc using bwconncomp

Initialize h to zero

for each connected component in I_b

    h = h + the height of each component

end for

Calculate the average height h_a

Filter (I_f) Image I is LoG filter

I_f is Binarized to I_{fb} using Threshold t

for each connected component c in I_b

Assign c to nearest connected component in I_{fb}

end for

4.2 FEATURE EXTRACTION

After finding the SIFT descriptors the two main features SD Signature and SO Histogram are extracted and used for identification.

4.2.1 SD Signature Algorithm

INPUT **:** SIFT Descriptors , Codebook

OUTPUT **:** SIFT Descriptor Signature

Intialize SDS vector with zeros

for each descriptor d

Initialize EDV vector with zeros

for each code c_i in Codebook

Initialize sum=0

for each dimension d in code c_i

if d > c

sum=sum+(d-c)^2

else

sum=sum+(c-d)^2

end if

end for

EDV(i) = sqrt(sum)

end for

EDV vector are sorted and top t index are stored in IDX

SDS vector is added with constant for top t indexes
end for
Initialize SDS_{sum}=0
for each SDS value SDS_v
$SDS_{sum} = SDS_{sum} + SDS_v$
end for
for each SDS value SDS_v
$SDS_v = SDS_v / SDS_{sum}$
end for

4.2.2 SD Scale and Orientation Histogram Algorithm

INPUT : SIFT Descriptor scales and Orientations

OUTPUT : SOH vector

Initialize x=8, y=3, phi=36, o_{bin}=10, M=240,pi=3.14
Initialize SOH vector with zeros
for each Orientation o
o=o*(180/pi);
end for
Initialize idx=0
for each Orientation o and Scale s
bin=ceil(o/phi)
idx=ceil(o_{bin}*(s-1)+bin)
SOH(idx)=SOH(idx)+1
end for
Initialize SOH_{sum}=0

for each SOH value SOH_v

    $SOH_{sum} = SOH_{sum} + SOH_v$

end for

for each SOH value SOH_v

    $SOH_v = SOH_v / SOH_{sum}$

end for

4.3 CODEBOOK GENERATION

SIFT Descriptors of training samples are clustered into categories and each category with its center is called a code. All the codes form SD Codebook. Pseudo code for codebook generation is shown below :

Codebook Generation

INPUT : SIFT descriptors of Training Samples

OUTPUT : Clustered SIFT Descriptor

Initialize cArr with Training Sample SDs

Create Self organizing map using selforgmap function

Train descriptors using train function

4.4 FEATURE MATCHING

Difference of SDS and SOH of two handwriting images are found. Then weighted sum is calculated using the difference of SDS and SOH. Using weighted sum the input image is matched with enrolled image. Pseudo code for feature matching is given below.

Feature Matching Algorithm

INPUT **:** SDS and SOH of input and enrolled images

OUTPUT **:** Dissimilarity distance

1 : Initialize iSDS vector and iSOH vector with SDS and SOH of input image

2 : Initialize eSDS vector and eSOH vector represents SDS and SOH of enrolled image.

3 : Initialize D1=0

4 : Initialize D2=0

5 : **for** each iSDS i and eSDS e

6 :   D1=D1+(i-e)

7 : **end for**

8 : **for** each iSOH i and eSOH e

9 :   D2=D2+((i-e)^2)/(i+e)

10 : **end for**

11 : Calculate dissimilarity distance **D=(0.6*(D1))+(0.5*(D2))**

CHAPTER 5

PERFORMANCE EVALUATION

5.1 RESULTS

Various Experiments are conducted to evaluate the performance of the writer identification system. Handwriting images used in the experiments are part of the IAM Dataset [18]. Different handwriting images are used in training, enrollment and identification stages. The system is tested for identifying a writer among the enrolled writers independent of the written text.

In the training stage, handwriting samples from different writer are used. From each sample approximately minimum of 1500 descriptors are generated and every descriptors all from training samples are clustered. Since the number of total descriptors are more, they are clustered in 300 clusters and centers of each clusters are used for identification of writer. Figure 5.1 shown below are SOM hits in 50 clusters used in smaller training set.

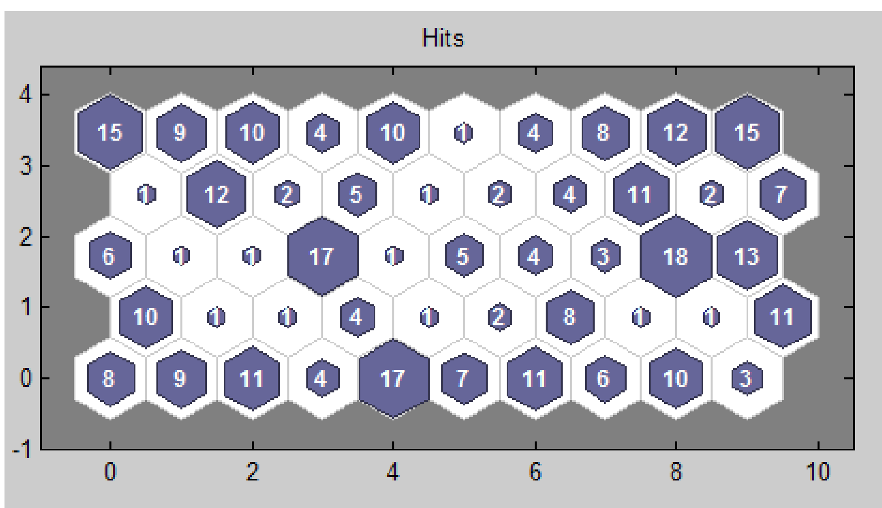
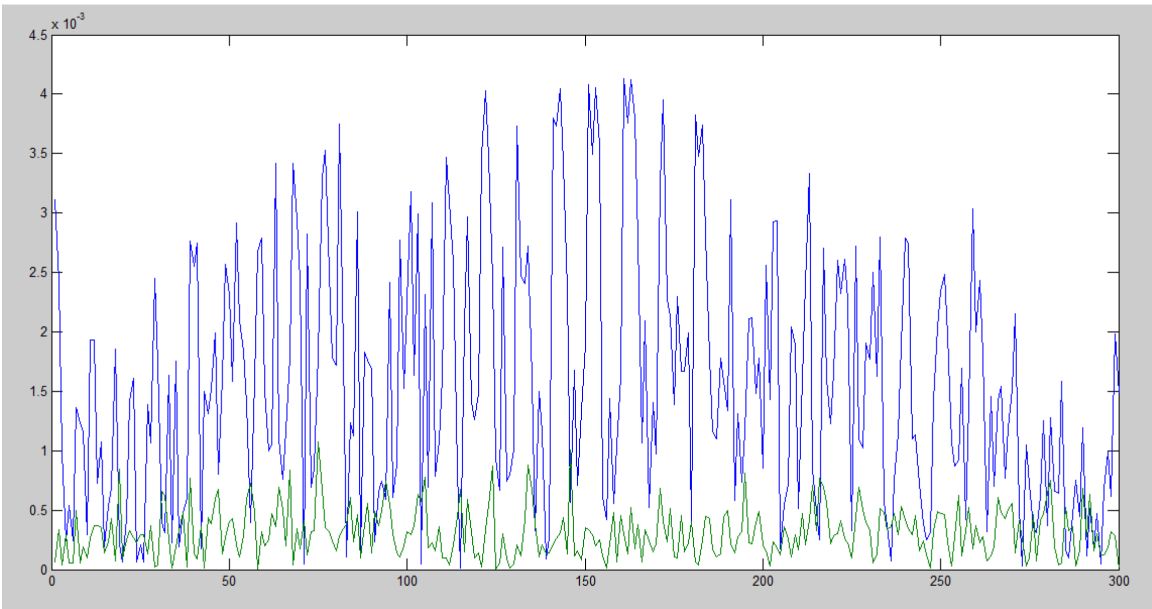


Figure 5.1 SOM Hits

In the enrollment and the identification stage features SDS and SOH are generated. Below figure shows difference of SDS between two samples from same writer and two samples from different writes. SDS difference of same writer shows smaller difference while there is a large difference between different writers.

Absolute Difference



Index of SDS Component

Figure 5.2 Histogram of SDS Difference of Same Writer and Different Writer

Figure 5.3 and Figure 5.4 shows the SOH difference of two same handwriting and different handwriting samples respectively. Difference for same writer vary from 0 to 0.6 x 10 $^{-4}$ and for different writer it varies from 0 to 5 x 10 $^{-4}$.

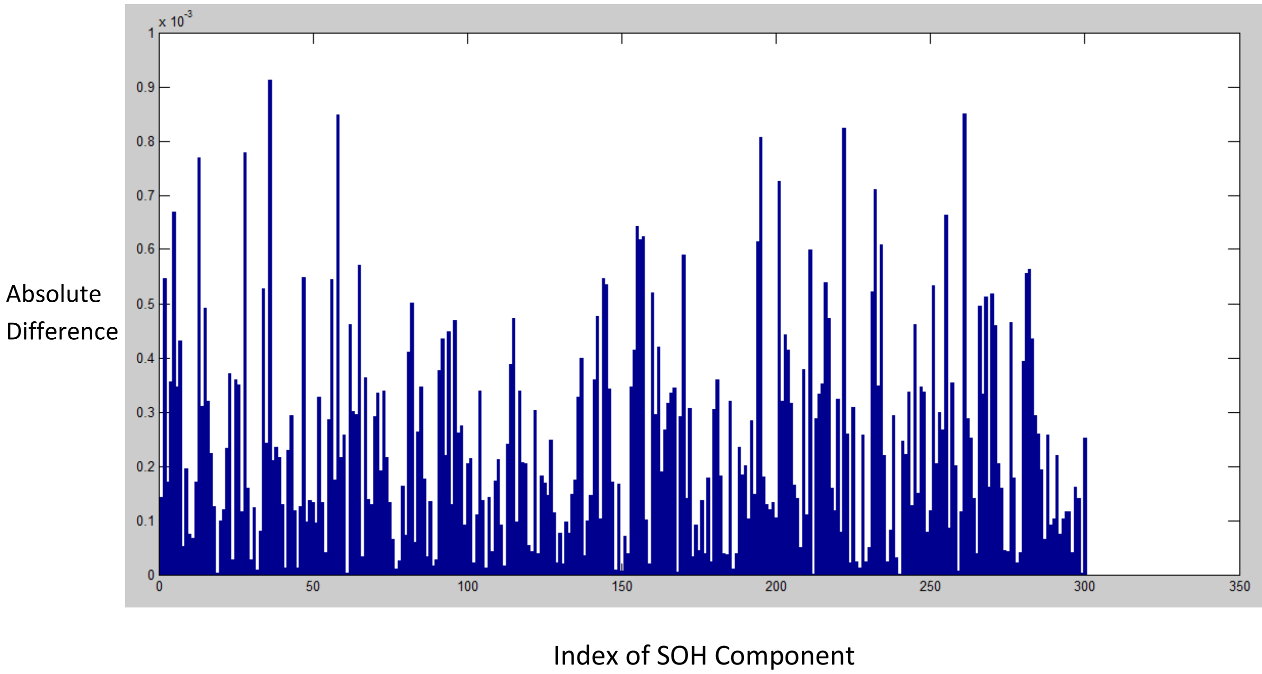


Figure 5.3 SOH Difference of Same Writer

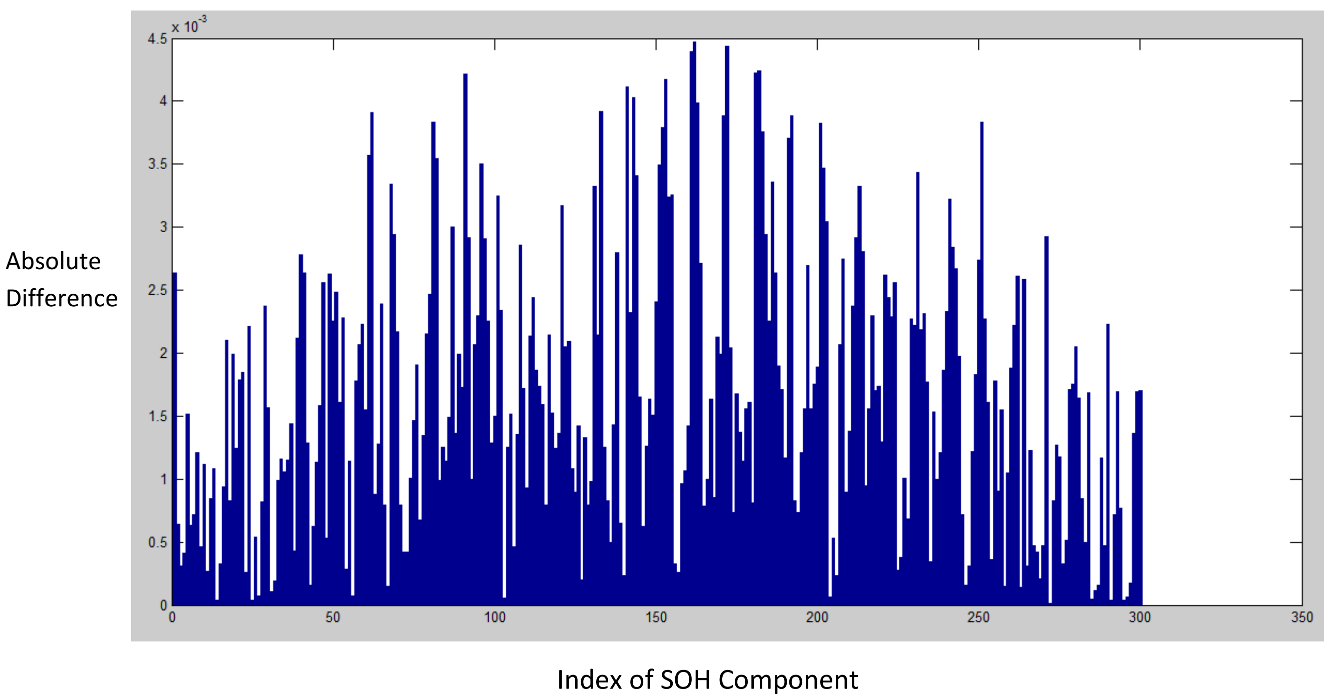


Figure 5.4 SOH Difference of two Different Writers

The SDS feature identifies better than the SOH feature. In order to increase the accuracy, both SDS + SOH are used in identification.

After computing SDS and SOH of input and enrollment images, SDS and SOH differences are calculated. Using the SDS and SOH difference dissimilarity between handwriting images are found. Those difference are added and handwriting of least sum is are found to be matched.

5.2 ANALYSIS

Two features SDS and SOH strongly distinguishes between writers and it is also possible to identify writer using either SDS or SOH. But to increase the accuracy in identifying the writer we use both SDS + SOH.

Figure 5.5 below shows the two samples of same writer and they are identified similar by the system.

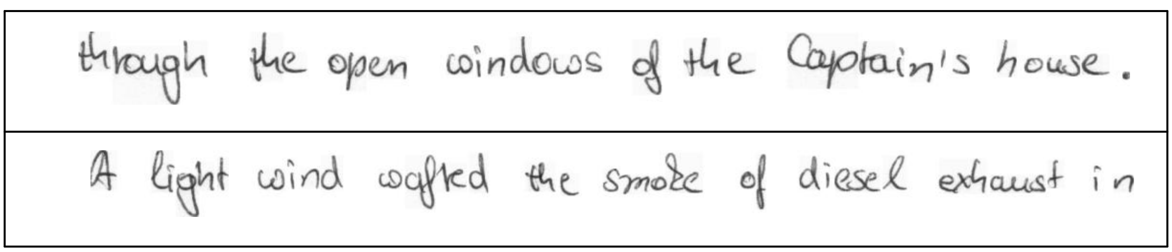


Figure 5.5 Two Samples from the Same Writer

Figure 5.6 below shows the two samples of different writer and they are identified samples of different writers.

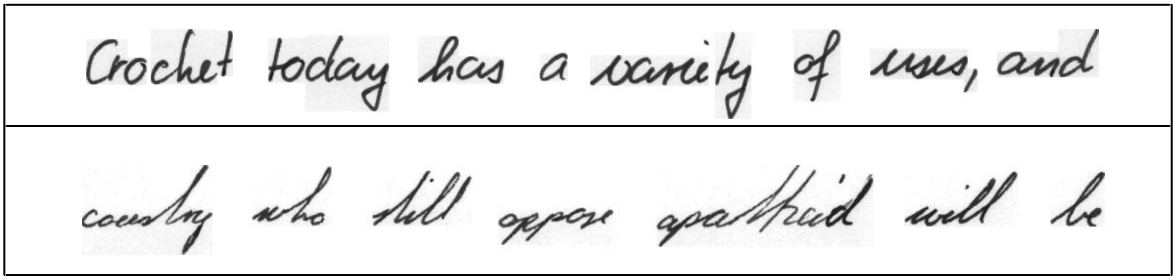


Figure 5.6 Two Samples from the Different Writer

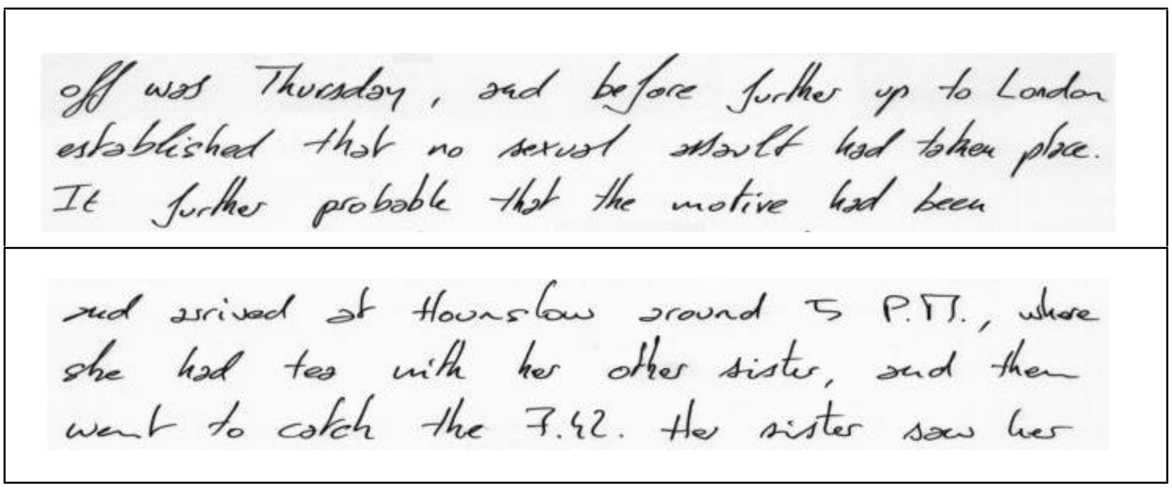


Figure 5.7 Two Samples from the Same Writer with Different Slants

In the figure 5.8 the top image is more slant than the other. It is identified as same handwriting by the proposed method, but incorrectly determined by the other feature based approaches such as Contour-hinge, GMF since there is no slant normalization, which make them insensitive to slant of the handwritings.

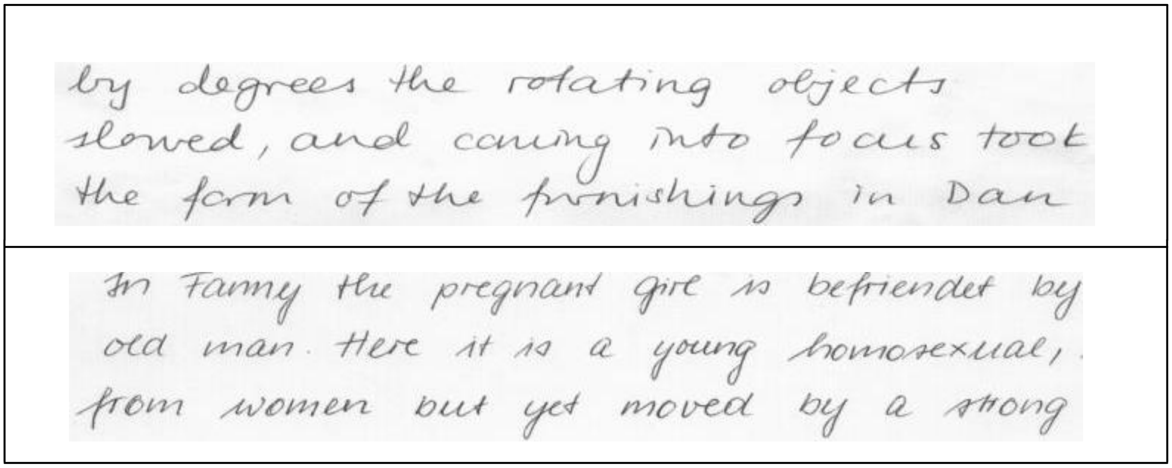


Figure 5.8 Two Similar Samples from the Different Writers

Figure 5.8 correctly identifies that they are from two different writers and incorrectly identified by other approaches. The two handwriting images are similar in slant, shape and orientation but the allograph of second image is compact that the first image hence incorrectly identified by other approaches but correctly identified by proposed system.

Five different writers are used in experiments. The system is experimented with different number of samples for each writer in training stage. Sets I, II, III represent 4,3,2 samples of each writer used in training stage.

Table 5.1 Time taken in Training Stage

SET	Number of descriptors	Training time (in Seconds)
I	5390	759
II	4129	670
III	2593	341

From table 5.1, in training stage the number of descriptors increases with training time. Increase in number of descriptors increases accuracy in identifying the writer. After training stage only 300 descriptors are used for identification.

Table 5.2 Identifying Handwriting using Different Approaches

SET	Samples identified using SDS + SOH	Samples identified using SDS	Samples identified using SOH
I	5	4	2
II	4	3	2
III	3	2	2

Different approaches used in identifying a writer are experimented with different training sets. Samples identified out of 5 enrolled writers are tabulated. Handwriting identified using SDS + SOH shows better accuracy than using SDS and SOH.

SDS and SOH differences are fused to form a new distance to measure the dissimilarity between two images I_1 and I_2:

$$D(I_1, I_2) = w \times D1\ (u, v) + (1 - w) \times D2\ (x, y) \tag{5.1}$$

where $0 \leq w \leq 1$ is a weight, D1 denotes SDS difference of two images and D2 denotes SOH difference of two images. Experiments are tried with different weight values and chosen as 0.6.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

In Writer Identification system, the keypoints found in the words of the handwriting image are extracted using SIFT. SIFT is used since it is insensitive to aspect ratio and slant of the characters and gives better results when compared to existing structure based approaches. The Codebook is created by clustering the descriptors extracted using SIFT. After codebook creation, two features SDS and SOH are generated. These features are used to characterize the writers individuality. The experiments are done in English IAM datasets. The proposed system is language-insensitive and can work well on different languages.

6.2 FUTURE WORK

Experiments show that SDS and SOH generated from handwriting image yield better result in identifying a writer compared to other features. As discussed in the previous chapter, results were found to be poor when using SOH feature for identification individually. The future work can be focused on improving the accuracy of SOH feature. So that when combining both SDS and SOH features the overall result will be improved. This shall be done by making use of another codebook, clustering both scale and orientation. This can be used in same way as SDS algorithm. Codebook can be generated using different clustering technique and the results can be analyzed.

REFERENCES

[1] Xiangqian Wu, Youbao Tang, and Wei Bu, "Offline Text-Independent Writer Identification Based on Scale Invariant Feature Transform," IEEE Transactions on Information Forensics and Security, vol. 9, no. 3,pp. 526-536, March 2014.

[2] R. Hanusiak, L. Oliveira, E. Justino, and R. Sabourin, "Writer verification using texture-based features," International Journal on Document Analysis and Recognition, vol. 15, no. 3, pp. 213–226, Sep. 2012.

[3] Z. He, X. You, and Y. Tang, "Writer identification of Chinese handwriting documents using hidden Markov tree model," Pattern Recognition, vol. 41, no. 4, pp. 1295–1307, Apr. 2008.

[4] L. Du, X. You, H. Xu, Z. Gao, and Y. Tang, "Wavelet domain local binary pattern features for writer identification," in Proc. 20th International Conference on Pattern Recognition, Istanbul, Turkey, pp. 3691–3694, Aug. 2010.

[5] B. Helli and M. Moghaddam, "A text-independent Persian writer identification based on feature relation graph (FRG)," Pattern Recognition, vol. 43, no. 6, pp. 2199–2209, Jun. 2010.

[6] D. Bertolini, L. Oliveira, E. Justino, and R. Sabourin, "Texture-based descriptors for writer identification and verification," Expert Systems with Applications, vol. 40, no. 6, pp. 2069–2080, May 2013.

[7] M. Bulacu and L. Schomaker, "Text-independent writer identification and verification using textural and allographic features," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 4, pp. 701–717, Apr. 2007.

[8] L. Maaten and E. Postma, “Improving automatic writer identification,” in Proceedings of the 17th Belgium-Netherlands Conference on Artificial Intelligence, Brussels, Belgium, pp. 260–266, 2005.

[9] X. Li and X. Ding, “Writer identification of Chinese handwriting using grid microstructure feature,” in Proceedings of the 3rd ICB, Alghero, Italy, pp. 1230–1239, 2009.

[10] A. Brink, J. Smit, M. Bulacu, and L. Schomaker, “Writer identification using directional ink-trace width measurements,” Pattern Recognition, vol. 45, no. 1, pp. 162–171, Jan. 2012 .

[11] L. Schomaker and M. Bulacu, “Automatic writer identification using connected-component contours and edge-based features of uppercase Western script,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 6, pp. 787–798, Jun. 2004.

[12] G. Ghiasi and R. Safabakhsh, “Offline text-independent writer identification using codebook and efficient code extraction methods,” Image and Vision Computing, vol. 31, no. 5, pp. 379–391, May 2013.

[13] R. Jain and D. Doermann, “Offline writer identification using k-adjacent segments,” International Conference on Document Analysis and Recognition, Beijing, China, pp. 769–773, 2011.

[14] C. Djeddi, I. Siddiqi, L. Souici-Meslati, and A. Ennaji, “Textindependent writer recognition using multi-script handwritten texts,” Pattern Recognition Letters, vol. 34, no. 10, pp. 1196–1202, Jul. 2013.

[15] D. Lowe, “Distinctive image features from scale-invariant keypoints,” International Journal on Computer Vision, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[16] G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis, “Text line and word segmentation of handwritten documents”, Pattern Recognition, vol. 42, no. 12, pp. 3169–3183, Dec. 2009.

[17] V. Papavassiliou, T. Stafylakis, V. Katsouros, and G. Carayannis, “Hand-written document image segmentation into text lines and words”, Pattern Recognition, vol. 43, no. 1, pp. 369–377, Jan. 2010.

[18] http://www.iam.unibe.ch/fki/databases/iam-handwriting-database