# Development of a Writer Verification System-
# A Case Study on Bangla Scripts


Thesis submitted by
Jaya Paul


DOCTOR OF PHILOSOPHY (Engineering)


Department of Computer Science and Engineering,
Faculty Council of Engineering & Technology,
Jadavpur University
Kolkata, India

2024

# JADAVPUR UNIVERSITY
## KOLKATA-700032, INDIA

**INDEX NO. -102/17/E**
**Registration Number: 1021704008**

1. Title of the Thesis:

   **Development of a Writer Verification System- A Case Study on Bangla Scripts**

2. Name, Designation & Institution of the Supervisor/s:

   (a) Dr. Anasua Sarkar
   Assistant Professor
   Department of Computer Science and Engineering
   Jadavpur University, Kolkata-700032, India

   (b) Prof. Kaushik Roy
   Professor
   Department of Computer Science
   West Bengal State University, Barasat, India

*List of Publications

## Papers in Journals

1. Paul, Jaya, Dutta, Kalpita, Sarkar, Anasua, Roy, Das, Nibaran, & Roy, Kaushik, "A survey on different feature extraction methods for writer identification and verification." International Journal of Applied Pattern Recognition, 7(2), 122-144,2023.

2. Paul, Jaya, Dutta, Kalpita, Sarkar, Anasua, Roy, Kaushik, and Das, Nibaran, "Writer verification using feature selection based on genetic algorithm: A case study on handwritten Bangla dataset", ETRI Journal (2024), 1–12, DOI 10.4218/etrij.2023-0188, Impact Factor(1.3).

3. Paul, Jaya, Dutta, Kalpita, Sarkar, Anasua, Roy, Kaushik, & Das, Nibaran, "An integrated approach towards improvement of writer verification performance using page level, line level and word level data on Bangla handwritten document."(Communicated)

## Papers in Conference Proceedings

1. Paul, Jaya, Sarkar Anasua, Das Nibaran, and Roy Kaushik. ”HOG and LBP Based Writer Verification.” In Proceedings of International Conference on Frontiers in Computing and Systems: COMSYS 2020, pp. 3-12. Springer , 2021.

2. Paul, Jaya, and Sarkar Anasua. ”Handwritten Bangla numeral recognition using convolutional neural networks.” In 2018 2nd International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech), pp. 1-4. IEEE, 2018.

3. Paul Jaya, Sarkar Anasua, Halder Chayan, and Roy Kaushik. ”Offline writer verification based on Bangla handwritten characters using enhanced textual feature.” In Hybrid Intelligent Techniques for Pattern Analysis and Understanding, pp. 1-22. CRC Press, 2017.

4. Paul, Jaya, Dutta, Kalpita, Sarkar, Anasua, Roy, Kaushik, and Das, Nibaran ”Tri-scripts writer verification based on handcrafted feature vs vision transformer learning”. (Accepted for publication)

## List of Presentations in National/International Conference:

1. Paul, Jaya, Sarkar Anasua, Das Nibaran, and Roy Kaushik. "HOG and LBP Based Writer Verification." In Proceedings of International Conference on Frontiers in Computing and Systems: COMSYS 2020, pp. 3-12. Springer , 2021.

2. Paul, Jaya, and Sarkar Anasua. "Handwritten Bangla numeral recognition using convolutional neural networks." In 2018 2nd International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech), pp. 1-4. IEEE, 2018.

3. Paul, Jaya, Sarkar Anasua, Das Nibaran, and Roy Kaushik. "A Survey on Feature Extraction Methods For Handwriting Biometric." In 2021 3rd International Conference on Computational Intelligence in Communications and Business Analytics (CICBA-2021), on 7th and 8th January,springer, 2021.

4. Paul Jaya, Dutta Kalpita, Sarkar Anasua, Roy Kaushik and Das Nibaran. " Evaluating the Tri-Script Writer Verification System Using a Handcrafted Features and Vision Transformer Learning Approach." (Accepted for publication).

## Statement of Originality

I, **Jaya Paul** registered on **20.02.2017** do hereby declare that this thesis entitle **"Development of a Writer Verification System- A Case Study on Bangla Scripts"** contains literature survey and original research work done by the undersigned candidate as part of Doctoral studies.

All information in this thesis have been obtained and presented in accordance with existing academic rules and ethical conduct. I declare that, as required by these rules and conduct, I have fully cited and referred all materials and results that are not original to this work.

I also declare that I have checked this thesis as per the "Policy on Anti Plagiarism, Jadavpur University, 2019", and the level of similarity as checked by iThenticate software is 7%.

Jaya Paul
Signature of Candidate:

Date: 08/01/2024

Certified by Supervisor(s)
(Signature with date, seal)

1. Anasua Sarkar 08.01.2024
ANASUA SARKAR
*Assistant Professor*
Computer Science & Engineering Department
Jadavpur University

2. Kaushik Roy 8/1/24.

KAUSHIK ROY, M.E.,Ph.D
*Professor*
Dept. of Computer Science
West Bengal State University

# CERTIFICATE FROM THE SUPERVISOR

This is to certify that the thesis entitled **Development of a Writer Verification System- A Case Study on Bangla Scripts** submitted by Jaya Paul, who got her name registered on 20.02.2017 for the award of Ph.D. (Engineering) degree of Jadavpur University, is absolutely based upon her own work under the supervision of Dr. Anasua Sarkar, Department of Computer Science and Engineering, Jadavpur University, Kolkata-700032, India, and that neither her thesis nor any part of the thesis has been submitted for any degree/diploma or any other academic award anywhere before.

Anasua Sarkar 08.01.2024

..............................................

..............................................

Signature of the Supervisor
and date with Office Seal

ANASUA SARKAR
*Assistant Professor*
Computer Science & Engineering Department
Jadavpur University

Kaushik Roy 9/1/24.

..............................................

..............................................

Signature of the Supervisor
and date with Office Seal

KAUSHIK ROY, M.E.,Ph.D
*Professor*
Dept. of Computer Science
West Bengal State University

# Acknowledgements

Finally, it is time for me to acknowledge all those who inspired me, supported me and helped me to get to the place where I am today.

I take this opportunity to express a deep sense of gratitude to Dr. Anasua Sarkar for her supervision and invaluable co-operation. This thesis would not have been possible without constant inspiration and unbelievable support of them over the last few years.

I have had an amazing group of Labmates. Each of them deserves my gratitude: Kalpita Dutta. Working with them was a great experience that brought together several good and fruitful ideas. Thank you to all of you for the unselfish help, insights, feedback, and for making the science a collaborative effort. With that, a special thanks to Prof. (Dr.) Nibaran Das and Prof. (Dr.) Kaushik Roy. The Journey would remain incomplete without their suggestions, unconditional help, and encouragement.

I would also like to thank all of all faculty members from Computer Science and Engineering Department, Jadavpur University and my for providing me a friendly research environment. Finally, most important of all, I would like to dedicate the thesis to my mother Mrs. Padma Paul, my husband Mr. Biswajit Das and my daughter Shinjini Das, my mother in law Mrs. Sabit Das and father in law Mr. Rarendra Nath Das to honor their love, patience, encouragement, and support during my research.

Jaya Paul

(Jaya Paul)

Date: 08-01-2024

# Abstract

Writer verification plays a pivotal role in various domains, including forensics, document analysis, and biometrics. In this study, we explore the difficulties of the Bangla script's complex structure and various writing styles, suggesting a new way to overcome them. The system's architecture combines advanced image pre-processing techniques with machine learning algorithms, facilitating precise verification of individual writers. A new dataset of handwritten Bangla samples is utilized to train the model. The proposed system demonstrates promising accuracy rates and highlights its potential for real-world applications. This contribution improves the study of document analysis and writer verification, especially regarding the complexities of Bangla scripts. The introductory chapter establishes the significance of writer verification and outlines the research objectives. Chapter 2 conducts an exhaustive review of existing literature, encompassing both online and offline features and introduction of multi-level scripting approach. Identifying and verifying a person based on scanned images copy of their handwriting is a needful biometric application in historical document analysis, behavioral biometrics study, forensic science, access control, graphology, and copyright management. Writer identification and verification are still challenging in offline and online handwriting recognition analysis. Since the performances of handwriting biometric identification and verification systems depend on both the quality and types of chosen features, this is one of the most critical phases. This chapter represents a literature survey on offline and online biometric features used in different scripts for writer verification and identification techniques. Several previous efficient works on online and offline writer authentication methods for biometrics using cutting-edge hand-crafted features in different levels of handwriting analysis like documents, paragraphs, words, and characters are analyzed systematically to date for the first time in detail. Chapter 3 explores dataset analysis and creation, finding gaps in existing resources, and introducing a new dataset created for the research. In details chapter 4, we explore the important role of features in figuring out who wrote a piece of text. We focus on the features that make each writer's style unique, covering linguistic, stylometric, and structural traits. We explore features at different levels in the Bangla script to get a detailed view.

First, we talk about why extracting features is so important in recognizing patterns. We then look at features, like shapes and textures. Shape detection involves using the Radon Transform, and we also use various texture-based descriptors at different levels. After that, we dive into using a Genetic Algorithm to pick out the most important features. This algorithm helps us find the features that make our model work the best. Finally, we shift to using auto-derived features, which means letting a computer learn the features on its own. We use different models like Alex-Net, VGG, ResNet, and Vision Transformer to help us recognize patterns in handwriting. We explain how these models work and the settings we use to make them effective. Chapter 5 focuses on how computers can learn to identify different writers' writing styles using classification algorithms. These algorithms are like smart tools that can sort and categorize data, helping in tasks such as recognizing images or analyzing text. Machine learning, a part of artificial intelligence. It's like teaching computers to learn from data, recognize patterns, and make predictions without being told exactly what to do. Machine learning is used in various areas like recognizing images, understanding speech, and making recommendations. In the main part of this chapter, we describe about different machine learning methods. We use tools like Weka to apply these methods to handwritten script recognition. We explore classifiers like Multilayer Perceptron (MLP), Support Vector Machine (SVM), Simple Logistic, Sequential Minimal Optimization (SMO), Radial Basis Function (RBF) Networks, and K-Nearest Neighbor (KNN).

We explain each classifier, like how Sequential Minimal Optimization (SMO) is good for large feature vectors in handwritten images, or how Radial Basis Function (RBF) Networks are effective in recognizing patterns. We also introduce SimpleLogistic, Multilayer Perceptron (MLP), Support Vector Machine (SVM), and K-Nearest Neighbor (KNN) as different tools for the job. Especially, we look at how Support Vector Machine (SVM) is useful in telling apart different writers' writing styles. We use features like word frequencies and sentence structures to train the SVM model. We measure its success using accuracy and precision metrics. Finally, we touch on K-Nearest Neighbor (KNN), a simple but effective tool that looks at the neighbors of a data point to figure out its category. This chapter explores different tools to find out which ones work best for identifying writers based on their writing styles.

Chapter 6 address different experimental results. In the first experiment, the study focuses on offline Bangla handwriting content and evaluates the approach using specific hand-crafted features with Simple Logistic and RBF networks, SMO, and auto-derived features using a CNN architecture. The hand-crafted feature set outperformed auto-derived features, achieving 94.54% average verification accuracy on a 100-writer database. Hand-crafted features included Radon Transform, Histogram of Oriented Gradient, Local Phase Quantization, and Local Binary Pattern, extracted from inter- and intra-writer content. The Genetic Algorithm reduced feature dimensionality and selected salient features using a Support Vector Machine fitness function. The top five experimental results were obtained from the final set of optimal features selected by a consensus strategy. Comparisons with other methods and features have shown satisfactory results.

The second experiment presents a comprehensive methodology that integrates techniques at the page, line, and word levels to verify the identity of the writer.

A method is developed, leveraging the newly created dataset, JUDVLP-BLWVdb, which significantly enhances page-level writer verification performance. By employing the ensemble technique of majority voting, three classifiers (Support Vector Machine, Multilayer Perceptron, and Simple Logistic) are amalgamated, yielding a notable 12% enhancement in writer verification accuracy at the page level. This achievement reaches an impressive 97.62% accuracy across 101 diverse writers. This paper compares results with state-of-the-art writer verification approaches and explores deep learning-based methods, including VGG16, ResNet34, and AlexNet.

Final and last experimental results indicate that the SMO classifier outperforms other classifiers such as simple logistics and KNN. A novel dataset for writer verification systems using the tri-script approach is introduced, achieving a peak verification accuracy of 91.50% through a combination of Radon Transform, HOG, LBP, and LPQ features. The overall performance of the tri-script approach reaches 91.80%. Furthermore, this study employs the Vision Transformer (ViT) model for writer recognition, demonstrating the superior performance of ViT when using tri-level block images of the page.

Lastly, Chapter 7 provides a conclusive summary of the thesis, highlighting its contributions and discussing potential avenues for future research in the field of writer verification within Bangla scripts. This structured progression showcases the evolution of ideas and methodologies, ultimately contributing to the advancement of writer verification techniques, particularly in the intricate realm of Bangla scripts.

# Contents

# List of Figures

# List of Tables

# 1 Introduction

## 1.1 Introduction

### 1.1.1 Handwriting Analysis

Handwriting analysis [1], also known as graphology, is the study and interpretation of handwriting characteristics [2] to gain insights into the personality traits, emotions, and behavioral patterns of an individual. It involves analyzing various aspects of handwriting, including letter formations, slants, spacing, pressure, size, and overall style. Handwriting analysis is often used in forensic investigations to analyze and compare handwriting samples for purposes such as signature verification, document authentication, and suspect identification [3]. It is also employed in fields like personnel selection, career counselling, and personal development to gain insights into an individual's strengths, weaknesses, and compatibility with certain roles or professions.

### 1.1.2 The Internet of Things

The Internet of Things (IoT) encompasses a network of interconnected physical devices, objects, and systems that communicate and exchange data over the Internet. Equipped with sensors, software, and other technologies, these devices can collect and share data autonomously, without human intervention. An IoT-based writer verification system utilizes this technology to authenticate and verify writers' identities, ensuring content integrity. This system enhances security and accuracy across various digital platforms. The IoT enables diverse applications, such as biometric registration [4], smart devices [5], real-time verification [6] and multi-factor authentication [7].

### 1.1.3 Biometric

Biometric authentication is a cutting-edge technology that has revolutionized the way to verify and confirm individuals' identities. By physical and behavioral characteristics, such as fingerprints [8], voiceprints [9], facial features, or even typing patterns, biometric systems offer highly secure and accurate methods of identification. Unlike traditional passwords or PINs, biometric data is difficult to forge or replicate, making it a robust solution for various applications. From unlocking smartphones and accessing secure facilities to authorizing financial transactions and enhancing user authentication in IoT environments, biometrics plays a pivotal role in safeguarding sensitive information and ensuring a seamless user experience. As the technology continues to advance, biometric authentication is becoming increasingly prevalent across industries, offering a future where individuals can confidently and securely prove their identities with just a touch or a glance.

### 1.1.4 Behavioral Biometrics

Behavior biometric systems [10] collect and analyze data from various interactions, such as keystrokes, mouse movements, touch gestures, signature dynamics, and even the way a person walks. These behavioral patterns are then used to create a unique biometric profile for each individual. The advantage of behavior biometrics lies in its ability to provide continuous authentication, as users' behavior can be continuously monitored during their interactions with devices or systems, adding an extra layer of security.

### 1.1.5 Handwriting biometrics

Handwriting recognition or dynamic signature verification [11], is a behavioral biometric technology that focuses on analyzing and identifying individuals based on their unique handwriting patterns. Each person has a distinct way of writing, including variations in pressure, speed, pen angle, and stroke patterns, making handwriting an ideal candidate for biometric authentication. Handwriting biometric systems capture and analyze various characteristics of a person's handwriting when they sign their name or write a sample text. The system creates a unique biometric profile that reflects the individual's specific handwriting traits. This profile can then be used for subsequent verification and authentication purposes.

### 1.1.6 Writer identification and verification

Writer identification and verification are essential processes in the field of biometrics that aim to determine the identity of an individual based on their writing style or handwriting. These two processes have distinct purposes.

Writer Identification [12] which involves the task of determining the identity of an unknown writer by comparing their handwriting or writing style with a database of known writers. It is often used in forensic handwriting analysis to identify potential suspects or authors of anonymous documents. The process typically involves extracting specific features from the handwriting, such as stroke patterns, letter formations, and overall writing style, and then comparing these features with those of known writers to find potential matches.

Another one is writer verification [13] which is the process of confirming whether a person claiming to be a specific writer is indeed that writer. In this scenario, a known sample of the writer's handwriting is compared to a question sample to determine the degree of similarity. It is commonly used for secure authentication in various applications, such as digital signatures, access control, and document verification. The goal is to ascertain if the provided sample matches the expected writing style of the claimed writer. Figure 1.1 shows the outline of the writer verification system. Both writer identification and verification rely on behavioral biometrics, as they analyze unique behavioral patterns exhibited in an individual's handwriting. These biometric methods are utilized in fields like forensic investigations [14], law enforcement [15], finance [16], and other areas where accurate authentication and identification of writers are of utmost importance. In advanced technologies, including machine learning and pattern recognition algorithms, have significantly improved the accuracy and efficiency of writer identification and verification systems, making them valuable tools for ensuring security and reliability in a wide range of applications.

## 1.2 Objective

The objective of the thesis is to design and develop a specialized writer verification system for Bangla handwriting. The thesis aims to address the limitations of existing systems in dealing with Bangla scripts, by collecting a comprehensive dataset from native Bengali writers and incorporating deep learning and computer vision techniques. The developed system will be evaluated and compared to existing approaches, contributing valuable insights and solutions for writer verification tasks specific to the widely used Bangla language, with potential applications in forensic analysis, document authentication, and security-related tasks.

## 1.3 Research gap

Existing writer verification research has predominantly focused on non-Indian languages, leading to a scarcity of comprehensive studies specifically tailored for Indian scripts like Bangla. The crucial research gap lies in the necessity to develop a specialized writer verification system that effectively addresses the complexities and unique characteristics of Bangla scripts. While machine learning techniques have shown promising results in writer verification for non-Indic scripts, their

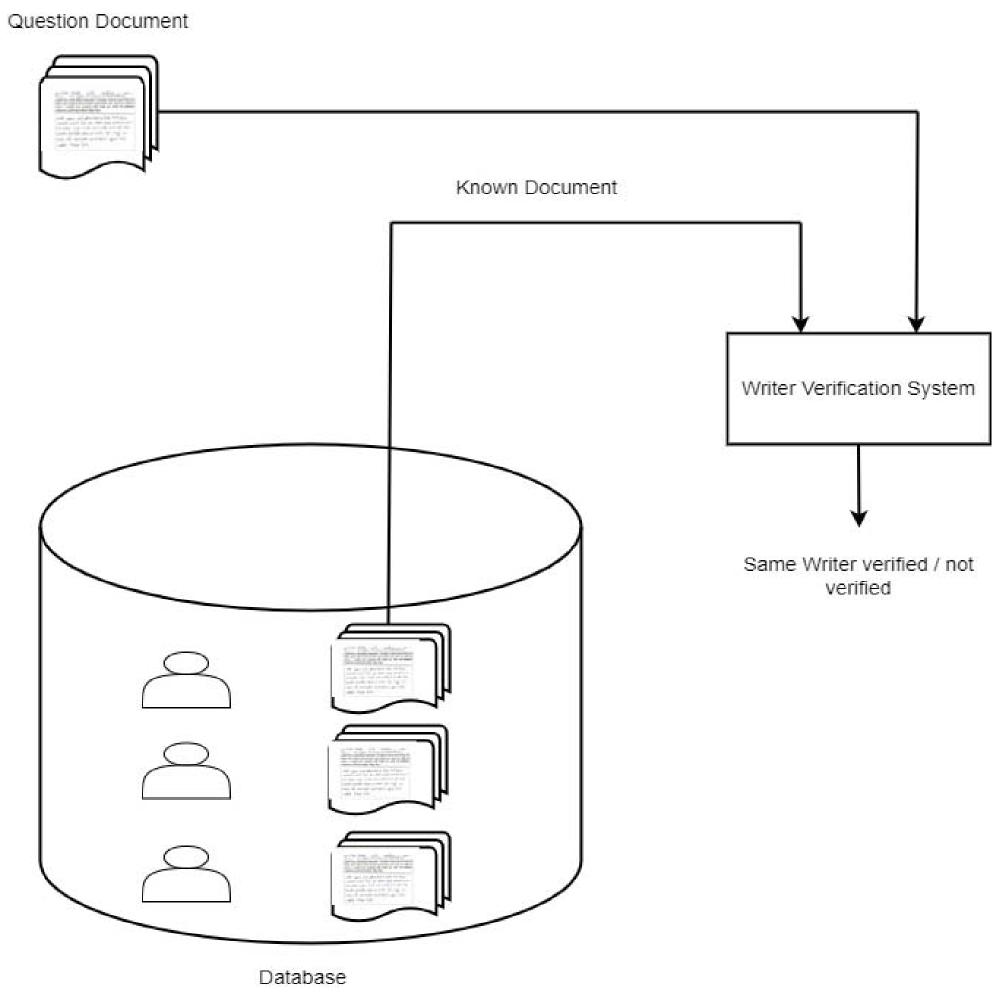



Figure 1.1: Outline of Writer verification system

application and adaptation for Bangla scripts have been largely unexplored. The research gap lies in the need to develop and evaluate machine learning models that can effectively analyze Bangla handwriting patterns and accurately differentiate between individual writers.

The availability of standard datasets plays a crucial role in training and evaluating writer verification systems. However, the research gap in this field pertains to the limited availability of publicly accessible Bangla handwriting datasets for writer verification purposes. Creating and curating a suitable dataset specifically tailored to Bangla handwriting verification is essential to advance research in this area.

In the context of writer verification systems, evaluation metrics commonly utilized are optimized for non-Indic based languages. However, these metrics may not be directly suitable for evaluating writer verification in Indic scripts, such as Bangla. The research gap exists in establishing appropriate evaluation metrics that can effectively and reliably assess the performance of proposed writer verification methods specifically tailored for Bangla scripts. Defining metrics that consider the unique characteristics and complexities of Bangla handwriting is crucial to ensure accurate and meaningful evaluation of writer verification systems in this linguistic context. Addressing this gap will contribute significantly to advancing writer verification research for Bangla and other Indic languages.

## 1.4 Motivation

The rapid growth of digital communication and information exchange demands reliable writer verification systems to ensure the authenticity and integrity of textual content, particularly in multilingual environments.

Bangla is one of the most widely spoken languages in India, witnessing significant digital content creation. However, verifying the legitimacy of authors and their texts in the Bangla script poses challenges. Existing methods for writer verification predominantly focus on non-Indic-based languages, limiting their applicability to Indic scripts like Bangla, which requires tailored approaches due to its unique characteristics and complex graphemes.

This thesis aims to bridge this gap by proposing an advanced system that accurately verify writers in Bangla scripts. Leveraging machine learning and pattern recognition techniques, the envisioned system will efficiently analyze handwriting patterns in digital documents.

The research outcomes have potential applications in online content authentication, digital document forensics, and ensuring author credibility in digital publishing platforms. Additionally, this study will contribute to the field of multilingual writer verification, benefiting Bangla scripts.

Through this case study on Bangla scripts, seek to enhance the accuracy and reliability of writer verification systems for Bangla and other Indian languages facing similar challenges. The implications extend beyond linguistic boundaries, impacting digital communication worldwide, and fostering trust and credibility in the digital age.

## 1.5 Thesis organization

The thesis consists of a total of seven chapters, including the introductory to concluding chapters. Chapter 2 offers a comprehensive literature survey concerning the relevant research topic. The literature review covers an analysis of both online and offline features, encompassing diverse scripting methods. Furthermore, it provides an overview of the multi-level scripting approach in relation to writer verification and writer identification efforts. Chapter 3 outlines the publicly available databases utilized in the field of writer verification, while also identifying any gaps within these resources. Additionally, this chapter elaborates on the creation of a dataset for use in my research. The rationale behind developing this new dataset is also explained in detail. Chapter 4 introduces a approach to enhance handwriting verification through hand-crafted features, employing Genetic Algorithm optimization. Focused on Bangla handwriting, the study showcases the superiority of hand-crafted features, achieving an impressive 94.54% average verification accuracy across a dataset of 100 writers. The Genetic Algorithm efficiently reduces the dimensionality of features and selects relevant ones, leading to significant performance improvements when compared to alternative techniques and feature sets. In Chapter 5, the challenge of enhancing Bangla handwriting analysis is addressed, particularly focusing on improving page-level writer verification. The proposed integrated approach combines techniques across page, line, and word levels, resulting in an improvement of nearly 12% to achieve 97.62% accuracy using an ensemble of classifiers. Comparative evaluations and further exploration of deep learning underscore its effectiveness. Chapter 6 addresses the challenge of verifying writers across Bangla, Hindi, and English scripts. By combining handcrafted features and SVM classification, the approach achieves a maximum 91.50% accuracy and an overall tri-script performance of 91.80%. The ViT model also demonstrates superior performance on tri-level block images. Lastly, Chapter 7 provides the conclusion and future scope of the thesis.

# 2 Literature Survey

## 2.1 Introduction

The word ’biometrics’ comes from’ bios’ (mean life) and ’metrics’ ( mean measure). In the last few years, behavioural biometrics had several applications for personal authentication and are widely used in forensics to detect identity and security applications. Handwriting biometric systems have mainly two different parts: verification and identification. In the writer verification phase, a user claims their identity, and the system checks if the writer is indeed who their claims to be. It becomes an important task in QDE (Question document Examination) [17]. In the writer identification system [18], the writer provides a biometric handwriting sample as a input. The system identifies it among all users who have enrolled in the system. Although writer verification and identification two are different problems, there are similarities in data acquisition, interpretation and solution methodology phases. Figure 2.1 shows the framework of an writer authentication system [19].

An online system needs special machine-related tools like a digital pen or tablet to write and automatically digitize handwriting content. However, offline methods involve our manual tools, pen and paper for writing content. The collected sample was digitized from analogue to the digital process according to the need of the work. Offline and online approaches used for handwriting document analysis works with multi-lingual scripts. Offline writer identification and verification are very complicated compared to the online mode due to the different factors variations in handwriting style, online and offline media quality, and other preprocessing methods. Online writer verification and identification systems implemented using spatial, temporal and pressure information [20, 21, 12] of the writing. An offline system implemented only using spatial information [22, 23].

The signature verification [24] is an essential biometric trait in financial, official and legal spheres due to verifying a person’s identity. These verification systems automatically discriminate if the signature sample is valid for a claimed individual. Otherwise, query signatures are classified as genuine or forgeries. Forgeries are

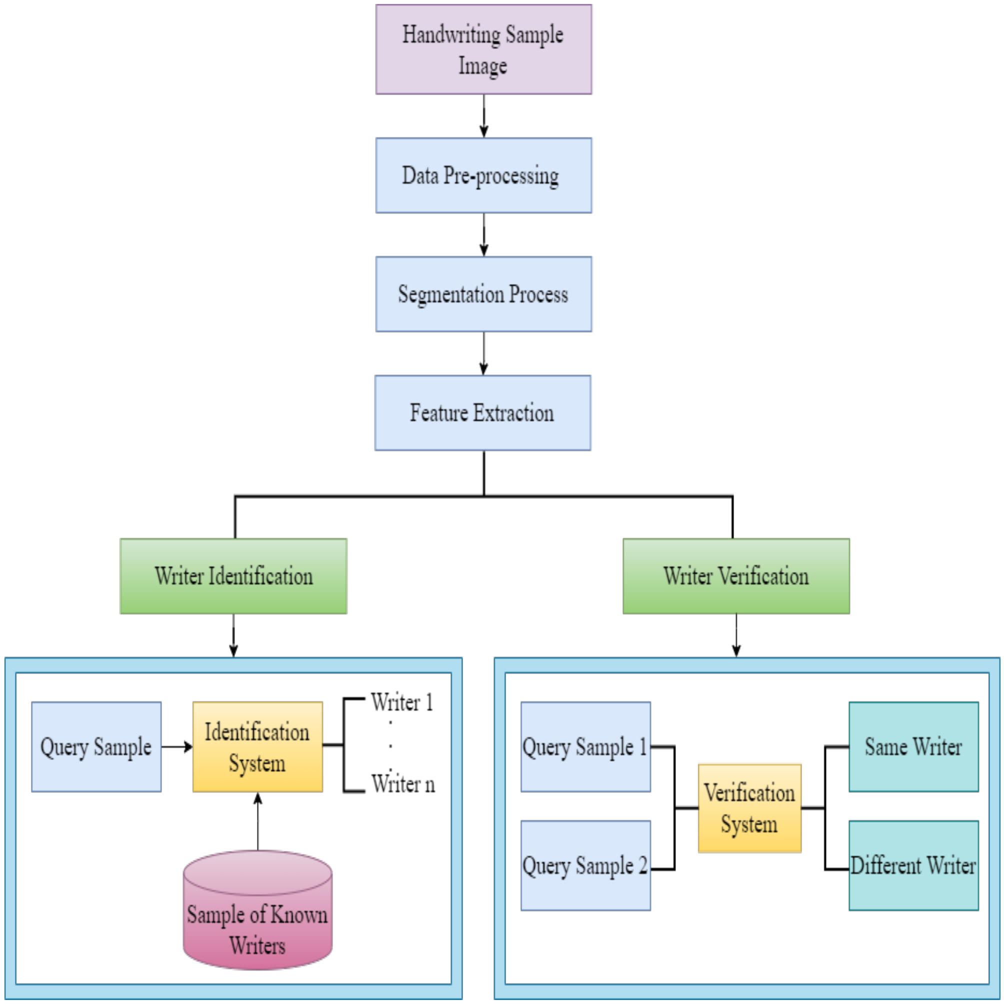



Figure 2.1: Writer authentication system framework

three types: simple, skilled and unskilled. These types of authentication systems can also have used in hand mobile devices, historical document analysis [25], writer identification and verification systems [26] and security applications [27, 28]. Different types of people have been involved in these experiments with these systems to obtain adequate and efficient solutions. Figure 2.2 categorises different handwriting types. Handwriting biometric analyses are categorized as writer identification, writer verification and writer analysis. The flow of this literature survey is presented in Figure 2.2. Offline (pen & paper) and online modes (writing on electronic devices) are two main categories. Document, paragraph,

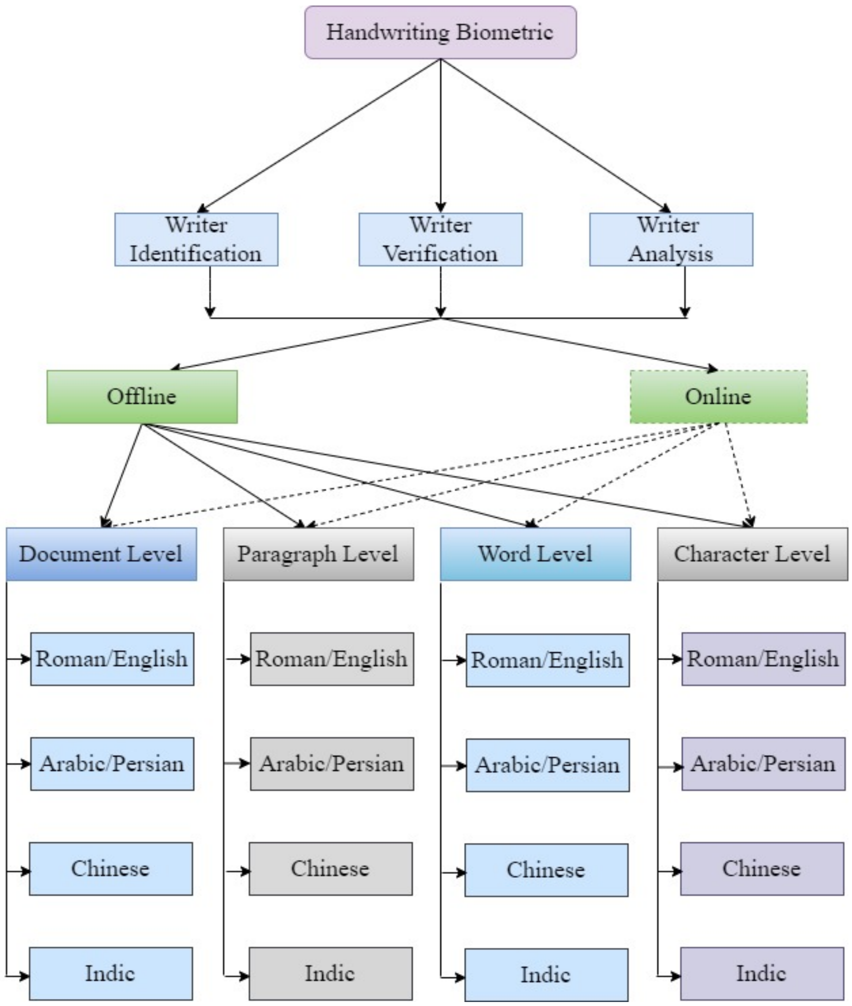



Figure 2.2: The survey on different features extracted from different levels of different scripts

line, word, character, and hybrid level tasks are available in handwritten biometric analysis, which can be specific to any script or even multi-scripts. The authors have summed up the methods for feature extraction and classification problems in this area. Awaida and Mahmoud [29] have mainly included writer identification and verification systems of Arabic script and Persian (Farsi) script, similar to the Arabic script. The author Welekar and Rao [30] survey on online and offline writer identification methods, mainly non Indic script (English and Arabic) and Indic script (Bangla, Malayalam and Gujrati). They also focus on text dependent and text independent writer verification techniques. Girdher and Sharma [31] review the writer identification system on Indic script. Halder et al. [32] presented a survey on the different offline state-of-the-art techniques for writer identification and verification tasks. So, different types of surveys [29, 31, 32, 33] published on the different segments (related to writer recognition) to cover this domain in past years. In the past few decades, a plethora of traditional techniques has been identified in this domain. Nevertheless, the utilization of deep learning approaches in the writer identification and verification domain has been notably limited [34, 35].The writer recognition system marked a pivotal moment by introducing the first deep learning approach, pioneered by Fiel and Sablatnig [36]. Authors worked on ICDAR, IAM, ICDAR 2011, ICDAR 2013 and CVL databases with a few samples. Yang et al. [37] worked on online writer identification with automated feature extraction methods like CNN. They have achieved 99.52% accuracy on the CASIA-OLHWDB1.0 dataset with few samples. Noted that the research on deep learning approach performance does not reach satisfaction. In most cases, deep learning approaches are too computationally expensive for real applications, while the traditional counterparts have much lower performance. This paper focuses on different scripts (non-Indic and Indic) at various levels (document, paragraph, line, word, and character) and online and offline techniques for writer identification and verification tasks.

Reviewed the literature on various handcrafted features used in handwriting script identification and verification tasks. Different types of handcrafted feature extraction techniques are used in several languages, and this article discusses the results of these techniques. Our proposed model is also comprehensively explained in our other positions, which are referenced as [38, 39].

### 2.1.1 Research Design and Data Collection

This section outlines how this study was planned and how data was gathered. It explains the careful steps taken to design the research and collect information systematically to achieve the study's goals. The complete PRISMA investigation is illustrated in Figure 2.3, providing a visual representation of the systematic approach employed in preparing and summarizing the literature.

### 2.1.2 Standards for identification:

To identify and collect relevant publications for writer verification and identification, a comprehensive search was conducted on platforms such as Google Scholar, IEEE-Xplore, ResearchGate, Springer, DBLP, MDPI, etc. Specific search phrases, including ”writer detection,” ”deep learning-based writer identification,” and others, were employed to enhance the search performance. Similarly, for

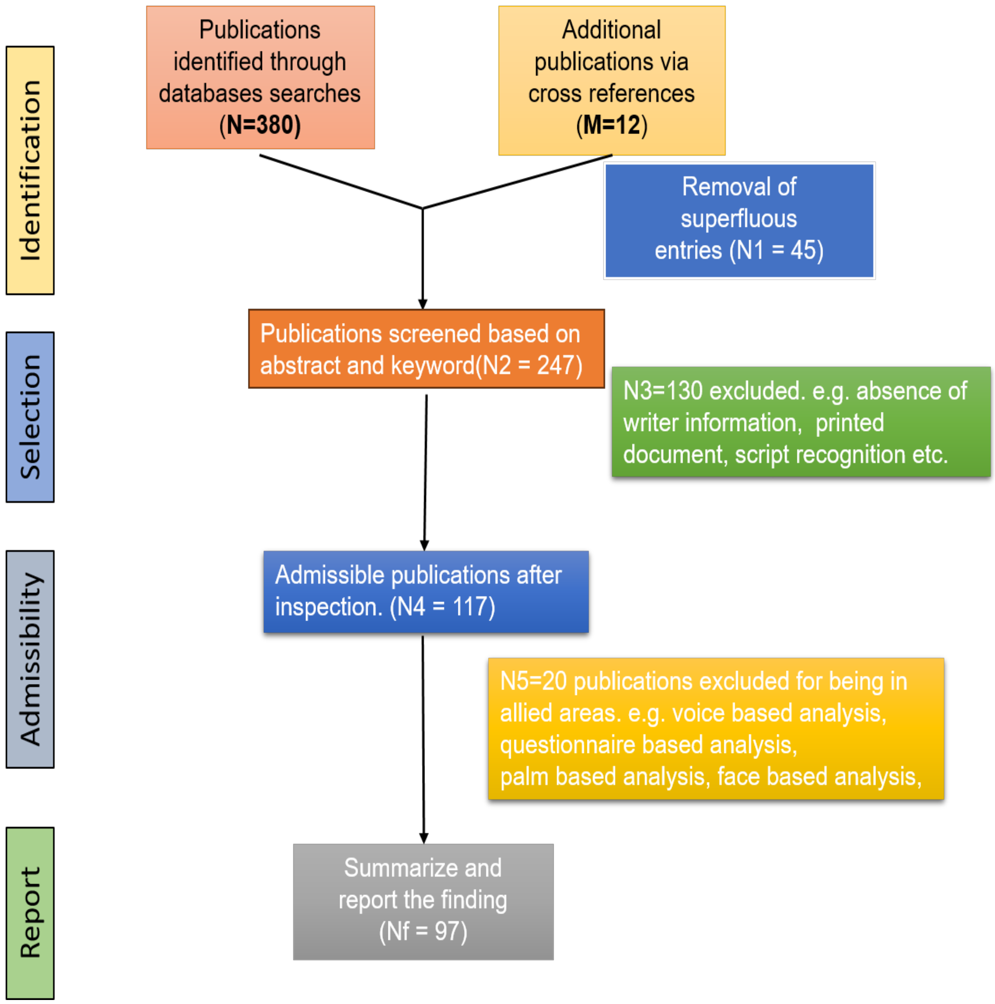



Figure 2.3: The systematic journey: navigating identification, selection, and analysis of publications in the survey

writer identification-related papers, phrases like "writer identification in the wild," "writer classification in handwriting images," and others were utilized.

**Selection Criteria:**

Papers were chosen through a meticulous screening of keywords and abstracts, guided by specific considerations

- Techniques employing handcrafted methods for writer verification and identification,

- Conventional machine learning-based approaches for writer separation,
- Utilization of deep-learning frameworks for handwriting prediction,
- Implementation of deep learning-based techniques for feature extraction, writer identification, and verification,
- Approaches involving transfer learning in writer analysis

Following a thorough inspection against these criteria, publications that did not meet these standards were excluded.

**Admissibility Criteria:**

Dataset details were cataloged according to the diverse methodologies employed in handwriting extraction, writer identification, verification, and/or writer recognition/analysis. Publications were categorized based on the datasets used in competitions, whether employing handcrafted, deep learning, or hybrid techniques. The sequence of publications was organized chronologically, focusing on year-wise methodological progress. Rigorous examination and summarization of these works were conducted, culminating in the preparation of a report based on the observed findings from related studies.

**Features at different levels**

**Document level features:**

Tayeb Bahram [40] introduced a technique for writer identification based on handwritten documents, incorporating Modified Local Binary Pattern (MLBP) and measurements of Ink-trace Width and Shape Letters (IWSL). The proposed system was evaluated on eight established handwriting databases. Remarkably, the recommended approach achieved the highest performance on six benchmarked databases, namely KHATT, CVL, Firemaker, BFL, CERUG-CN, and ICDAR2013. Texture based [41] features are Local Binary Patterns (LBP), Local Ternary Patterns (LTP), Local Binary Patterns (LBP) and Local Phase Quantization (LPQ) have been used for writer identification on IFN/ENIT and IAM databases. This study has encompassed handwritten databases comprising both Arabic and English scripts. Bulacu and Schomaker [42] have formulated a text-independent method for writer identification in Arabic handwriting documents. Their approach proves highly effective, particularly in considering two levels of features: texture level and allograph level, with the extraction of two types of features. Allographs level methods like shape code-book are used on 61 writers to train, and 350 writers with five samples have been used in writer verification and identification to tests. Singh et al.

[43] work on page level Indic script identification in Bangla, Devnagari and Telugu languages using texture features, namely Gray Level Co-occurrence Matrix. The accuracy of their identification results reaches 91.48% using the MLP classifier. Obaidullah et al.[44] present an Indic script identification approach using two convolution-based techniques. Twenty dimensional Gabor and Morphological feature vectors have used 157 document images (in Bangla, Devnagari, Roman and Urdu scripts). Finally, they have reached 94.4% average identification rate on the test set. Described all of these in the section of Table 2.1,2.2.

In online document level systems, some works have been done in English, or Roman [45], Arabic [46] and Chinese hand writings [47]. The PDM technique [47] has focused on the variations in parametric shape forms. They have used the sum of the eigen modes as a similarity metric. The PDM approach has worked in two-mode steps. Mode one is the alignment training data necessary to remove the shape variations due to affine transformation. Principal component analysis is commonly used to reduce the dimension of the feature, but here authors used the modal analysis to find proper shape variations. Mode one result is better than mode two, 97.3% accuracy as writer identification. Al-Dmour and Zitar [46] perform an earlier writer identification approach using the images of Arabic handwriting compared to Roman handwriting in a text-dependent approach. Kameya et al. [48] have presented an online writer verification method based on the document level method. Their method name is called Continuous Dynamic Programming, which extracts the features. They have proposed a figure based online writer verification technique. An end-to-end system called DeepWriterID, presented by Yang et al. [49], employs a deep convolutional neural network (CNN) known as DropSegment to address various problems. This method aims to achieve data augmentation and enhance the generalized applicability of CNN. The experiments were carried out on the NLPR handwriting database, exclusively utilizing pen-position information during the pen-down state of the provided handwriting samples. Remarkably, this approach attained new state-of-the-art identification rates, reaching 95.72% for Chinese text and 98.51% for English text.

**Paragraph level features:**

Schomaker and Bulacu [50] have created a new technique with connected component contours used in uppercase handwritten samples for the offline writer identification method. This method obtains very high corrected identification rates. Also, they have used a codebook of connected-component contours. The signal-to-noise ratio value, as presented by Audrey [51], has been employed in the paragraph-level feature process for writer identification. Bulacu and Schomaker [42] introduced a system that evaluates the performance of text-independent Arabic handwriting writer identification methods compared to Western scripts. Their experiments, conducted using the IFN/ENIT database, yielded highly effective results. Lastly, they conclude that combining textural features with allographic features may get high performance. In the study [52], the authors proposed a set of novel geometrical features, which included direction, curvature, and tortuosity. They also suggested an enhancement of the edge-based directional and chain code-based features. These methods were applied to Arabic and English paragraph-level handwriting, achieving 82% and 87% accuracy on the respective datasets.

Online paragraph level text independent writer identification approaches have been experimented with in [53]. Multi-channel Gabor filter has been applied as the static feature (texture) and dynamic features. They conducted their experiments on the NLPR handwriting database, involving 55 writers. Another online writer identification system was proposed by Schlapbach et al. [54]. They introduced the task of writer identification for online handwriting, captured from images on a whiteboard. The point-based and stroke based features have been used there. Gaussian mixture models (GMMs) are used to evaluate their models. They have archives 98.56% rate of writer identification accuracy on the paragraph level.

**Line level features:**

Schlapbach and Bunke [55] introduced a text-independent writer verification and identification system that used six local and three global features based on Hidden Markov Model(HMM), which acts on handwritten English text lines. A Log likelihood score has been generated for the writer identification and verification phases using HMM. These scores are then stored and used to create the rankings for obtaining results. Chaudhuri and Bera [56] have presented a text line identification approach based on the interdependency between text line and inter line gaps of handwritten Indian scripts like Hindi, Bangla, English, and Malayalam etc. Goyal and Bathla [57] provide a new projection-based approach for line segmentation. They segment one line into different segments: skewed lines, touching lines, and upper modifiers in broken parts. This method has been applied to other Indian multilingual scripts like Devanagari and Hindi. In very recent work, Sulaiman et al. [58] proposes a combination system of both the hand-crafted and deep descriptors based Writer identification method to learn patterns of images from the handwritten dataset. They have worked on three benchmark datasets like IAM, CVL and Khatt. These proposed methods' outcome has high capabilities resulting from both German and English scripts simultaneously on the CVL dataset.

**Word level features:**

Biometric based writer identification and verification problem is a very active research area. Author, Chapran [59] introduces a biometrics-related writer identification system. Mainly dynamic features and some static features are considered here. Gradient, structural, and concavity features applied to words of Roman or English scripts in [60]. Allographs of combinations or words of handwriting were experimented with in [60] with a writer verification and identification system of over 12000 words of handwriting images extracted from 1000 U.S writers. The method generates very effective results in differentiating handwriting. The researcher Zhang [61] introduced a novel classifier named the Residual Swin Transformer Classifier (RSTC). This classifier is specifically crafted to amalgamate local and global handwriting styles, generating robust feature representations from individual word images. The local information is captured by the Transformer Block, which interacts with individual strokes, while the global information is encoded using the Identity Branch and Global Block, enabling holistic encoding. To assess its effectiveness, the proposed method was evaluated on the IAM and CVL benchmark datasets, demonstrating its superior modelling capability for word-level writer identification.

The authors Tomai et al. [62] introduced a segmentation-free approach to word recognition and built upon the recent success of Convolutional Neural Networks (CNNs) in handwritten word recognition. They proposed a CNN model for recognizing offline handwritten Gurumukhi words, leveraging the strengths of deep learning in image recognition and computer vision. The performance of their system was remarkable, achieving accuracy rates of 95.11% and 94.96% based on the partitioning scheme. Pal et al. [63] introduced a dynamic programming (DP) method with a modified quadratic discriminant function (MQDF), applied to the street name recognition problem. A dataset consisting of 4450 Bangla handwritten street names was utilized in their study. The likelihood has computed to get 0.79% error rates. Authors Pal et al. [64] have proposed another work on a postal automation system for Indian multi-script(English, Bangla and Hindi) and city name recognition problems. They approach it in two ways, one approach identifies the script to use appropriate OCR, and the second one has proposed a system to recognize city names. They have applied the water reservoir segmentation method to the city names divided into characters or parts. Here they have used 64-dimensional features. Finally, 92.25% recognition accuracy has been obtained. Another stroke-based technique has been introduced in this work [65] on handwritten Bangla word images. They have considered only two aspects of the word, namely word length and shape of the word. This experiment generates a 3.8% error. Halder and Roy [66] introduced a segmentation method applied to unconstrained handwritten Bangla words. This method involves dividing the words into two distinct zones: upper and lower. Initially, they compute a histogram of the word and subsequently employ a local approach to determine the different zones of the word. In a related context, Chanda et al. [67] utilized word-wise identification methods in two stages for various scripts, including Devnagari, Roman, and Bangla. Hangarge et al. [68] proposed a word retrieval technique based on directional energy features(Gabor wavelets). The similarity measure technique cosine distance has applied to between two Kannada words. The achieved results are respectively 81.25% in terms of average precision, 82.09% for average recall and 84.53% for F-measure at a threshold of 97%.

An online word level global approach was introduced by Gargouri et al. [69]. They have used different sets of dynamic and statistics features for Arabic words. Word level features such as point, stroke, space between strokes and the whole word are extracted. SVM and DTW classifiers are used for final decision making. Fink et al. [70] proposed a new online approach for the Bangla handwriting word recognition system. They have used hidden Markov models and compositionally scripted sub-stroke level features in the Bangla words. Their work compared four writing models (holistic, combined characters, pseudo-characters and context-dependent). They have achieved better results for context-independent sub-word units and used 14073 Bangla online unconstrained handwritten samples of 163 writers.

**Character level features:**

Aubin et al. [71] use a huge number of numeral image datasets to experiment with the individuality of numerals for the identification or verification approaches. Word model recognition (2 global and 72 local) and Gaussian Mixture Models (GMM) have been used as feature vectors. They used the clusterization method and measured the discriminability for verification. The structural micro features [72] are used on three characters (d, y, f) and grapheme 'th'. An optimal feature set was obtained through a neural network and genetic algorithm-based approach. In this method, twenty-five structural micro-features were extracted from grapheme samples [73]. The DistAl classifiers achieved an accuracy ranging from 87% to 94% through 5-fold cross-validation. In a different study, Okawa and Yoshida [74] employed a generic model incorporating pen pressure information for writer verification. They have discussed intra-person variability effects on the model performances. Their experiment decreases the geomean error rate from 10.0% to 4.6%. The sample size of their proposed work on 54 writers. Authors Li and Ding [75] have proposed a character level semi text independent method of writer verification that deals with the question and reference handwriting. The independent text method used directional element features analysis and three different distance (Euclidean, Manhattan and Chi-Square) metrics to measure the distance between feature vectors. They have achieved 9.96% of the average equal error rate. Halder et al. [76] present the individuality of handwritten Bangla numerals using 400-dimensional feature vectors, extracted features from the numeral character images. These authors have used LIBLINEAR classifier in the Weka [77] tool to show Bangla numeral character ’five’ is more individual than the least individual ’zero’. There was more similarity among the different writers in their work, and they obtained 96.5% accuracy. In work, [78, 79] present writer verification and identification of Bangla and Devanagari handwriting script with textual and gradient based features. Since much research uses multi script online databases at the character level. Nakamura and Kidode [80] work on individuality analysis of online Kanji characters. They have selected a string with four Kanji characters, including most of the basic strokes available on those characters. 1,230 samples were collected from different male and female persons. They collected different stroke-related features (like shapes of strokes, the composition of strokes, relative pen pressure between strokes, writing duration and writing speed). However, only a few research works are available on an Indic script. Sen et al. [81] work on an online Bangla 10000 characters dataset and structural and topological features. Classifier SMO generates 98.5% of recognition accuracy rate combined with the proposed features in their work. The previous authors’ work proposed an offline writer verification method based on HOG and LBP features [38]. The experiment was done using 100 writers. Offered a new database with multilingual data in our previous work, but only used Bangla for executing our last method. 89.62% of accuracy achieved.

## 2.2 Summary

Throughout journey toward writer identification and verification automation, discussed various existing solutions that employ a feature-based approach. In the current scenario, the standard database’s inadequacy of inter-writer and intra-writer samples is the primary issue with database design, particularly for the Indic script. Noticed a dearth of publicly accessible datasets for the Indic script.

As a result, additional work is required to develop additional benchmark datasets and to conduct research on write identification and verification. Creating a collection for a hand-written database is a significant undertaking. As illustrated in Figure 2.4, the majority of experiments focused on writer identification rather than verification. Several techniques from the literature to create the Table 2.3. This paper details the process of identifying and verifying writers in terms of research methods used to produce a satisfactory output. Between 2004 and 2018, the year depicted in Figure 2.4, a structure-based approach was frequently used. From 2010 to the present, texture-based approaches have been widely used. It

Table 2.1: The document, page, line, word, and character level feature sets are summarised in detail (section A)

| Online and Offline Script Level | Features Used | Method Used and tasks | Researchers |
|---|---|---|---|
| Document level | Directional, grapheme, and run-length PDFs | Hamming distance, identification and verification | Bulacu and Schomaker (2007) [42] |
| | Gabor,GLCM | LDC, SVM, WED and K_NN, identification | Al-Dmour and Zitar (2007) [46] |
| | Continuous dynamic programming | Determination of threshold, verification | Kameya et al. (2003) [48] |
| | Curve-based features | Neural network,identification | Namboodiri and Gupta (2006) [82] |
| Paragraph level | Allographic | Distance Chi-squared,identification and verification | Bulacu and Schomaker (2007) [42] |
| | Point based and stroke-based | GMM, identification | Schlapbach et al. (2008) [54] |
| Line level | Global and local features | HMM,identification and verification | Schlapbach and Bunke (2007) [55] |
| | Local binary pattern and CNN | GMM super vector and exemplar SVM, identification | Sulaiman et al. (2019) [58] |

was observed that the majority of research on writer identification and verification has been conducted using traditional algorithms over the years, whereas deep learning-based approaches made their first appearance in 2015. The traditional approach to feature extraction and classification is more applicable in real-world situations than the deep learning-based approach for these problems.

Table 2.2: The document, page, line, word, and character level feature sets are summarised in detail (section B)

| Online and Offline Script Level | Features Used | Method Used and tasks | Author and Year |
|---|---|---|---|
| Word level | SIFT key points and HOG with CNN | SVM Classifier, identification | Kumar et al, (2022)[83] |
| | Textural PDF and Allographic PDF features | CNN, MLP,Verification | Adak et al. (2018) [84] |
| | Histograms of direction chain code | Modified quadratic discriminant function, identification | Pal et al. (2012) [64] |
| | Length and shape of the word | HMM,identification | Bhowmik et al.(2012)[65] |
| | 64-dimensional chain-code-histogram | SVM, identification | Chanda et al. (2009) [67] |
| | Statistic and dynamic features | DTW and SVM, identification | Gargouri et al. (2013) [69] |
| | Scalar feature | HMM,identification | Fink et al. (2010) [70] |
| Character level | Global and local feature | K-NN, verification and identification | Srihari et al. (2003) [85] |
| | weighted direction code histogram,GLCM | SVM,verification | Okawa and Yoshida (2013) [74] |
| | directional element features | Euclidean distance, Manhattan distance and Chi-Square distance, verification | Li and Ding (2009) [75] |
| | graphem surface gray level distribution feature | SVM, verification | Aubin et al. (2018)[34] |
| | MFFT, MGLCM, MDCT, 64-dimensional feature | Mahalanobis distance, LIBLINEAR and LIBSVM, verification, identification | Halder et al. (2015a)[78] and Halder et al. (2015b) [79] |
| | HOG and LBP | KNN, SMO and MLP,verification | Paul et al. (2021) [38] |

Table 2.3: A summary of the features, type of work, classifiers, and models used for writer identification and verification tasks and results.

| Author/s | Task | Techniques | Features and Scheme Classifier Used | Result(% Accuarcy) | Remarks |
|---|---|---|---|---|---|
| Tselios et al. [86] | Writer Verification | Grid based approaches | 28 dimension Grid Features, local feature like orientation, curvature and chain code, SVM | 95.0 | Using novel feature extraction method for writer verification and achieve promising result |
| Abdi and Khemakhem [35] | Writer identification and verification | Grapheme based approaches | Universal synthetic codebooks distance, $\chi^2$ | 90.02 for top1 96.35for top5 and EER is 2.1 | Archive promising results with a large dataset |
| Srihari et al [87] | Writer Identification and verification | Structure based approach | Macro-feature, Micro-feature, ANN | Identification 98.0 and Verification 95.0 | Do the comparison of the writer identification and verification Models |
| Bensefia and Paquet [13] | Writer verification | Combination of structure and grapheme based approach | Structural, statistical and Graphemes, Edit distance | 92.0 | Obtained promising result and same person have different samples to producing different graphemes that cannot be matched the system |

Continued on next page

**Table 2.3 – continued from previous page**

| Author/s | Task | Techniques | Features and Scheme Classifier Used | Result(% Accuarcy) | Remarks |
|---|---|---|---|---|---|
| Bulacu and Schomaker [42] | Writer identification and verification | Texture-based approach | Textual and allographic, SVM | 98.0%, 97.0 | Using ImUnipen, Firemaker and IAM database and get promising results |
| Bertolini et al [88] | Writer identification and verification | Texture-based approach | local phase quantization, Local binary patterns, SVM | 99.2, 95.0 | The same framework using for two problems and achieve the promising result |
| Okawa and Yoshida [74] | Writer verification | Texture based approache | Local directional pattern (LDP), SVM | 96.0 | Generic model to improve verification accuracy and reduce error rate 10.0 to 4.6 |
| Aubin et al [34] | Writer Verification | Grapheme based approach | Simple Graphemes, SVM | 98 | The system did not require special devices for data acquisition and achieve 100% success in identity verification |

Continued on next page

**Table 2.3 – continued from previous page**

| Author/s | Task | Techniques | Features and Scheme Classifier Used | Result(% Accuarcy) | Remarks |
|---|---|---|---|---|---|
| Kore and Apte [89] | Writer Verification | Structural based approach | Spatial Domain, City block L1 distance | 75.89 | First time design the feature with different ink width conditions with large dataset. |
| Adak et al. [90] | Writer identification and verification | Structural based approach and Deep Learning based approaches | macro and micro, contour hinge and CNN, Squeeze net, Xception net, VGG-16 | 96.84,86.02 and 68.09, 82.58 | worked on handcrafted and auto-derived features |
| Paul et al. [38] | Writer Verification | Texture based approach | HOG and LBP, SMO, MLP, KNN | 97.64, 98.74, 99.2 | Archive promising results with a small dataset |

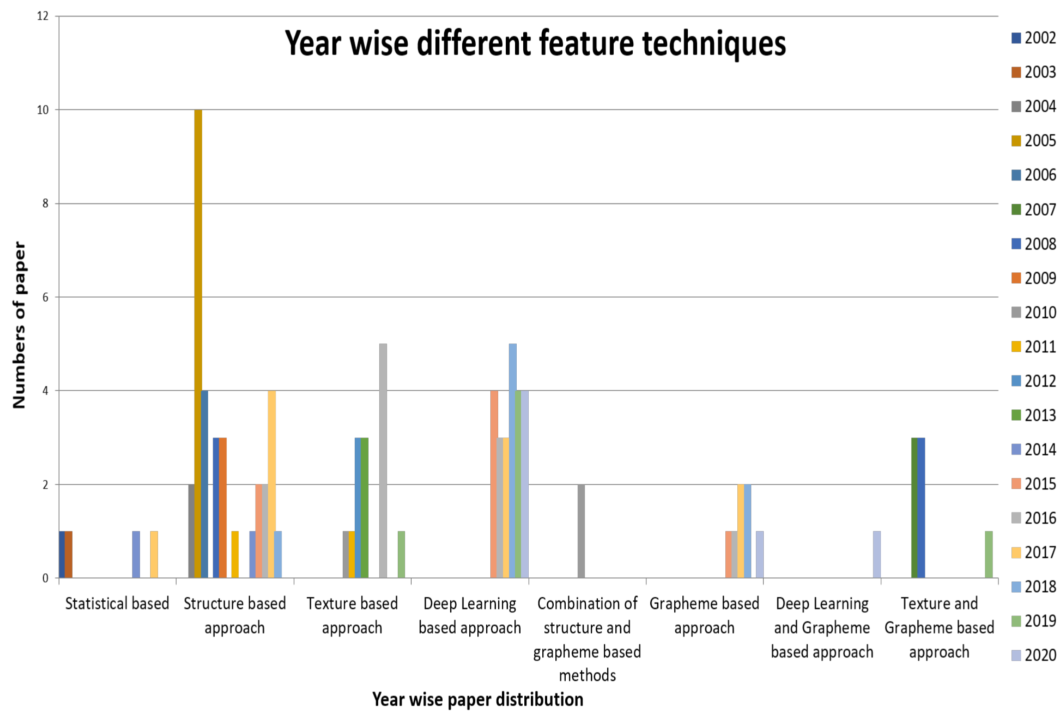



Figure 2.4: Various feature extraction techniques have been applied to handwritten images over the past two decades.

# 3
# Database

## 3.1 Introduction

The database plays a crucial role as it serves as a collection of samples used for training, testing, and evaluation. It includes genuine handwriting samples from known writers and forged or simulated samples created by individuals attempting to mimic the handwriting. The database is divided into subsets for training, testing, and evaluation. Proper data management, including well-organized labelling and metadata, is essential. The database should be regularly expanded and updated to incorporate new samples and account for changes in handwriting styles or forgery techniques. A representative and diverse database is vital for training and evaluating accurate and reliable writer verification systems.

## 3.2 Public Dataset

Over the last two decades, there have been extensive digitization projects aimed at converting paper documents and ancient historical manuscripts into digital formats. However, the lack of robust the handwritten text recognition solutions often renders these documents inaccessible. The ability to recognize handwritten text is crucial for modern document analysis systems. Consequently, there is an immediate need to provide content-level access to millions of manuscripts, personal journals, and large court proceedings. Additionally, developing handwritten text recognition applications can automate the processing of medical transcripts, handwritten assessments, and other related tasks.

Online data acquisition mode for document analysis and recognition process of online identifying or verifying the writer. Handwritten samples were collected online using upgraded electronic devices to reduce the preprocessing and segmentation errors. But Offline data collection is based on scanned text data and converted into a computer as an image file. Offline document analysis and recognition are more complicated than online because of dependence factors like handwriting style variations, paper quality, preprocessing methods, etc. This section

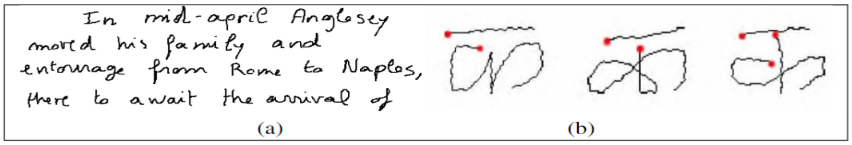



Figure 3.1: Samples of online images: (a) IAM English handwriting a database and (b) Devanagari handwriting database.

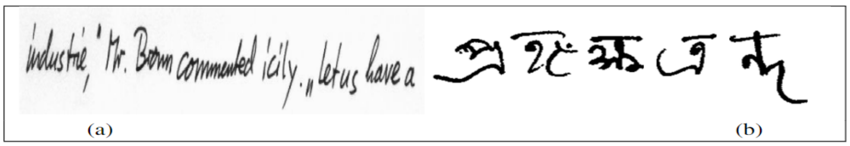



Figure 3.2: Samples of offline images: (a) IAM English handwritten text and (b) CMATERdb3.1.3.1 Bangla handwriting database.

summarizes the different online and offline script databases for document-based biometric analysis. Figure 3.1 and 3.2 presents some sample image of standard online and offline databases. Some previous research works are depicted in Table 3.1. The tasks undertaken by different previous research papers, along with the application of well-known online and offline databases, are summarized in Table 3.1. English standard datasets such as IAM [91, 92], Firemaker set [42, 93], and CEDAR [94, 95, 17] are accessible both online and offline. Standard Arabic language datasets, including the IFN/ENIT database [96, 97] and AHDB/FTR [98], are available in both online and offline modes. Standard Chinese and Indian datasets are HCL2000 [99], ETL1-ETL9 [100] and ISI [101] respectively. German and English combined offline dataset is CVL[102, 103].

Table 3.1: A brief description of multilingual offline and online databases for writer identification and verification tasks

| Database Name | Database Size | Task | Remarks |
|---|---|---|---|
| IAM (Online and offline )[91, 92, 42, 92, 104] | 1. 150 different writers<br>2. 400 writers<br>3. 650 writers, two samples per writer<br>4. 301 writers | 1.Text independent writer identification and verification system<br>2.Offline handwriting recognition<br>3.Text-independent writer identification and verification<br>4.Text independent writer identification | Compare the results of the experiments study have done in English script. The deep learning approach has used a short data sample |
| Firemaker(offline)[42, 93, 105] | 1. 250 writers<br>2. 250 writers | 1. Writer identification and verification<br>2. Writer identification | Bench-mark Dutch script dataset |
| ImUnipen set (Online and offline)[42] | 65 writers, two samples per writer | 1.Character recognition<br>2.Analysis of Line Structure in Handwritten Documents<br>3.Writer verification | English script-based Benchmark dataset |

Continued on next page

**Table 3.1 – continued from the previous page**

| Database Name | Database Size | Task | Remarks |
|---|---|---|---|
| CEDAR (offline)[94, 95, 17] | 1. Cursive handwritten digits and characters<br>2. Total of 1568 pages<br>3. 1,000 writers | 1. Character recognition<br>2. Handwriting recognition<br>3. Writer verification | The database consists of 3.6 GB of English data and intended to encourage research in offline handwriting recognition |
| IFN/ENIT (offline) [96, 97] | 26459 handwritten words | Handwriting recognition | Easy to access the Arabic script |
| AHDB / FTR (offline)[98] | 497 word images | The Arabic handwritten text recognition<br>and writer identification systems | It can freely have accessed by researchers worldwide who are interested in the Arabic script |
| CASIA-OLHWDB1 (offline and online)[37, 106, 37] | 1. 3,866 classes contributed by 420 writers<br>2. 1,280 writers (online and offline) 3. 420 writers (online) | 1. Writer identification(online)<br>2. Writer identification(online and offline)<br>3. Writer identification(online) | Publicly available and online data collection using Anoto pen by Chinese script. Employ deep convolutional network |
| ETL1-ETL9 (offline and online)[100] | 1.2 million character images including Japanese, Chinese, Latin and numeric characters | Offline handwritten Chinese character analysis | the Chinese database have used for free only for research purposes |
| | | | Continued on next page |

**Table 3.1 – continued from the previous page**

| Database Name | Database Size | Task | Remarks |
|---|---|---|---|
| Indian database of ISI (offline)[101, 107] | 1. 1049 Devnagari writers, 556 angla writers, 356 Oriya writers 2. 50 Bangla basic characters | 1.Characters Recognition 2.Characters Recognition | Indian scripts, namely Devnagari, Bangla and Oriya, are described |
| CMATERdb1,2.1 (offline)[108, 109] | 25 men and 15 women, 100 pages in its first version and 50 pages containing both Bangla and English words | Optical character recognition (OCR) | It is a benchmark database for research on offline Bangla and English languages |
| IfN/Farsi-database (offline)[110] | Total of 600 writers produced by 7,271 handwritten Farsi words | City names recognition | A new handwritten Farsi words database is available for research and educational use. |
| IRONOFF (offline and online)[111] | 32,000 characters and 50,000 words | Character recognition | French script database publicly available for research. |
| PE92 (offline)[112] | 2350 handwritten Korean character images which size is 100 /spl times/ 100 with 256 gray levels | Korean character recognition system | Korean character image database has widely used so that the database receives a fair evaluation of its quality. |

Continued on next page

**Table 3.1 – continued from the previous page**

| Database Name | Database Size | Task | Remarks |
|---|---|---|---|
| ICDAR(Offline) [36, 113, 114] | ICDAR-2011(26 writer), ICDAR-2013(50 writers) | Writer identification | Improve the result using deep learning approach |
| CVL(offline)[102, 103] | 311 writers | Writer identification | Publicly available offline database and including two different scripting languages employ a deep learning approach |
| Tamil db (offline) [115] | 50 writers | Handwritten word recognition | Dataset publicly available consists of only 256 city names. |

The table above (Table 3.1) presents publicly available standard datasets for document-based biometric analysis, specifically focusing on handwritten text recognition. Data acquisition methods are categorized into online and offline approaches. Online data acquisition involves collecting handwritten samples using upgraded electronic devices, while offline data collection relies on scanned text data converted into image files. Several well-known datasets, such as IAM, Firemaker set, CEDAR, IFN/ENIT, AHDB/FTR, HCL2000, ETL1-ETL9, ISI, and CVL, are available for researchers to use.

Despite the availability of standard datasets, there are certain gaps and limitations that need to be addressed. Particularly, offline datasets present challenges due to variations in handwriting styles, paper quality, and preprocessing methods, making recognition more complex. Moreover, there may be a lack of sufficient data for Bangla languages or scripts, which limits the applicability of existing solutions.

To address these limitations and contribute to the field, this thesis aims to develop a Bangla offline writer verification dataset. This dataset will serve as a valuable resource for the research community, enabling the development and evaluation of writer verification systems for Bangla handwriting.

## 3.3 Developed Dataset

### 3.3.1 JUDVLP-BLWVdb dataset

Introducing a new dataset to address the limitations identified in Section 3.2, present the Jadavpur University Deep Learning in Vision and Language Processing Bangla Language Writer Verification database (JUDVLP-BLWVdb dataset). Despite our exhaustive efforts, discovered a lack of publicly available Bangla script databases. Given the prominence of the Bangla language in various regions of Eastern India, the absence of a benchmark dataset for writer verification in this language is notable.

To bridge this gap, meticulously collected the JUDVLP-BLWVdb dataset from 101 native Bengali writers at the document level. Each participant was requested to reproduce the same content five times using a standard ball pen with either blue or black ink. The data collection process received valuable assistance from faculty members of the Computer Science & Engineering Department at Jadavpur University. Multiple samples were obtained from each writer to encompass variations in their handwriting styles across different instances.

All writers involved in the study were students in the undergraduate engineering department of the Government College of Engineering and Leather Technology, with ages ranging from 19 to 21 years. The dataset encompasses a total of 488 pages of Bangla script, with contributions from 90 writers, where 5 pages were provided by each of the majority, 6 writers contributed 4 pages each, 4 writers contributed 3 pages each, and 1 writer contributed 2 pages. For the experiments, a total of 3416 lines were selected from the 488 pages, constituting 20,778 words written in the Bangla script. The JUDVLP-BLWVdb dataset serves as a valuable resource for the research community, offering a benchmark for writer verification in the Bangla language. Its meticulous collection and diverse representation of native Bengali writers make it a crucial addition to the field of deep learning in vision and language processing.

Figure 3.3 presents a sample data collection form from the JUDVLP-BLWVdb dataset, which was meticulously designed considering factors such as age, sex, date, time, and vernacular language. The figure illustrates that while writer 1 and writer 2 wrote the same content, their handwriting styles differ due to factors like time, space, mood, and writing speed. As indicated in [116], the JUDVLP-BLWVdb dataset comprises offline handwriting samples of the Bangla script, a language spoken and utilized by a substantial community of over 250 million people. The Computer Science Department of Jadavpur University played a significant role in creating this dataset. Psychological factors such as age, sex, date, and time, which can potentially influence a writer's handwriting, were meticulously taken into account during the data collection process for JUDVLP-BLWVdb. During the

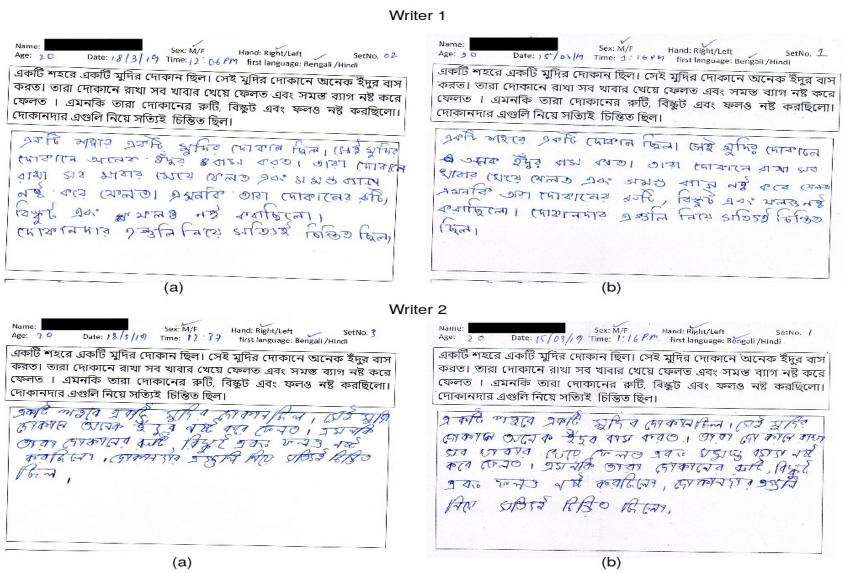



Figure 3.3: Sample images of two different writers taken from JUDVLP-BLWVdb dataset for writer verification task

data collection, each writer was provided with an A4 size paper and a pen (either blue or black ink) featuring a 0.5 - 1.0 mm ball-point tip, allowing them to write in their typical handwriting style. There were no specific restrictions on the writing equipment for the writers, except for the prescribed ink color and tip size.

The JUDVLP-BLWVdb dataset has been divided into four subcategories:

- First page level: This category contains data at the page level, which includes information and content specific to each page.
- Second block level: This category includes data at the block level, where blocks refer to distinct sections or regions within a page.
- Third line level: This category comprises data at the line level, focusing on individual lines of text within the blocks.
- Fourth word level: This category contains data at the word level, with a focus on individual words within the lines of text.

### Page level and Block level

The text pages of handwritten documents were scanned using an HP LaserJet Pro M1136 scanner at 8-bit gray levels and a resolution of 300 dots per inch (dpi). This scanning process resulted in digital text images with dimensions of 2481 x 3507

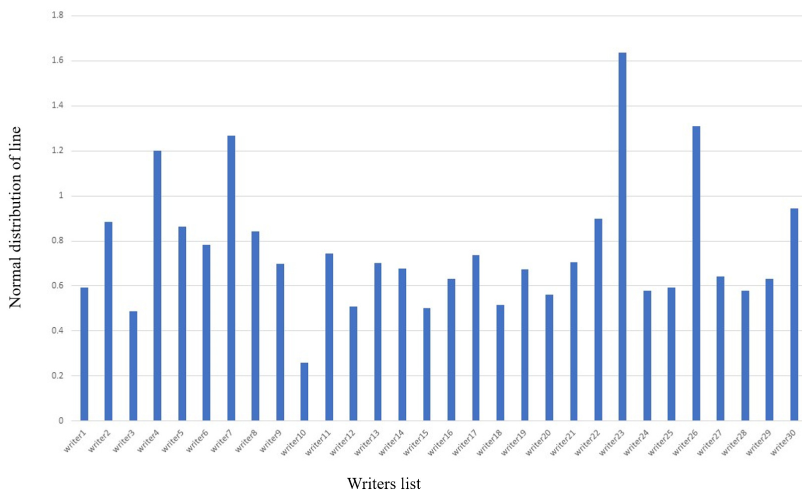



Figure 3.4: Normal distribution of sample text image line

pixels. At the preprocessing stage, label the digital text image data according to the writer's number and set number. The preprocessing process involved converting the digital colour image into a grayscale image, where pixel values were adjusted between 0 and 255 to represent their intensities in the colour space. GIMP software was used to extract the handwritten text content from the grayscale image.

Prior to using the 'crop' tool on the handwritten text, correct skew of the grayscale images. The grayscale image was then automatically converted to a binary image using Otsu's method to determine the optimal threshold [117]. Next, the minimum bounding box algorithm, as detailed in [118], was employed on the binary text image to determine the minimum-area enclosing rectangle.

After extracting the minimum bounding box of the text image, the Gaussian distribution statistical method was applied to ascertain the line distribution, as illustrated in Figure 3.4. If the resulting value is 7 using this method, the text image is divided into a 7x7 grid of rows and columns. Further enhanced the quality of collected writer samples by applying preprocessing steps as illustrated in Figure 3.5. Moreover, Figure 3.6 showcases two page-level Bangla scripts.

To ensure text-dependent work, collected the same writing content from different writers at different times and moods, which was validated by handwriting experts. The dataset was subsequently partitioned into a 2:3 ratio for training and testing. This involved selecting 3 non-overlapping samples out of 5 for training and 2 non-overlapping samples out of 5 for testing for each writer. From a total of 488 pages, chose 291 sample pages for training and 197 sample pages for testing.

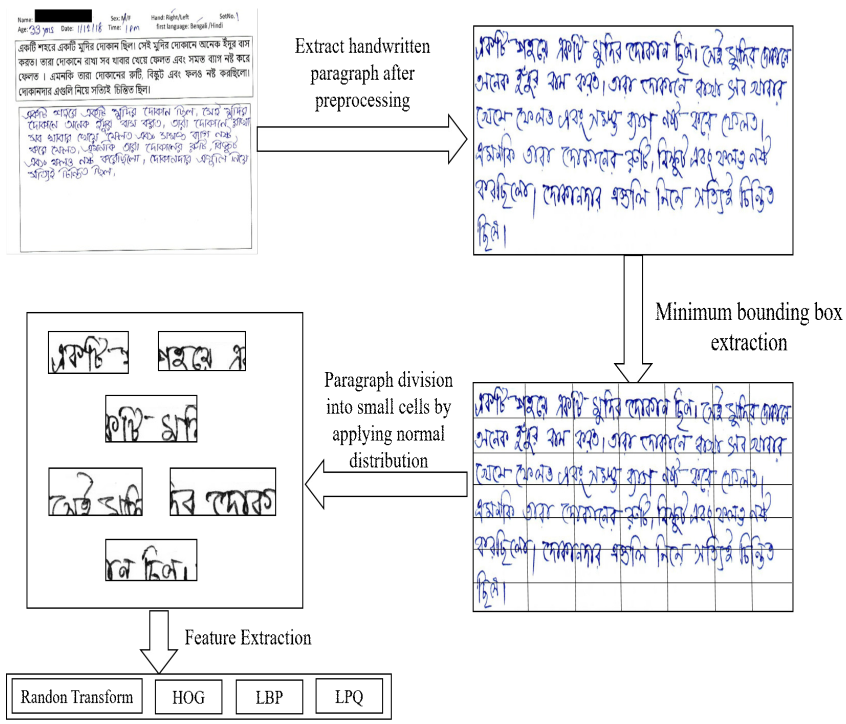


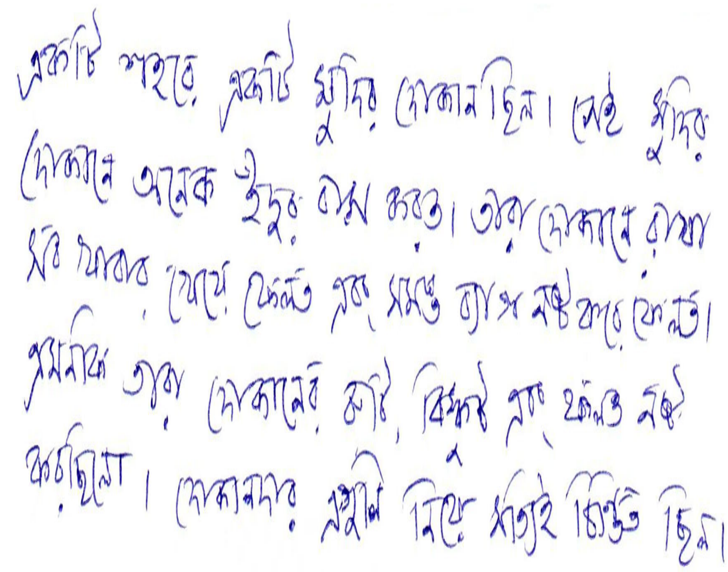

Figure 3.5: The outline of Data preprocessing and feature extraction steps for training and testing phases in our proposed framework

### Line level

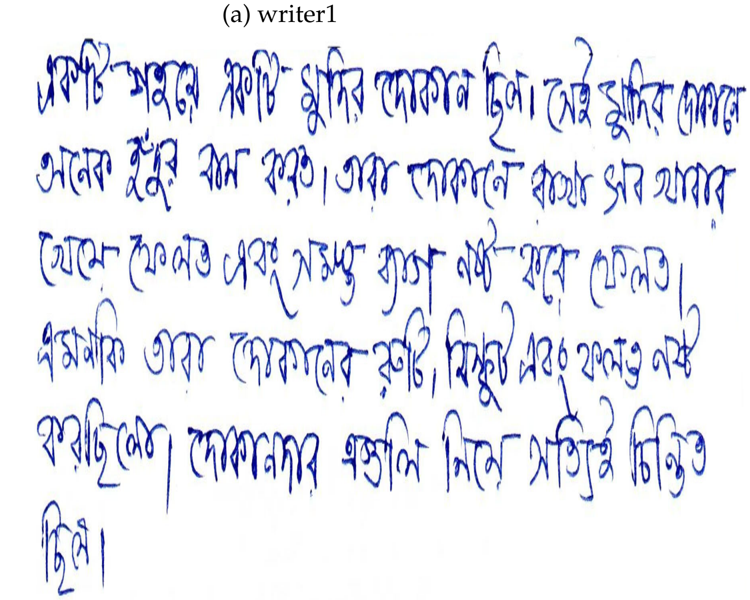
At the line level, 4985 lines of images were utilized for training, while 2896 lines of images were used for testing. Figure 3.7 presents four line-level images of Bangla scripts, extracted from the same page-level images depicted in Figure 3.6. To extract the minimum bounding box of the page-level text image, applied binarization and used horizontal and vertical projection profiles to segment the lines of text. The preprocessing process is visually represented in Figure 3.5. Algorithm 1 delineates the process of extracting data from the page level to the line level.

### Word level

Figure 3.8 displays a sample of word-level images extracted from the same lines of the same page as depicted in Figure 3.7. The line image data underwent Connected Component Analysis (CCA) to extract individual words. The preprocessing process is visually depicted in Figure 3.9. The transition from the line level to the word level is detailed in Algorithm 2. Specifically, at the word level, I used 27,400 sam-



(a) writer1



(b) writer2

Figure 3.6: Sample of writer1 and writer2 page-level of Bangla scripts of the JUDVLP-BLWVdb dataset

ples of word images for training and 18,200 samples of word images for testing.

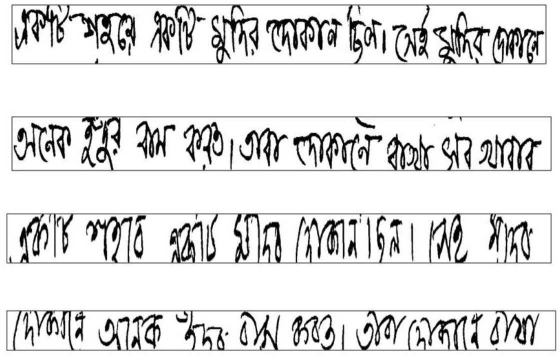


Figure 3.7: Line-level samples extracted from the same page (Figure 3.6) of the JUDVLP-BLWVdb dataset, the top two lines written by Writer 1 and the bottom two lines written by Writer 2 in Bangla scripts

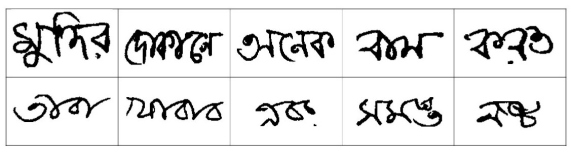


Figure 3.8: Word-level samples extracted from the same page (Figure 3.6) of the JUDVLP-BLWVdb dataset, the top row with 5 words written by Writer 1 and the bottom row with 5 words written by Writer 2 in Bangla scripts

### 3.3.2 JUDVLP-TLWVdb dataset

The Jadavpur University Deep Learning in Vision and Language Processing Tri-language Writer Verification Database (JUDVLP-TLWVdb dataset) has been created for handwritten tri-script writer verification. India, being a highly multilingual country, has over a dozen different spoken languages. Hindi and Bengali are two of the most extensively spoken languages in India, alongside the official use of English. As far as we are aware, there is currently no benchmark dataset publicly available for tri-language writer verification. Therefore, created a new offline handwritten tri-language database, as shown in Figure 3.10. The dataset comprises samples collected from 31 Indian writers at the document level. Although all writers are native Bengali speakers, they are also fluent in Hindi and proficient in reading and writing English. Each writer was instructed to write the same content in multiple languages five times using a standard ball pen with blue or black ink during the data collection phase. The faculty members of Computer Science and Engineering Department at Jadavpur University assisted in collecting this dataset. For text-dependent analysis, the same writing content is in multiple languages at different time intervals. Each writer was provided with A4 size paper

**Algorithm 1** Pseudo-code of Line extract from page image

```
Input: Page level image im
Output: Line level image
input ← READIMAGE(im)
[im1, theta] ← SKEWCORRECTION(input)
if Imagesize(im1) == 3 then
    im ← ConvertRGBtoBinary(im1)
end if
[rs, cs, no.ofcolorbands] ← size(im1)
if (Numberofcolourbands) > 1 then
    im1 ← im1(:, :, 2)
end if
VF ← sum(im1, a)                        ▷ create the projection profiles if a is 2
R ← VF < v                              ▷ v range 1300 to 1700
Toplines ← Find(diff(R) == 1)
Bottomlines ← Find(diff(R) == −1)
for k = 1, ....., N do
    Toprow ← Toplines(k)
    Bottomrow ← Bottomlines(k)
    Thisline ← Image(Toprow : Bottomrow, :)
    if (Rowsofline) < b then            ▷ Here b value is 10
        Continue
    end if
    s ← saveimage(k)
    HP ← sum(thisline, 1)
    k ← k + 1
end for
```

and a pen for writing, without any restrictions on the choice of writing equipment. To ensure uniformity and consistency, designed a data collection form and used multi-script text containing the same meaning for the experiments. In our experiment, considered various factors such as age, sex, date, time, and vernacular language. The dataset distribution for the experiment is as follows: a total of 443 pages (148 pages in Bangla + 147 pages in Hindi + 151 pages in English) from 31 writers.

### 3.3.3 Data preprocessing

All the handwritten document text pages are scanned by a scanner (HP LaserJet Pro M1136) at 8-bit gray levels with 300 dpi (dot per inch) to produce 2481 x 3507 dimension digital text images. Label the digital text image data as per the writer's number and set the number at the preprocessing stage. To convert a digital colour

**Algorithm 2** Pseudo-code of Word extract from Line image

**Input: Line level image im**
**Output: Word level image**

```
input ← READIMAGE(im)
[im2, theta] ← SKEWCORRECTION(input)
[row, col, colorbands] ← size(im2)
[pixelcountgraylevels] ← CreateHistogram(im2)
Binaryimage ← im2
SE ← MorphologicalOperation('Line', len, deg)
BI ← Dilate(BI, SE)
BI ← ConnectComponent(BI, pixel)
M ← 8ConnectComponentProperty(BI, Area, BoundingBox)
M ← [M.Area]
for k = 1, ....., N do
    ThisBoundingBox ← M(k).BoundingBox
    ThisWord ← CropImage(im2, ThisBoundingBox)
    [row, col, colorbands] ← size(ThisWord)
    if (Col) < b then                                  ▹ Here b value is 50
        Continue
    end if
    s ← saveimage(k)
end for
```

image into a grayscale image, the pixel values are adjusted within the range of 0 to 255 based on their intensities derived from the colour space. Use GIMP to extract the handwritten text content from the grayscale image. Before using the 'crop' tool on the handwritten text, correct skew of the grayscale images. After the grayscale image is obtained, it can be automatically converted into a binary image using the threshold acquired through Otsu's method [117]. Then apply the minimum bounding box algorithm [118] on the binary text image to find the minimum-area enclosing rectangle. Figure 3.11 shows the block level tri script sample data.

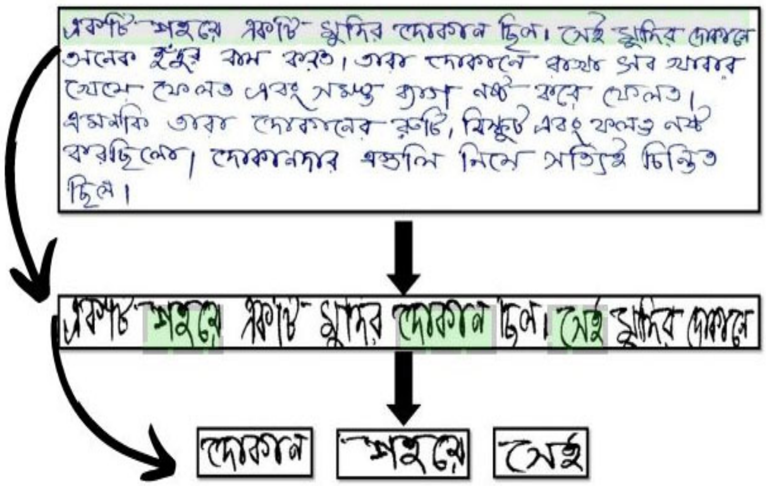


Figure 3.9: Line and word extraction from a page-level sample in the JUDVLP-BLWVdb dataset

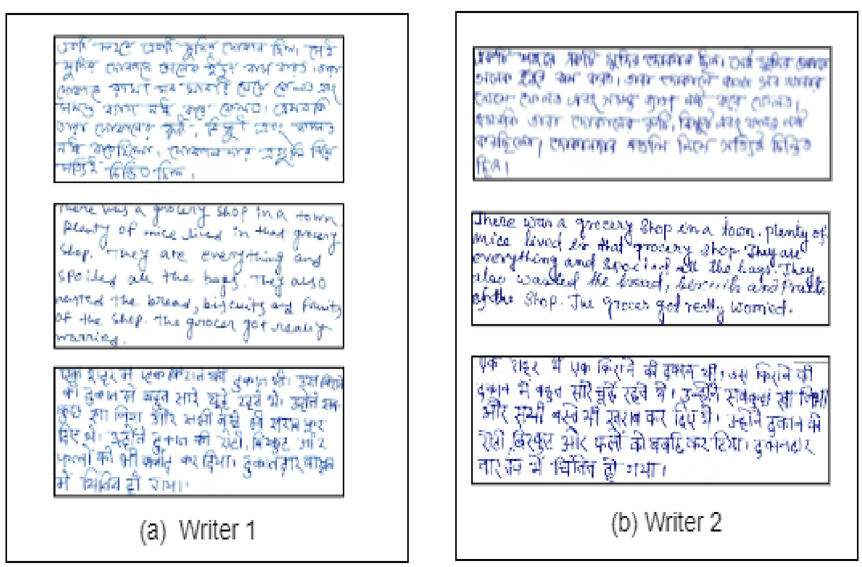



Figure 3.10: Sample database images for (a) writer 1, (b) writer 2

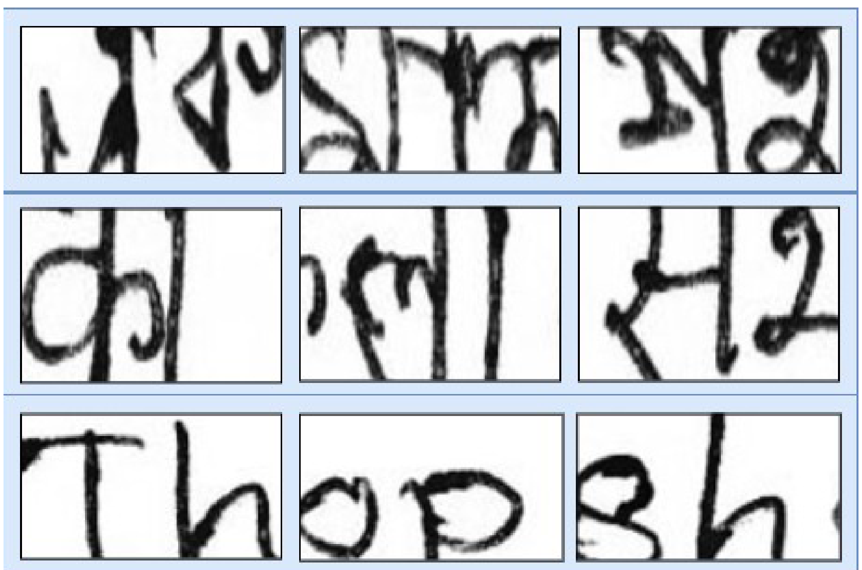


Figure 3.11: Block level tri scripts sample data

# 4
# Features

## 4.1 Introduction

Features are essential for writer verification because they serve as the building blocks for differentiating one author's writing from another. Feature extraction plays a crucial role in pattern recognition problems, as it facilitates the transformation of raw data into a more condensed and meaningful representation. In this study, multilevel document features and block features of the page level in the Bangla script were employed. This approach encompasses the extraction of features at various levels, including page level, line level, and word level. Each level captures distinct characteristics and patterns inherent in the document.

### 4.1.1 Hand Crafted Feature

In this thesis, we have used shape-based and texture-based features. In Chapter 2, after reviewing the literature, we have identified features that have been employed in writer verification, and these features have demonstrated effective results.

#### Shape detection

To compute the Radon Transform, the projection of the page-level and word-level images is determined from a specific direction. This is accomplished by rotating the image around its center to generate various angles. The projection is then computed using Equation 4.1 for a two-dimensional binary image $f(x, y)$ [119]:

$$R(r, \theta)[F(x, y)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)\delta(r - \theta x \cos\theta - y \sin\theta), dx, dy \tag{4.1}$$

In this equation, $\theta$ represents the rotational angle, and the maximum value of the Radon Transform angle $\theta$ falls within the range $1 \leq \theta \leq 180$. As a result, the feature dimension for our experiment is 180-D.

### Texture base descriptor

Writer verification is referred to as a binary classification problem. The model is designed to address a handwriting document's question, which is to determine whether it has been written by the same writer or a different writer. To tackle this challenge, we have extracted various texture-based handcrafted features, including HOG (Histogram of Oriented Gradients), LBP (Local Binary Pattern), LPQ (Local Phase Quantization), GLCM (Gray Level Co-occurrence Matrix), DWT (Discrete Wavelet Transform), and Gabor filters, at the block level of a page, page level, line level, and word level. These features are also utilized in tri-script documents.

- Radon Transform The Radon Transform feature calculation begins with the determination of the projection of an image from a specific direction. Different angles have been generated by rotating around the center of the image. The projection is followed by the equation 4.2 of a two-dimensional binary image f(x, y) [120] as shown below.

$$R(r,\theta)[F(x,y)] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(r-\theta x\ xcos\theta - y\sin\theta dxdy) \tag{4.2}$$

  Here, $\theta$ is the rotational angle, and we ensure that the maximum value of the Radon Transform angle, $\theta$, stays within the range $1 \leq \theta \leq 180$. The dimension of the feature is 180-D for the experiment.

- Histogram of Oriented Gradients Histogram of Oriented Gradients (HOG) [38] is a powerful feature for pattern recognition tasks. The image gradient is calculated for each pixel. This feature is shown in Equations 4.3 and 4.4.

$$dp = I(p,q) - I(p-1,q) \tag{4.3}$$

$$dq = I(p,q) - I(p,q-1) \tag{4.4}$$

  Following that, it is necessary to calculate the gradient magnitude $v$ and orientation $\theta$ as outlined in Equation 4.5.

$$v = \sqrt{p^2+q^2}, \theta = \tan^{-}1(\frac{dp}{dq}) \tag{4.5}$$

  Each image block is scaled to a resize of 64 X 64 pixels before applying HOG features in our work. We have computed HOG features for the tiny fragment of handwritten text using 3x3 HOG windows. This feature dimension is 81-D. We are using nine rectangular blocks per image and 32 bin histograms per block. The nine histograms, each consisting of nine bins, are concatenated to produce an 81-dimensional resulting feature for each image. Figure 4.1

illustrates the output angle and magnitude of the input image.

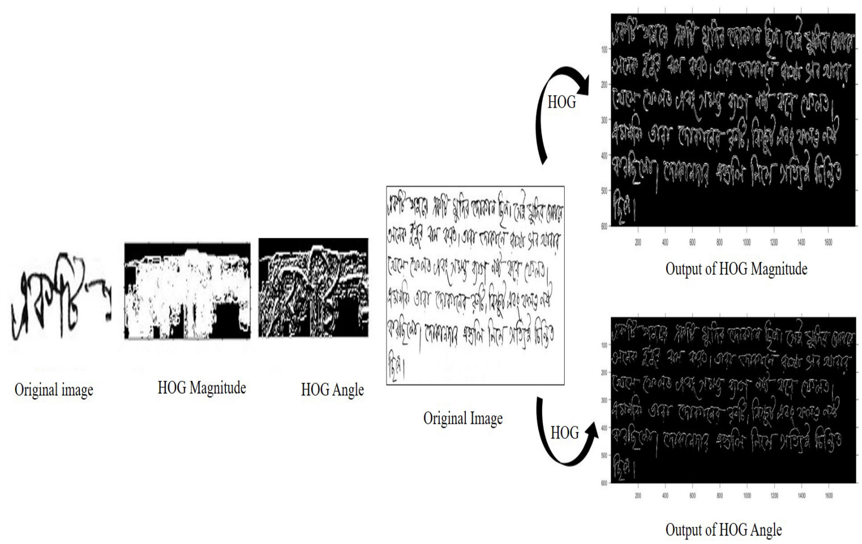



Figure 4.1: Output of described HOG feature on block and page image

- Local Binary Pattern In this chapter, propose a curvature free feature for writer verification based on the Local Binary Pattern (LBP). Apply a grayscale and rotation invariant texture operator based on Local Binary Patterns [121]. This operator measures the spatial structure of the local image texture. The operator computes the LBP transformation of the input image. Normalize each LBP block (block size is 32x32) histogram using the L1 norm. The obtaining feature dimension is 236-D. Figure 4.2 shows the feature output in respect of input image data.

- Local Phase Quantization: Local Phase Quantization (LPQ), introduced by Bertolini [122], serves as a texture descriptor aimed at capturing phase information from line-level and word-level images, subsequently encoding it into a binary pattern. This descriptor involves quantizing phase components obtained from the Short-Term Fourier Transform (STFT) within local neighborhoods. Equation 4.6 defines the computation of STFT over a rectangular $W \times W$ neighborhood $M_i$ at each pixel position $i$ in the image $f(i)$.

$$F(u, i) = \sum_{y \epsilon N_i} f(i, y) e^{-j2\pi u^T y} = w_u^T f_i \tag{4.6}$$

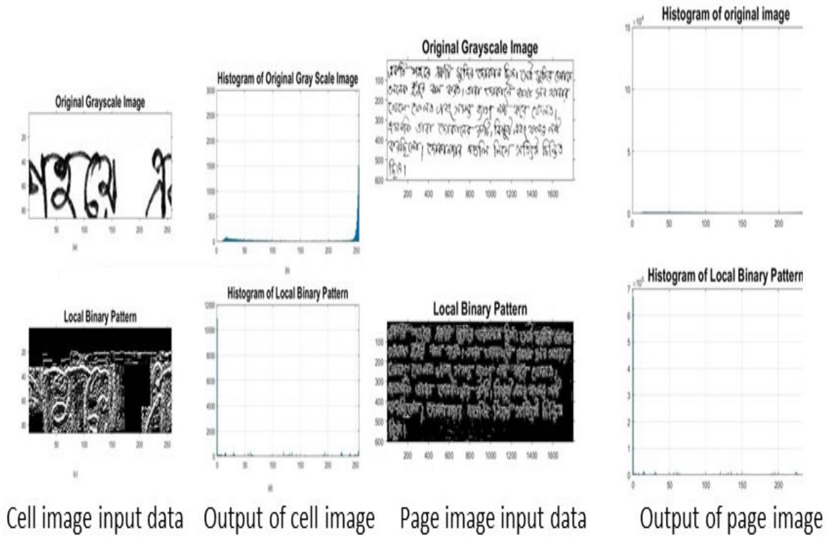



Figure 4.2: Output of described LBP feature on block and page images

The STFT is implemented using 2D convolution $f(i)^{-j2\pi u^T i}$ for all $u$. We obtain a more detailed description of the LPQ approach in [123]. These quantized coefficients are generated in a range of 0-255 integer values using binary code and accumulated in a 256 bin histogram. The resulting vectors are then quantized using a scalar quantizer method and accumulated in a histogram with 256 bins. The presented process generates LPQ features at both the page block and line levels, each possessing a dimensionality of 256, as illustrated in Figure 4.3 for page block level and Figure 4.4 for line level. LPQ exhibits insensitivity to image illumination changes and global image contrast, making it well-suited for analyzing grayscale document images that may exhibit variations in contrast and illumination. LPQ has demonstrated promising results in various document analysis tasks, including writer identification and verification [88], signature verification [124], and multi-script writer identification [122].

LBP methodology and texture information retrieval from histograms of LPQ labels computed within local regions are the same. The traditional LPQ (Local Phase Quantization) works by quantizing the Fourier Transform phase in local neighborhoods at the block-level image, as shown in Figure 4.3.

- Gray Level Co-occurrence Matrix: Texture features extracted from the Gray Level Co-occurrence Matrix (GLCM) [125, 126, 127] offer supplementary insights into the texture patterns present in word-level images. The GLCM

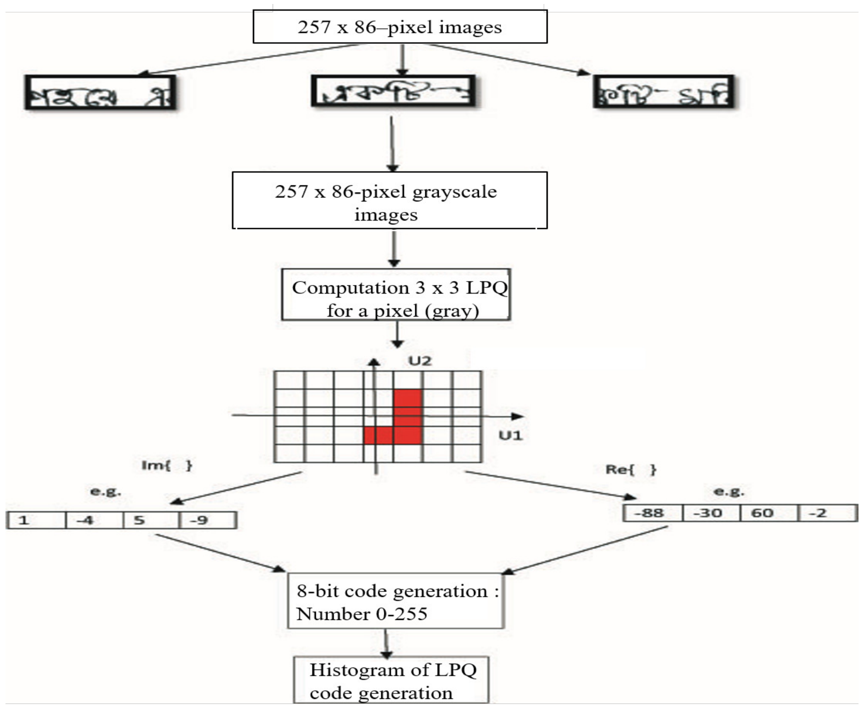



Figure 4.3: Diagram of calculation of LPQ feature on block level

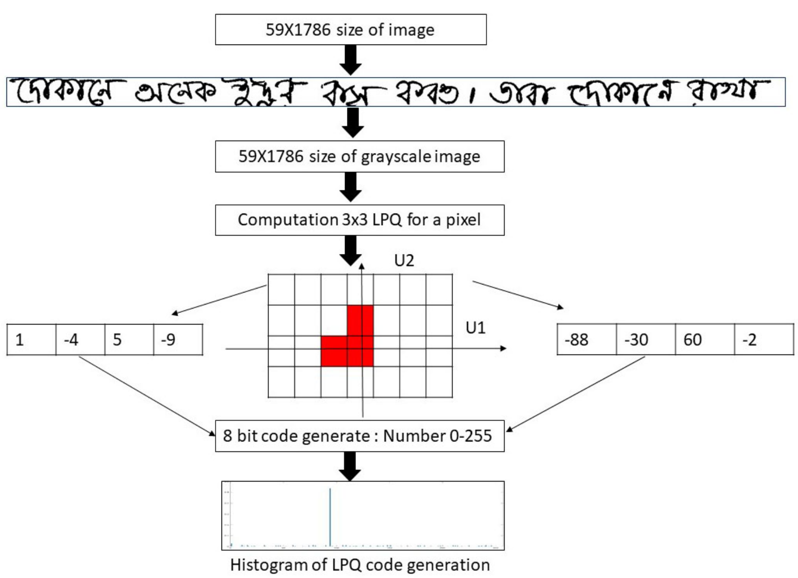



Figure 4.4: Diagram of calculation of LPQ feature on the line level

quantifies the co-occurrence of neighboring gray levels within the region of interest (ROI), generating a square matrix with dimensions equivalent to the number of gray levels in the image.

In this article, the texture features Homogeneity and Contrast are calculated using equations 4.7 and 4.8 respectively. These equations quantify specific characteristics of the GLCM.

$$Homogeneity = \sum_{pq} \frac{k(p,q)}{1 + |p - q|} \tag{4.7}$$

$$Contrast = \sum_{pq} |p - q|^2 k(p,q) \tag{4.8}$$

These equations produce texture features that describe the GLCM. In the context of handwriting analysis, specific writer traces are treated as textures, and discriminant texture features extracted from the co-occurrence matrix are utilized. The resulting feature dimension for our experiment is 5-D.

- Discrete Wavelet Transform: In this chapter, the single-level 2-D Discrete Wavelet Transform (DWT), specifically utilizing the Daubechies Wavelet [128], is employed. The Daubechies Wavelet serves as a widely used basis for image decomposition, providing a multi-scale breakdown of the image at the word level. The 2-D DWT decomposition at level i results in four components: the approximation at level i+1 and the details in three orientations - horizontal, vertical, and diagonal coefficients. These components capture distinct frequency information, presenting a hierarchical representation of the image.

  Entropy serves as a crucial measure in the decomposition process, with various types employed, including non-normalized Shannon entropy, log energy entropy, threshold entropy, sure entropy, and norm entropy. These entropy measures capture different characteristics of the wavelet coefficients, enabling the analysis and description of texture information in the word-level image.

  Figure 4.5 illustrates the outcomes of applying the 2-DWT to the input matrix $I$. In the following expressions, s represents the signal, $s_i$ denotes the coefficients of s in an orthonormal basis, and E is the entropy function. The non-normalized "Shannon" entropy is one such measure employed in this context.

$$E1(s_i) = {s_i}^2 log({s_i}^2) \tag{4.9}$$

The “log energy” entropy.

$$E2(s_i) = log(s_i{}^2) \tag{4.10}$$

The “threshold” entropy. Here P is an optional parameter.

$$E3(s_i) = 1 if |s_i| > pand0elsewhere \tag{4.11}$$

The “sure” entropy.

$$E4(s_i) = s_i{}^2 log(s_i{}^2) \tag{4.12}$$

The “norm” entropy with $1 \leq p$

$$E5(s_i) = |s_i|^p \tag{4.13}$$

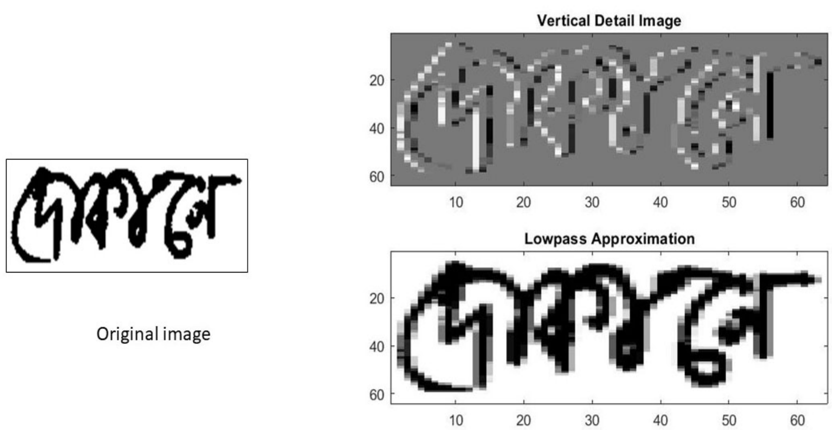



Figure 4.5: 2-Dimensional Discrete Wavelet Transform (2-DWT) applied to a word-level image

- Gabor filter: The Gabor filter bank, utilizing a linear Gabor filter [129], stands as a texture-based descriptor highly sensitive to image textures with specific wavelengths and orientations. Widely employed for texture analysis in image processing, this filter bank extracts texture information at various scales and orientations by convolving an image with Gabor filters of different parameters. In the context of line-level data, applying the Gabor function as a feature allows for the extraction of texture information specific to lines. The resulting output, as illustrated in Figure 4.6, showcases the response of the Gabor filter bank when applied to line data, effectively highlighting texture variations and patterns along the lines. Thus, the Gabor filter bank proves to be a valuable tool for capturing and representing texture information in images. Its application to line-level data is particularly beneficial for tasks such as line detection, texture-based classification, and other line-level analysis objectives.

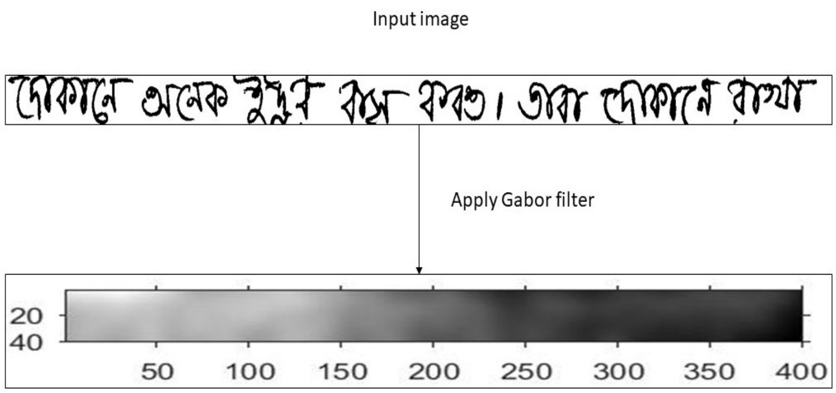



Figure 4.6: Gabor output at line level image

### 4.1.2 Feature Selection Using Genetic Algorithm

Genetic algorithms (GA) are commonly used techniques. In our GA implementation [130], the number of genes represents the total number of features in the dataset. GA consists of several essential factors, including chromosome encoding, population initialization, fitness function (objective function), selection, and termination criteria. The objective function evaluates the quality and effectiveness of each candidate feature subset by assigning a fitness score based on its performance with a chosen evaluation metric. The genetic algorithm then iteratively improves the feature subsets over generations using selection, crossover, and mutation operations guided by these fitness scores. Filter selection evaluates the relevance of features independently of the learning algorithm, while wrapper selection incorporates the learning algorithm itself to evaluate subsets of features. In the chapter, a wrapper selection approach was employed, which incorporates the learning algorithm itself to evaluate subsets of features. This approach allows for the training and testing of the learning algorithm on different feature subsets, enabling the identification of the optimal subset that maximizes performance. The detailed implementation of this GA algorithm is shown in [131]. The parameters for configuring the GA algorithm are presented in Table 4.1.

Table 4.1: Parameters used in GA for optimum subsets selection from the initial large number of features

| GA Parameter | Value |
|---|---|
| Population-size | 50 |
| Chromosome length | 753 |
| Population-type | bitstrings |
| Fitness-Function | SVM |
| Number of generations | 100 |
| Crossover-Probability | 0.8 |
| Mutation-Probability | 0.1 |
| EliteCount | 2 |

### 4.1.3 Auto-Derived Feature

This chapter utilizes a Deep Convolutional Neural Network (DCNN) and auto-derived features to learn image features for pattern recognition, specifically for handwriting text images.

- The Alex-Net model is used due to its advantages over Lenet, such as higher performance and robustness. A simple convolutional architecture of Alex-Net [132] is used to extract auto-derived features, consisting of a front part for feature extraction and a rear part for classification. Pre-processing steps are used to resize the input image to 224 x 224 dimensions and horizontally flip it to introduce variation. The text is divided into block images, and data augmentation is performed using 25 and 35-degree flip operations. The Alex-Net model used for classification has multiple hidden layers, including five convolutional layers, three max-pooling layers, two normalization layers, two fully connected layers, and one softmax layer. Table 4.2 shows the parameter set in Alex-Net. Figure 4.7 has shown our Alex-Net learning model architecture. To mitigate overfitting and augment the dataset, various

Table 4.2: Settings for the parameter values of Alex-Net in our model

| Parameters Used | Value |
|---|---|
| Number of classes | 2 |
| Data augmentation | resized to 224 X 224 |
| Number of Epoch | 50 |
| Batch Size | 1 |
| Number of workers | 1 |
| Loss function | Negative Log-Likelihood |
| Optimizer | Adam |
| Classification purpose | Softmax function |

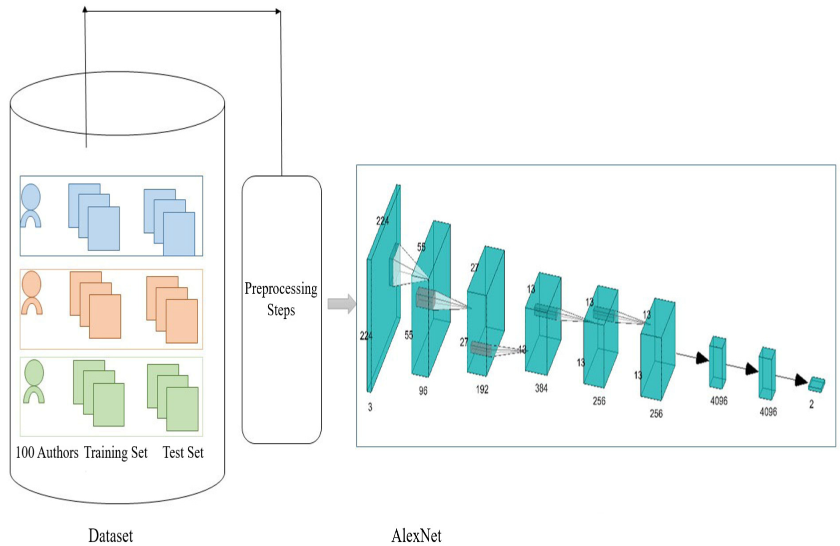



Figure 4.7: The architecture of our auto-derived feature based deep learning model

techniques including flipping, jittering, cropping, and color normalization are employed. This augmentation approach is applied to both our line-level and word-level data.

- The VGG architecture, specifically the VGG16 variant with 16 layers, which demonstrated success in the ILSVRC2014 competition [133], was selected for its uniformity and simplicity. Standardizing input images to a fixed size of 224-by-224 pixels, the detailed architecture information can be found in [133]. This architecture is implemented for both line-level and word-level data in our study.

- The ResNet architecture is thoroughly described in [134], utilizing the ResNet34 variant with 34 layers. ResNet34 comprises $3x3$ filters of the same size as VGGNet for CONV layers. This architecture is implemented for both line-level and word-level data. A learning rate of 0.001 is applied over 100 epochs in our study.

- Vision Transformer (ViT) model In this thesis, we employ auto-derived features in conjunction with the Vision Transformer (ViT) model. This is depicted in Figure 4.8. The ViT model is a deep learning architecture that extends the Transformer model, originally designed for natural language processing tasks, to computer vision tasks. The introduction of the ViT model was presented in [135]. Traditionally, convolutional neural networks

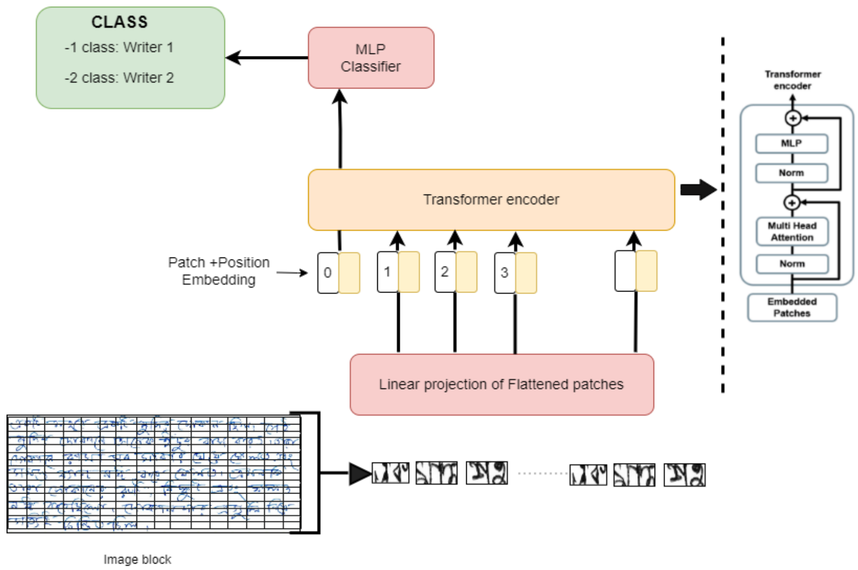



Figure 4.8: Architecture of vision Transformer (ViT) model

(CNNs) have been the prevailing architecture for computer vision tasks, particularly image classification. However, the ViT model takes a distinct approach by leveraging the Transformer architecture, which has exhibited considerable success in natural language processing tasks. The fundamental concept of ViT revolves around treating an image as a sequence of patches, with each patch representing a small spatial region within the image. After extracting patches from the input image, these patches are flattened into a sequence of 1D vectors. These vectors are then fed into the Transformer model for further processing.

Transformer architecture incorporates self-attention layers, which allow the model to focus on various patches and capture their interdependencies. Moreover, ViT incorporates positional embeddings to encode the spatial information of the patches, enabling the model to understand the relative positions and arrangements of the image elements. Here, we employ the ViT-B model with 16 units for patch embedding. For optimization, we utilize SGD with a fixed learning rate of 0.05 and a momentum of 0.9. The model is trained for 130 epochs, with a batch size of 128. The activation function employed is ReLU, and the training process is parallelized with 32 workers.

# 5 Classifiers

## 5.1 Introduction

A classifier is a computer program or a machine learning model that is used to categorize or classify objects, data, or inputs into different categories or classes based on certain features or attributes. Classifiers are a fundamental part of machine learning and artificial intelligence and are used for various tasks such as image recognition, text categorization, spam email filtering, sentiment analysis, and more.

Machine learning is a field of artificial intelligence that focuses on developing algorithms and models that allow computers to learn from and make predictions or decisions based on data. It involves training machines to recognize patterns, make sense of complex information, and improve their performance over time, without being explicitly programmed for every task. Machine learning has applications in various domains, including image, writer and speech recognition, natural language processing, recommendation systems, and predictive analytics, making it a fundamental technology in today's data-driven world.

### 5.1.1 Machine learning Methods

In this chapter, we introduce several machine learning methods that were discussed in the literature survey chapter. Weka tools [136] were utilized to apply classification algorithms to the extracted feature information. Several classifiers, such as Multilayer Perceptron (MLP), Support Vector Machine (SVM), and Simple Logistic, were employed in this context. Additionally, we employed the Sequential Minimal Optimization (SMO) model and incorporated the Radial Basis Function (RBF) network for our analysis, along with K-Nearest Neighbor (KNN). The choice of these classifiers was grounded in their widespread application across various handwritten script recognition tasks, encompassing both identification and verification, as demonstrated in our study.

### Sequential Minimal Optimization (SMO) Algorithm

The Sequential Minimal Optimization (SMO) method, as introduced by Hassen [137], is an algorithm specifically designed for efficient processing of large-sized feature vectors extracted from handwritten images and extensive training and testing datasets. SMO is well-suited for handling handwritten images with large feature vector sizes. In our work, we employed the polynomial kernel, as defined in Eqn. 5.1. In this equation, the polynomial kernel function is denoted as $K(x, y)$, where p is the degree of the polynomial.

$$K(x, y) = (x, y)^p or K(x, y) = (x, y) + 1^p \tag{5.1}$$

To adapt SMO for our specific task, we tuned the parameter values of the SMO classifiers. The complexity parameter C is set to 1.0. The batch size value is set to 100.

### Radial Basis Function (RBF) Networks

An effective and novel type of feedforward artificial Neural Network is the Radial Basis Function (RBF) network [138]. RBF network class implements a normalized Gaussian radial basis function network. The random seed has passed on to K-Means. The number of clusters is set to 2 for K-Means to generate. The minimum standard deviation for the clusters is set to 0.1.

### SimpleLogistic

SimpleLogistic is a classifier methodology [139] for recent developments using a linear logistic regression model. In the SimpleLogistic classifier, set the maximum number of iterations for LogitBoost. This value is 500 for our experiments. For tiny and large datasets, lower and higher values might be preferable, respectively. The beta parameter has been used for weight trimming in LogitBoost.

### Multilayer Perceptron (MLP)

MLP [140] is a type of artificial neural network with multiple layers of interconnected nodes, which can be used for the various machine learning tasks, including classification. It is a feedforward neural network, and it's often employed for tasks like pattern recognition and classification. Figure 5.1 shows the MLP diagram.

### Support Vector Machine (SVM)

SVM [141] is a powerful supervised machine learning algorithm used for classification and regression tasks. It works well for both linear and non-linear classification

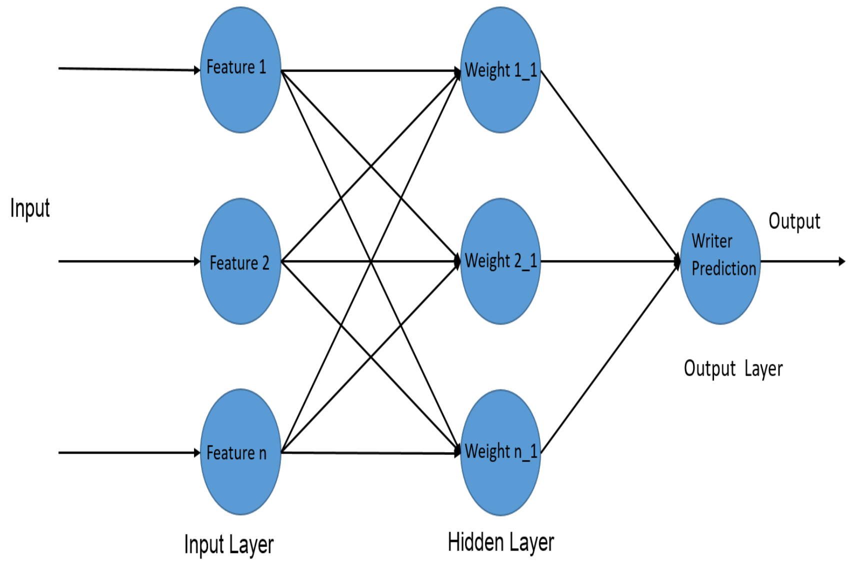



Figure 5.1: Multilayer Perceptron model diagramatically using writer verification method

problems by finding the optimal hyperplane that best separates different classes in the feature space. In this study, we employed a Support Vector Machine (SVM) classifier in Weka to analyze writing styles of various authors for writer verification. The SVM model, configured with a radial basis function (RBF) kernel and optimized parameters, demonstrated promising results in accurately classifying the authorship of text samples. Figure 5.2 shows the SVM diagram. The features extracted from writing, including word frequencies and sentence structures, played a crucial role in training the SVM model. Evaluation metrics, such as accuracy and precision, highlight the effectiveness of the SVM approach in distinguishing between different authors based on their writing styles.

**K-Nearest Neighbor**

KNN [142] is a simple but effective instance-based learning algorithm used for classification and regression. It classifies data points based on the majority class of their K nearest neighbors in the feature space.

### 5.1.2 Deep learning Based Methods

In deep learning Methods, the softmax function is typically used as the activation function in the output layer for classification purposes. The softmax function is employed to convert the raw output of the neural network into a probability

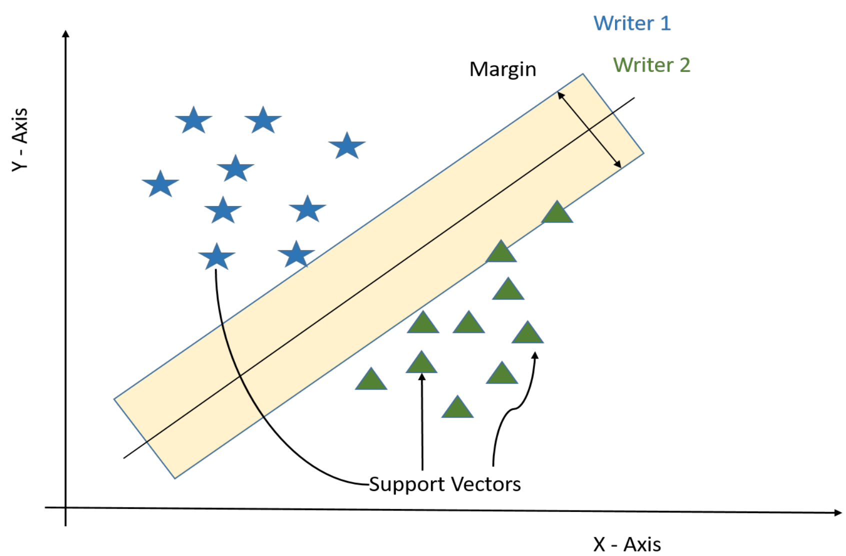



Figure 5.2: Support Vector Machine (SVM) model diagramatically using writer verification method

distribution over multiple classes. This is crucial for binary-class classification tasks.

The softmax function is applied in the context of AlexNet: The final fully connected layer in AlexNet is the classification layer. This layer has $P$ neurons, where $P$ is the number of classes in the classification task. The output of the $i-th$ neuron in the classification layer before the softmax activation is denoted as $y_i$.

$$Softmax(y_i) = \frac{e^{y_i}}{\sum_{j=1}^{P} e^{y_j}} \tag{5.2}$$

Here, $e^{y_i}$ is the exponential function, and the denominator is the sum of the exponentials of all the neuron values in the classification layer. The result of the softmax activation is a probability distribution over $P$ classes. Each value represents the probability of the input belonging to the corresponding class.

Many other convolutional neural networks (CNNs) like VGG16 utilize the softmax function for classification purposes.

ResNet34 also used softmax function 5.2 for image classification. ResNet34 has 34 layers and these layers include residual blocks.

The Vision Transformer (ViT) architecture uses a transformer-based approach for image classification. Unlike traditional convolutional neural networks (CNNs) where convolutional layers are the primary building blocks, ViT relies on self-attention mechanisms present in transformers. Here's an overview of the classification process, including the softmax function 5.2, in ViT. the softmax function in ViT is applied to the output of the final fully connected layer, converting it into a probability distribution suitable for binary-class classification. This probability distribution is then used for loss calculation during training and for making predictions during inference.

# 6 Result and Analysis

## 6.1 Experimental Setup

The experiments were conducted using MATLAB R2017b on a system equipped with an i5-8250U core processor clocked at 1.60 GHz and 8 GB of RAM. In addition, for ViT, AlexNet, ResNet34, and VGG16 experiments, we utilized Pytorch on a computer with an Intel Core i5-9300H processor clocked at 2.40 GHz, x64 architecture, 8 GB of RAM, and a 32.51 GB GPU.

For simulation, we employed the WEKA 3.6 version [143], which offers various machine learning algorithms for easily computing results from learning methods on any dataset. We used the newly developed JUDVLP-BLWVdb and JUDVLP-TLWVdb datasets for our experiments. Common shallow learners were employed as classifiers in our study, encompassing Sequential Minimal Optimization (SMO), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Multilayer Perceptron (MLP), and simple logistic regression. These classifiers were utilized to evaluate different levels of text, including page, line, and word. This chapter presents three different types of experiments conducted on both the JUDVLP-BLWVdb and JUDVLP-TLWVdb datasets.

### 6.1.1 Experiment Setup on JUDVLP-BLWVdb dataset

In the page and block level experiment, used the newly developed JUDVLP-BLWVdb dataset for this experiment. The dataset was distributed as follows: a total of 500 pages from 100 writers, with every writer having 5 samples. The dataset JUDVLP-BLWVdb was distributed as follows: a total of 500 pages from 100 writers, with every writer having 5 samples. For training, non-overlapping 3 samples were selected (i.e., 100 writers x 3 samples), and the non-overlapping 2 samples were selected for testing (i.e., 100 writers x 2 other the non-overlapping samples). For verification work, randomly selected writers have the same number of sample image distributions done at the train and test phases. For block-level training data, each writer had 147 block images (i.e., 49 blocks x 3 samples), and randomly selected writers had the same number of block images. In the training phase, we utilized 29,400 block images (i.e., (147+147) x 100 writers) for experimentation. During the test phase, a total of 19,600 block images (i.e., (98+98) x 100 writers) were used for experimentation. Therefore, the distribution of page and block-level data was in a ratio of 3:2. The outline of the proposed method, as depicted in Figure 6.1, consists of the following phases: first, the collection of handwriting samples; second, the pre-processing of data; third, the extraction of features and the combination of these features; fourth, the use of a GA-based algorithm to obtain optimum features; and lastly, the description of classifier schemes. For this experiment setup, five samples of writer data into non-overlapping three for training samples and two for testing samples. In our previous work [38] we maintained a 3:2 division of training and testing samples over a block-level and page-level dataset. We have chose this ratio to allocate a larger portion of the data for training, ensuring sufficient data for learning complex patterns and improving the performance of the model. The remaining portion was then used for testing to evaluate the model's generalization and performance on unseen data. For pair writers, randomly selected non-overlapping writer data of the page level and the block level. A genetic Algorithm is used to reduce the dimensions of the features, which produces optimum numbers of selected features to enrich the results. This procedure is performed on both a block-level and page-level dataset. Ultimately, a classifier based on these features has been constructed using cost-sensitive learning with Simple logistic, SMO, and RBF networks.

### 6.1.2 Experiment Setup on Multilevel JUDVLP-BLWVdb dataset

In this experimental setup, depicted in Figure 6.2, we outline the framework of our proposed writer verification approach applied to the JUDVLP-BLWVdb dataset. The approach involves four primary steps: pre-processing of sample data, feature extraction, classification, and majority voting. The majority voting process is conducted at each multi-level of the document and across all multi-levels to arrive at a final decision. The initial step encompasses data collection and scanning at 300 dpi to generate digital text images with dimensions of 2481 x 3507. These images undergo transformation into multi-level data for analysis, including page, line, and word levels. The process initiates by converting a sample color data image to grayscale, followed by skew correction. Subsequently, the resulting image is binarized using Otsu's method [117]. To extract the minimum-area enclosing rectangle for page-level text image data, the binary text image data undergoes the minimum bounding box algorithm [118]. After the extraction of page-level data, line-level data is obtained from the page data image using a vertical projection profile. From these lines, word-level data images are further extracted using morphological operations and connected component analysis. In the second step,

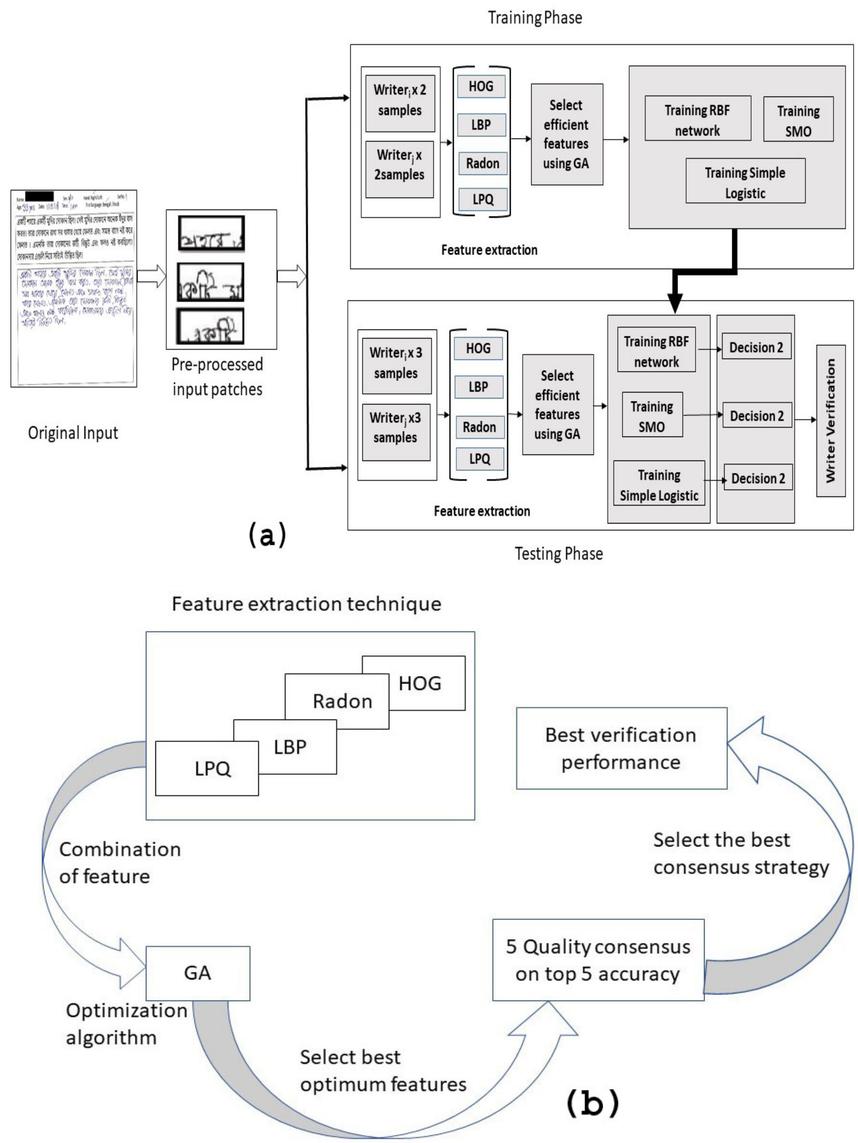



Figure 6.1: (a) The outline of the proposed writer verification framework and (b) is the sub-part of the figure (a)

textual features are generated for the data at the page, line, and word levels, with each level having its distinct feature vector. Proceeding to the third step, writer classification is performed utilizing Support Vector Machine (SVM), Multilayer Perceptron (MLP), and simple logistic regression to verify different levels of text, encompassing page, line, and word. The fourth and conclusive step involves a majority vote, where the verification predictions of writer samples at the page, line, and word levels are consolidated.

### 6.1.3 Experiment Setup on JUDVLP-TLWVdb dataset

In the final experiment, we utilized the newly developed JUDVLP-TLWVdb dataset. The dataset distribution for this experiment is as follows: a total of 443 pages, comprising 148 pages in Bangla, 147 pages in Hindi, and 151 pages in English, all from 31 different writers. The ratio of training to testing data in the JUDVLP-TLWVdb dataset is 3:1, and training and testing data samples do not overlap. Non-overlapping writer pairs are selected for both the training and testing data distribution. Figure 6.3 shows our proposed model. After the pre-processing steps, the text image is divided into 16x16 rows and columns, following the original ViT architecture, where the input image is divided into fixed-size patches. In total, 256 blocks are formed on every page. In the training set, an average of 2014 block-level data are considered, while in the test set, an average of 1236 block-level data are considered.

## 6.2 Experimental Results

### 6.2.1 Result on JUDVLP-BLWVdb dataset

To evaluate the effectiveness of this proposed method for text-dependent writer verification, compared the results with a features combination model. Table 6.2 summarizes the proposed model computational time and performances, which show that the SimpleLogistic classifier achieved an average performance of 89.63% and 78.66 % for 50 writer's block and page images, and 89.76% and 81.02% for 100 writer's block and page image respectively. However, the performance of the page image dataset was lower than that of the block image dataset, which could be due to the small size of the training and testing data. Therefore, the Genetic Algorithm for feature selection could not be applied to the page image dataset of JUDVLP-BLWVdb. The SMO classifier performed better than the SimpleLogistic and RBF network classifiers, and the accuracy increased as the number of writers increased for all classifiers.

To reduce computational time and memory space, reduce the dimensionality of the combined feature attributes. Table 6.3 summarizes the results, which show reduced the block image dataset features from a large dataset of size 753-D to 50% using a Genetic Algorithm. SMO achieved the best performance, with an average accuracy of 94.06% and 94.05% for 100 writers and 50 writers, respectively. SimpleLogistic and RBF achieved lower accuracies. The results also show the same trend in Figure 6.4. With an increase in the number of writers, the verification

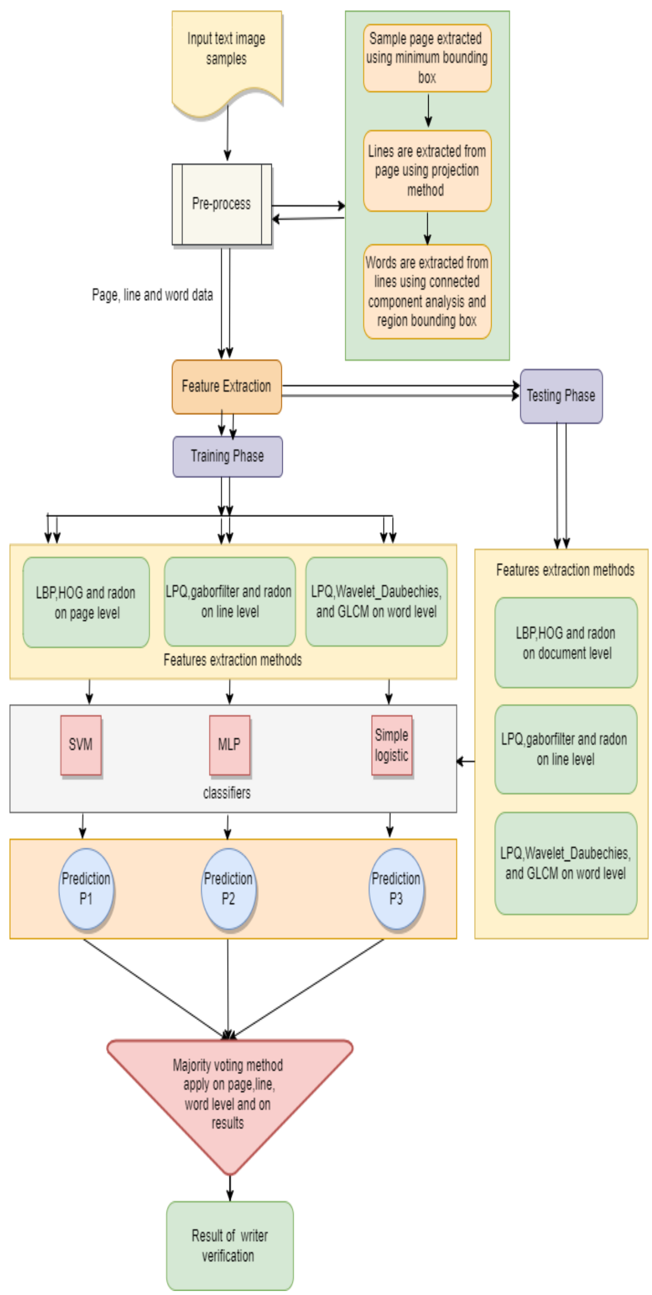



Figure 6.2: The Framework of our proposed verification approach

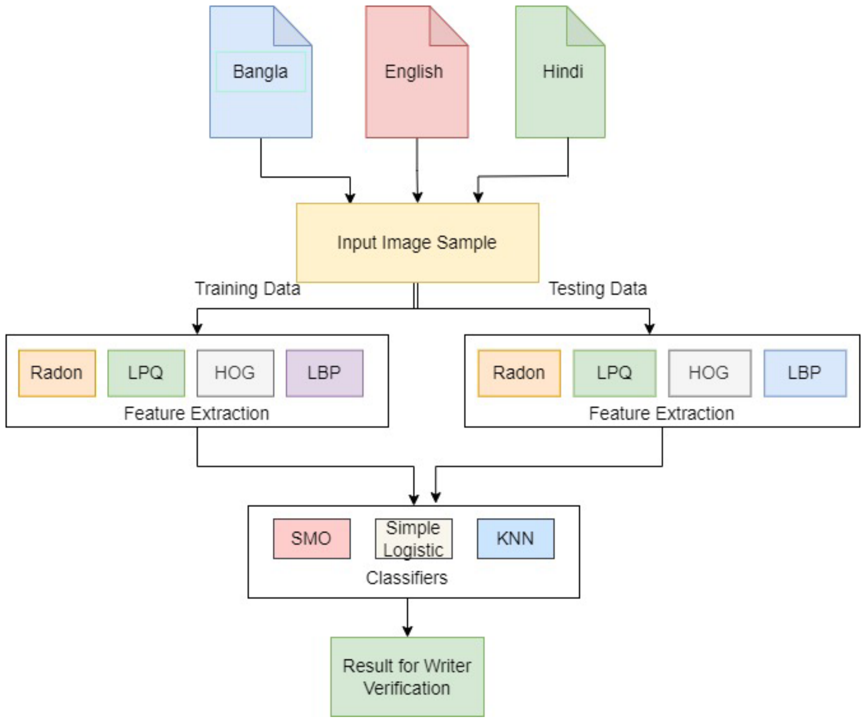



Figure 6.3: Block diagram for the proposed multi-script writer verification system framework

performance also increased. To obtain the best result, utilized the SMO classifier and employed a five quality consensus [144] by selecting the five optimal sets of features through five different runs of GA. The best five results were then obtained using the SMO classifier. Table 6.4 shows that the maximum writer verification accuracy of 94.54% with 724 features was achieved in the quality four consensus. Therefore, GA produces maximum writer verification performance using the SMO classifier. A single quality consensus generates a maximum verification rate of 94.54% for the best five runs using the SMO classifier only. In summary, evaluated auto-derived feature-based writer verification methods and obtained results on two types of samples: data block images and page images, as shown in Table 6.5. Used a learning rate of 0.001 for 100 epochs. The results for the block image JUDVLP-BLWVdb dataset in Table 6.5 were better than those for the page image JUDVLP-BLWVdb dataset. On average, the computation time of AlexNet at the block level is 11.66 hours with 100 epochs, while the computation time at the page level is, on average, 5 hours with 100 epochs Due to the limited number of handwritten samples per writer, the result did not improve using auto-derived features compared to handcraft features.

The results for various state-of-the-art methods and benchmark datasets are listed in Tables 6.6 and 6.7. Table 6.6 lists the writer verification performance when applied to different standard writer verification datasets. Table 6.7 shows the performances of various models on the JUDVLP-BLWVdb dataset. The scale-invariant feature transform was evaluated using GA dimensionality reduction, resulting in a reduction of less than 50% at both the block and page levels of the JUDVLP-BLWVdb dataset. The average accuracies obtained by using the scale-invariant feature transform were 60.03% and 57.33% at the block and page levels, respectively. Additionally, the graphemes feature was evaluated using GA dimensionality reduction at the page level of the JUDVLP-BLWVdb dataset, yielding an average accuracy of 58.65%. The computation time required for training the long short-term memory network at the block level was 9.83 h after 130 training epochs at a learning rate of 0.05 and the batch size of 128. On average, the long short-term memory network achieved an accuracy of 73.64%.

This performed *n*-fold cross-validation on the entire dataset. As in our data, we had five sets of samples per writer and performed from three to five folds using the SMO classifier. The SMO classifier was selected given its superiority over other classifiers, achieving 94.55% (94.37%) accuracy in five fold (four fold) cross-validation. Table 6.1 lists the results of *n*-fold cross-validation using the SMO classifier. The results were comparable to those obtained in our training and test model.

Table 6.1: Verification performance for *n*-fold cross-validation on entire dataset.

| No. folds for cross-validation | Precision | Recall | F-measure | Accuracy (%) |
|---|---|---|---|---|
| 3 | 94.10 | 94.06 | 94.05 | 93.08 |
| 4 | 94.40 | 94.38 | 94.38 | 94.37 |
| 5 | 94.57 | 94.60 | 94.56 | 94.55 |

**Statistical Significance Test**

The experiments evaluated two systems. Model 1 incorporated GA-based feature selection with the SOM classifier and achieved higher average results than the other two classifiers. Model 2, on the other hand, used the SOM classifier without GA-based feature selection. Both models represent the average results obtained from 100 writers. Applied these models on block image data and obtained accuracies

Table 6.2: Performance of block and page level image dataset

| Classifier Name | 50 writer accuracy (%) | | 100 writer accuracy (%) | | | |
|---|---|---|---|---|---|---|
| | Block image | Page image | Block image | Computation time (seconds) | Page image | Computation time (seconds) |
| Simple Logistic | **89.63** | 56.12 | 88.81 | 1.63 | 71.89 | 0.046 |
| SMO | 89.38 | **78.66** | **89.76** | 0.06 | **81.02** | 0.016 |
| RBF network | 78.84 | 68.22 | 78.38 | 6.25 | 71.89 | 0.027 |

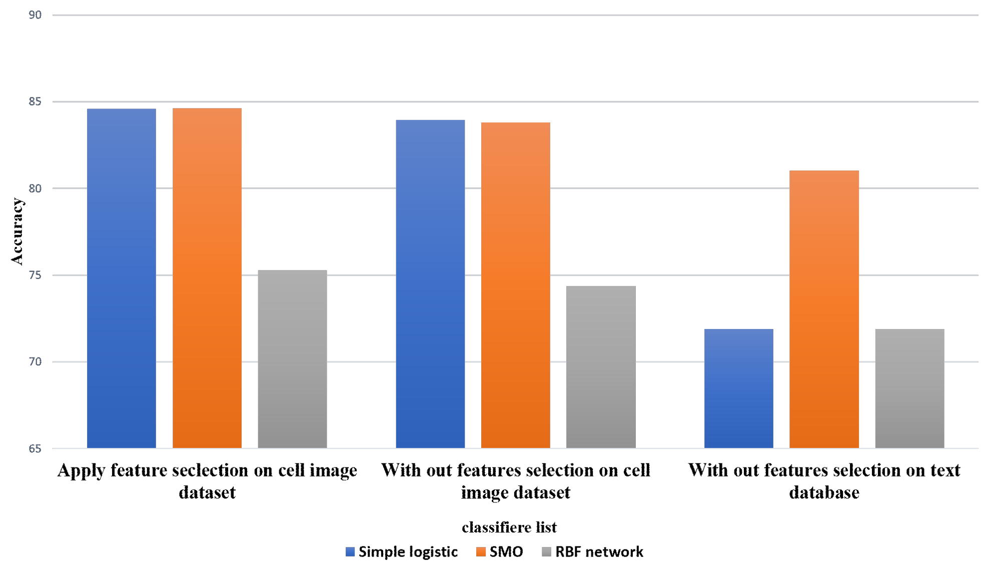



Figure 6.4: Comparison of two datasets, namely as the block and page image dataset results

Table 6.3: Performance of the block image dataset with GA method

| Classifier Name | 50 writer accuracy (%) | 100 writer accuracy (%) |
|---|---|---|
| SimpleLogistic | 93.56 | 93.71 |
| SMO | **94.05** | **94.06** |
| RBF network | 81.53 | 81.30 |

Table 6.4: 5 quality consensus result based on the best five experimental results of features selected by GA using SMO classifier

| N-Quality Consensus | Optimal No. of Features | 100 Writer Verification accuracy (%) |
|---|---|---|
| 1 Consensus | 724 | **94.54** |
| 2 Consensus | 165 | 91.69 |
| 3 Consensus | 92 | 90.53 |
| 4 Consensus | 49 | 88.19 |
| 5 Consensus | 22 | 79.94 |

Table 6.5: Auto derived feature based average writers verification results on the block image and text image JUDVLP-BLWVdb dataset

| Dataset Name | 50 writer accuracy(%) | 100 writer accuracy(%) | Computation time hours |
|---|---|---|---|
| Block image | **80.68** | **78.58** | 11.66 |
| Page image | 69.08 | 68.00 | 5 |

Table 6.6: Comparison chart of writer verification performance applied on different standard writer verification datasets

| Sl. | Dataset Name | Methods | Accuracy(%) |
|---|---|---|---|
| 1. | Halder et al. [145], 2014 | Proposed method | 89.85 |
| 2. | Brazilian Forensic Letter [146], 2011 | Proposed method | 93.26 |
| 3. | JUDVLP-BLWVdb | Proposed method | **94.54** |

Table 6.7: Writer verification performance of different methods applied on JUDVLP-BLWVdb dataset

| Sl. | Dataset Name | Methods | Accuracy (%) |
|---|---|---|---|
| 1. | JUDVLP-BLWVdb | Halder et al. method [147], 2016 | 63.68 |
| 2. | JUDVLP-BLWVdb | Hanusiak et al. method [146], 2011 | 80.44 |
| 3. | JUDVLP-BLWVdb | Aubin et al. method [35], 2022 | 85.29 |
| 4. | JUDVLP-BLWVdb | Proposed method | **94.54** |

of 84.61% and 83.81% for Model 1 and Model 2, respectively. To demonstrate the significance of the performance gain resulting from GA-based feature selection,

the McNemar test.

The estimated test value is greater than the critical value of 3.84 at a 95% confidence interval, therefore reject the null hypothesis and conclude that two methods have statistically significant differences in their performances. The differences in solutions between methods are said to be statistically significant at the 0.05 significance level.

### 6.2.2 Result on multilevel JUDVLP-BLWVdb dataset

In this experiment, feature vectors exhibit different dimensions for various levels of analysis. Here are the dimensions for each level:

Page level: Texture based feature has 236 dimensions, another texture based feature has 81 dimensions, and 180 dimensions. The total page-level feature vector $F_{page}$ is obtained by concatenating these three feature sets:

$$F_{page} = F_{page}.1 \cup F_{page}.2 \cup F_{page}.3 = 497 \text{ dimensions} \tag{6.1}$$

Line level: Texture based feature has 400 dimensions, shape based feature has 180 dimensions, and another texture based feature has 256 dimensions. The total word-level feature vector $F_{word}$ is obtained by concatenating these three feature sets:

$$F_{line} = F_{line}.1 \cup F_{line}.2 \cup F_{line}.3 = 836 \text{ dimensions} \tag{6.2}$$

Word level: Texture based feature has 5 dimensions, 256 dimensions, and 55 dimensions. The total word-level feature vector $F_{word}$ is obtained by concatenating these three feature sets:

$$F_{word} = F_{word}.1 \cup F_{word}.2 \cup F_{word}.3 = 316 \text{ dimensions} \tag{6.3}$$

Table 6.9 provides the distribution of samples in the multi-level Bangla script dataset (page, line, and word) used in this experiment. Table 6.8 presents a summary of the features, which are extracted feature vectors from multi-level Bangla script images along with their dimensions.

#### Assessment Procedure for Multi-Level Writer Verification Experiment

The initial division of all sample images of multi-level writer scripts follows a 3:2 training and testing ratio. For each writer's n samples, 2 samples are reserved for testing the model, while the remaining samples (n-2) are utilized for training the model. The verification accuracy is computed using the following formula (Equation 6.4):

Table 6.8: Feature set details along with their dimension

| Script level | Feature Description | Dimension |
|---|---|---|
| Page | Texture base | 236 |
| | Texture base | 81 |
| | Shape | 180 |
| Line | Texture base | 400 |
| | Shape | 180 |
| | Texture base | 256 |
| Word | Texture base | 5 |
| | Texture base | 256 |
| | Texture base | 55 |

Table 6.9: The distribution of samples for training and testing in the evaluation of multi-level Bangla script on the JUDVLP-BLWVdb dataset

| | Multi-Level of data | | |
|---|---|---|---|
| Sample Division | Page | Line | Word |
| Train | 291 | 4985 | 27400 |
| Test | 197 | 2896 | 18200 |

$$\text{Accuracy} = \frac{\text{Correctly Classified Multi-level Script Writers}}{\text{Total Multi-level Script Writers}} \times 100\% \tag{6.4}$$

**Performance of machine learners**

Evaluated this model on the JUDVLP-BLWVdb Bangla dataset at three levels: page, line, and word. To extract features at multiple levels, separately measured verification accuracy for each level. The summarized results can be found in Tables 6.10. Report the average verification accuracy (%) of 101 writers, where each writer is correctly verified against the question writers' pairs.

Figure 6.5 demonstrates that the MLP classifier achieved the highest average verification accuracies across all three levels: 85.97% for the page level, 93.23% for the line level, and 94.91% for the word level. This performance outperformed the SVM and Simple Logistic classifiers on the multi-level dataset. Moreover, observed a gradual increase in writer verification performance from the page level to the word level when using the SVM, MLP, and Simple Logistic classifiers, respectively.

Table 6.10: The verification accuracy at page, line, and word levels using SVM, MLP, and Simple Logistic classifiers for Bangla script on the JUDVLP-BLWVdb

| | Multi-level accuracy(%) | | |
|---|---|---|---|
| Classifier name | Page level | Line level | Word level |
| SVM | 81.02 | 92.42 | 94.62 |
| MLP | **85.97** | **93.23** | **94.91** |
| Simple logistic | 71.89 | 92.08 | 94.13 |

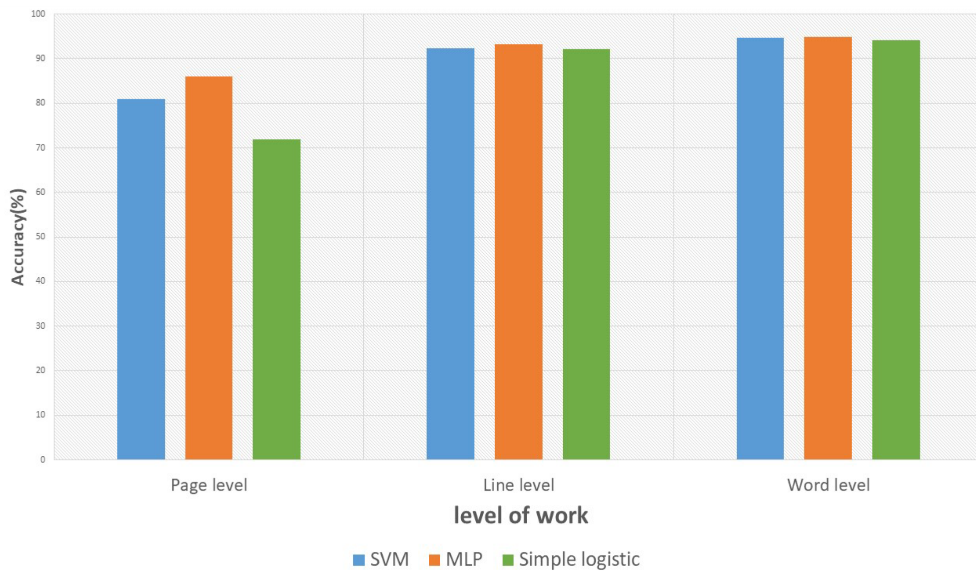



Figure 6.5: Comparative assessment of script classification performance across various levels (page, line, and word) employing SVM, MLP, and simple logistic classifiers

#### Effectiveness of Majority Voting in Script Performance Evaluation

In the initial phase of this experiment, the multi-level script verification tasks employed three classifiers: SVM, MLP, and Simple Logistic. In the subsequent experimental phase, these classifiers were integrated using a majority voting technique for multi-level script verification.

The majority voting ensemble is a method employed to enhance model performance, with the goal of achieving superior results compared to individual models. This chapter utilized three distinct classifiers, each contributing three classification rules denoted as $C_1(Y)$, $C_2(Y)$, and $C_3(Y)$. The combination of these three rules is orchestrated in a unified manner to generate a classifier that surpasses any of the individual rules within the trio.

$$M(X) = \text{mode}\{C_1(Y), C_2(Y), C_3(Y)\} \tag{6.5}$$

The majority vote classifier, determined by the highest number of votes, is defined as the aggregate of votes from rules $C_1, C_2, \ldots, C_n$.

$$M(X) = \arg\max_j \sum_{i=1}^{B} w_j I(h_i(X) = j) \tag{6.6}$$

In Table 6.11, the verification performance achieved using the majority voting scheme for the three levels of the JUDVLP-BLWVdb dataset is presented.

In pursuit of enhancing page-level writer verification performance, a novel method, not previously employed, was introduced. This approach involves leveraging the majority vote technique to amalgamate the probability estimations of a writer across page level, line level, and word level data. The aim was to elevate the page-level verification performance for the specific writer, as depicted in Figure 6.6.

In Table 6.11, the outcomes derived from amalgamating the majority voting results of the three levels (page, line, and word) and the two levels (line and word) are presented. This amalgamation led to an enhanced verification performance of 97.02%. These findings affirm our objective of improving the page-level writer verification performance.

Table 6.11: Performance results at various levels and combinations on Bangla script using majority voting scheme

| Verification Levels and Their Combined Configurations | Accuracy (%) |
|---|---|
| Page level | 84.15 |
| Line level | 94.18 |
| Word level | 95.34 |
| Line level+word level | **97.62** |
| Page level+line level+word level | 96.74 |

### 6.2.3 Performance Evaluation of Models Based on Deep Learning

In this study, our primary aim is to devise effective handcrafted features for comprehensive writer verification. To evaluate the efficacy of our designed verification model, we compared it with three prominent CNN-based models: AlexNet, ResNet34, and VGG16. Among these models, AlexNet exhibited superior performance, potentially attributed to its smaller size and the incorporation of ReLU activation functions. Given the relatively modest size of our dataset, a streamlined network like AlexNet proves more adept for the word-level task. The performance at the line level is also commendable, mirroring the word-level outcomes. However, concerning the page level, the inclusion of auto-derived features did not yield any improvement, likely due to the limited number of handwritten samples per writer.

To mitigate the risk of overfitting, we augmented our JUDVLP-BLWVdb dataset using a variety of techniques. These methods encompassed the introduction of 'salt and pepper' noise with a density of 0.01, Gaussian white noise with a variance of 0.01, Poisson noise with a mean of 10, 'localvar' noise with intensity values ranging from 0 to 1, and 'speckle' noise. Additionally, we horizontally flipped the images to further bolster the performance of deep learning models on multi-level datasets.

Moreover, we assessed the performance of auto-derived features, denoting features extracted automatically without manual intervention. The consolidated results of this evaluation are depicted in Table 6.12.

Table 6.12: Performance evaluation of writer verification using auto-derived features

| Model Name | Line level | Word level |
|---|---|---|
| Alexnet | 75.73 | **80.59** |
| Resnet34 | 56.89 | 77.78 |
| VGG16 | 56.10 | 62.75 |

### 6.2.4 Comparative Analysis with Established State-of-the-Art Feature Extraction Methods

Conducted an extensive comparison between our outcomes and existing methods utilizing the recently developed JUDVLP-BLWVdb dataset. Table 6.13 offers a synopsis of this comparative analysis, encompassing diverse methods employed for writer verification through various feature approaches.

Figure 6.7 visually demonstrates the superiority of our proposed method compared to others, particularly when employing the majority voting of page-level, line-level, and word-level data, leading to enhanced verification accuracy. However, it is essential to note that our dataset did not exhibit similar performance levels when evaluated using alternative methods.

Table 6.13: Comparative analysis with alternative methods on JUDVLP-BLWVdb dataset

| Dataset Name | Methods Name | Accuracy (%) |
|---|---|---|
| JUDVLP-BLWVdb | Halder et al. [147] method | 63.68 |
| JUDVLP-BLWVdb | Hanusiak et al. [146] method | 80.44 |
| JUDVLP-BLWVdb | Abdi and Khemakhem [35]method | 85.29 |
| JUDVLP-BLWVdb | Chawki et al. [148] method | 87.69 |
| JUDVLP-BLWVdb | Our proposed method | **97.62** |

### 6.2.5 Result on JUDVLP-TLWVdb dataset

Table 6.14 shows a comparison chart that illustrates the performance of this proposed method, which integrates hand-crafted feature sets, in terms of accuracy, precision, recall, and F-Measure. This experiment was conducted using newly proposed tri-scripts block-level JUDVLP-TLWVdb dataset. Table 6.14 presents the average performance of 31 writers, with the SMO classifier achieving the highest accuracy of 91.50%. Figure 6.8 demonstrates that the performance of the block-level JUDVLP-TLWVdb dataset surpasses that of the SimpleLogistic and KNN classifiers. Furthermore, Table 6.15 displays the performance of a single script which is 89-94%, indicating that the Hindi script outperforms Bangla and English. It is noteworthy that the tri-script overall performance is 91.80%.

Table 6.14: The tri-script block image dataset performance

| Classifier Name | Precision (%) | Recall(%) | F-Measure(%) | Accuracy(%) |
|---|---|---|---|---|
| SimpleLogistic | 86.27 | 90.53 | 90.46 | 90.52 |
| SMO | **91.78** | **91.50** | **91.42** | **91.50** |
| KNN | 74.63 | 73.14 | 73.20 | 73.08 |

In Table 6.14, it can be observed that the SMO classifier outperforms SimpleLogistic and KNN classifiers. The workflow of the writer verification methodology utilizing the Vision Transformer (ViT) model is depicted in Table 6.16. The

Table 6.15: The single-script block image dataset performance using SMO classifier

| Script Name | Precision (%) | Recall(%) | F-Measure(%) | Accuracy(%) |
|---|---|---|---|---|
| Bangla | 91.55 | 91.46 | 91.44 | 91.45 |
| English | 90.09 | 89.99 | 89.92 | 89.97 |
| Hindi | 93.92 | 93.89 | 93.89 | 94.00 |

Table 6.16: Performance based on vision Transformer of block level tri-script writer verification system task

| Model Name | Train Loss (%) | Test Loss(%) | Train Accuracy(%) | Testing Accuracy(%) |
|---|---|---|---|---|
| ViT-B | 6.528 | 1.969 | 95.46 | 74.80 |

JUDVLP-TLWVdb dataset has a total of 31 authors, each with specific training and testing sets. The training and testing ratio is set at 3:2. The verification model is capable of determining whether two multi-script text samples (Bangla, English, and Hindi), namely text1 and text2, belong to the same or different authors. This model achieves a multi-script verification accuracy of 74.80%.

Table 6.16 compares the combined handcrafted features model with the deep learning-based ViT model using the same JUDVLP-TLWVdb dataset. Due to the limited number of handwritten samples per writer, the performance does not improve significantly with auto-derived features compared to handcrafted features. The layout of the writer verification system that we have deployed is depicted in Figures 6.9 and 6.10.

## 6.3 Error Analysis

### 6.3.1 Page and Block Level of JUDVLP-BLWVdb dataset

In Figure 6.11, our analysis of the JUDVLP-BLWVdb dataset reveals a comprehensive examination of both page and block levels, showcasing instances of misclassification and accurate predictions. Notably, images originally attributed to Writer 1 were misclassified as belonging to Writer 2, underscoring challenges in distinguishing their writing styles. Conversely, images of Writer 3 and Writer 4 were correctly predicted as such by the SMO classifier. These findings illuminate the complexities of writer verification and provide valuable insights for future model refinement. Consideration of both the page and block levels is crucial for understanding misclassification intricacies and refining classification algorithms to enhance accuracy in diverse writing styles.

### 6.3.2 Multi-Level of JUDVLP-BLWVdb dataset

In Figure 6.12, a multi-level analysis of the JUDVLP-BLWVdb dataset is presented, revealing predicted images at the page, line, and word levels for various writers, highlighting instances of misclassification. Notably, images initially attributed to writer 1 were consistently misclassified as belonging to writer 2 at the page, line, and word levels, showcasing a systematic misclassification pattern by the MLP classifier. A significant observation arises from the preprocessing steps, particularly in line and word extraction from pages. Due to limitations in the extraction process, lines and words exhibit improper cutting, resulting in obscured visibility of handwriting. Consequently, the MLP classifier faces challenges in accurately predicting writers, as the true representation of handwriting is compromised during these preprocessing steps, impacting the model's predictive capabilities at both the line and word levels. These findings emphasize the critical role of preprocessing procedures and shed light on areas for improvement to enhance the overall accuracy of writer classification in multi-level analyses.

### 6.3.3 Block Level of of JUDVLP-TLWVdb dataset

In Figure 6.13, diverse errors in tri-script language predictions among various writers are showcased. Notably, Writer 20 achieves the lowest verification accuracy of 80.46%, specifically in Bengali script predictions. Similarly, Writer 23 obtains a relatively lower verification accuracy of 72.77%, particularly in English script predictions. Writer 25 exhibits a notable verification accuracy dip to 78.22% in Hindi script predictions. The most significant observation arises during multi-script combined block-level verification, where Writer 25's accuracy reaches its lowest point. This decline is attributed to the poor performance in predicting the writer accurately, emphasizing the challenges in achieving robust predictions in multi-script scenarios at the block level. These findings underscore the need for refined models and strategies, especially in scenarios involving multiple scripts, to enhance accuracy and reliability in writer prediction across various languages.

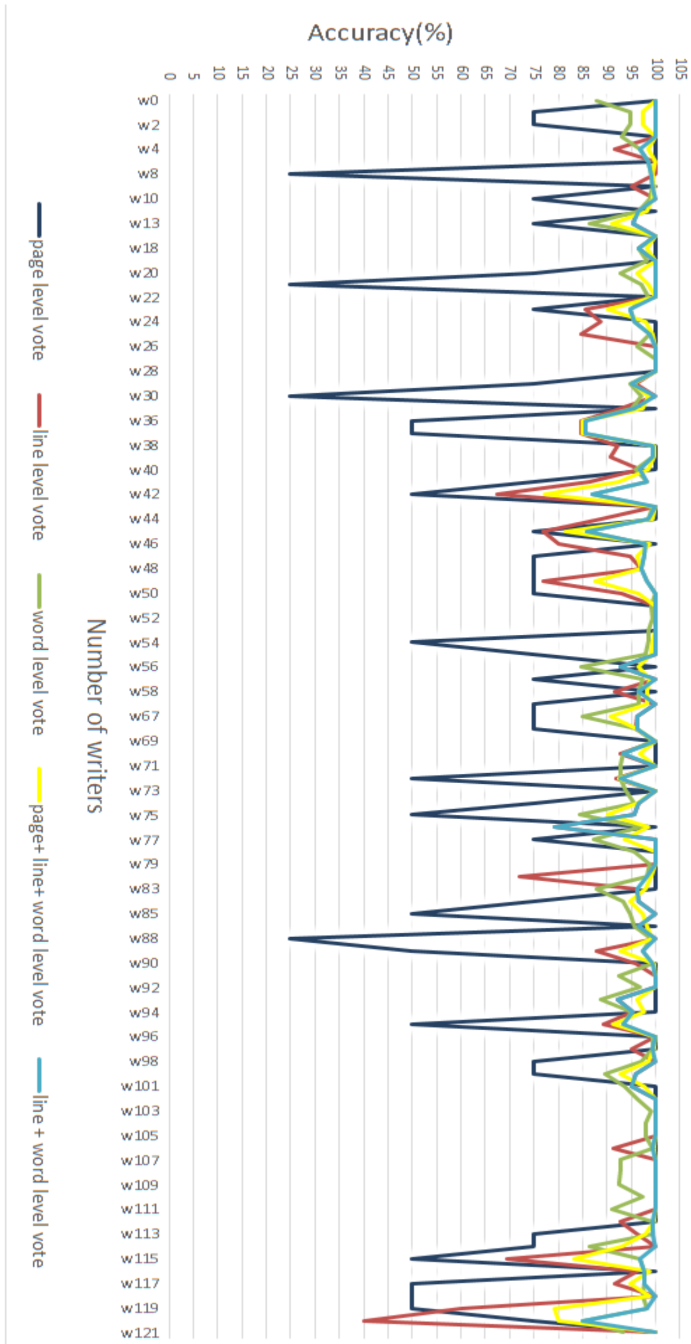



Figure 6.6: Writer verification performance on multi-level and their combination of the document

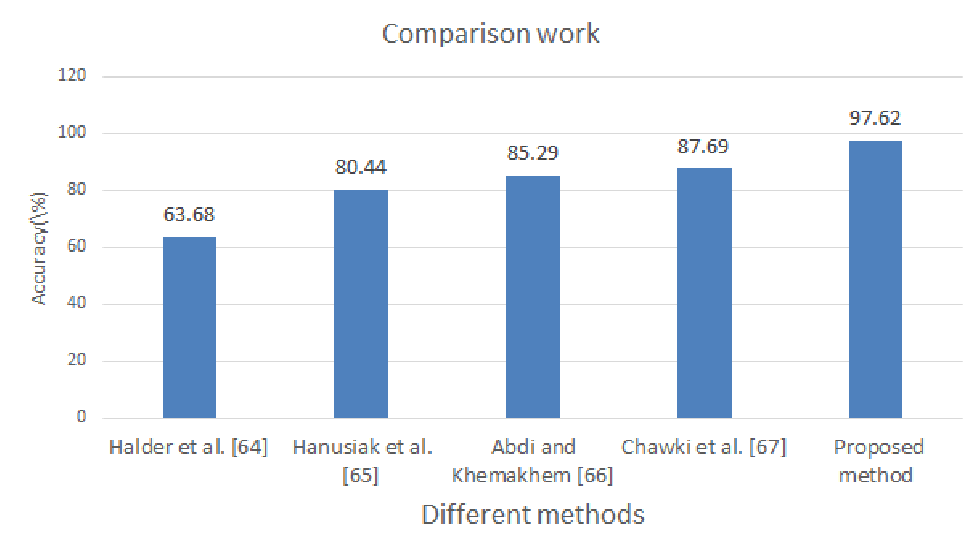



Figure 6.7: Comparative performance evaluation of the proposed writer verification method against other approaches using JUDVLP-BLWVdb dataset

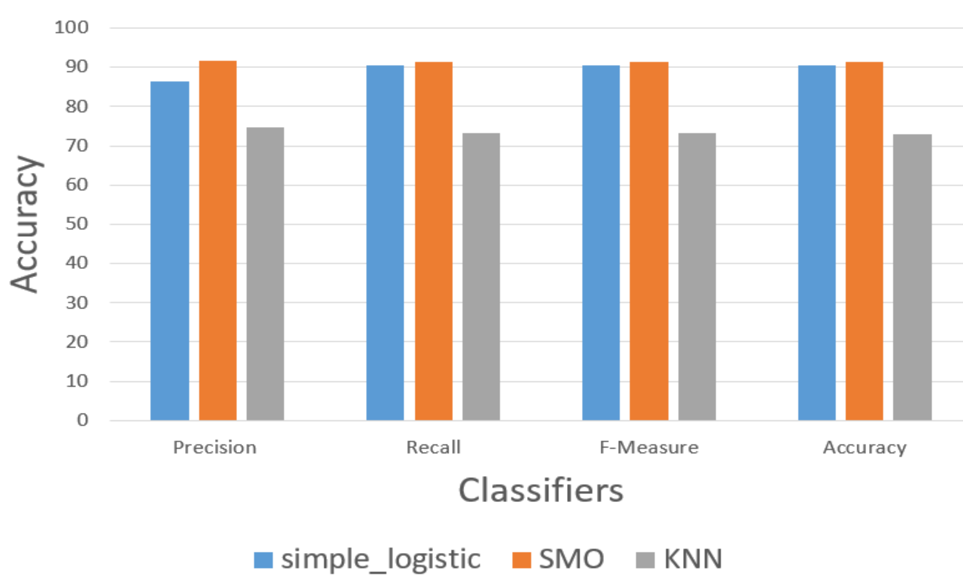



Figure 6.8: Comparison of different classifiers performance of Block level dataset

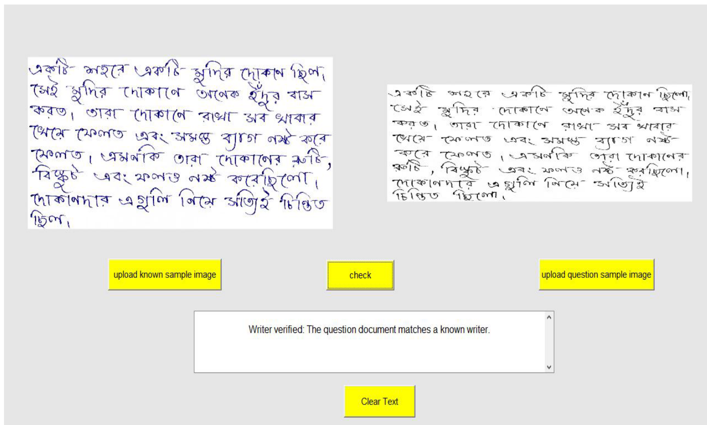



Figure 6.9: The known writer document and the question writer document are from the same writer.

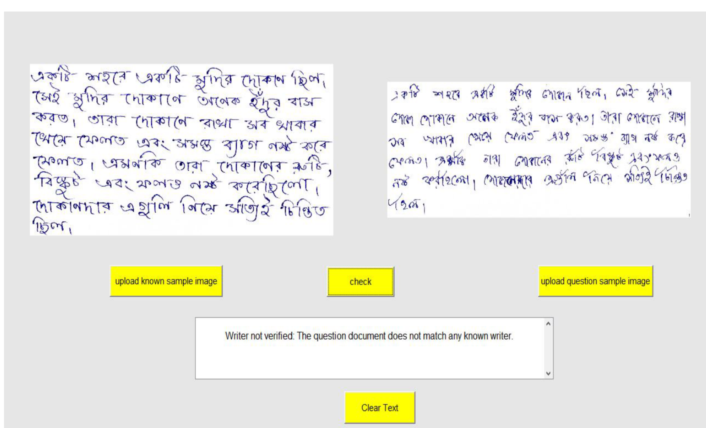
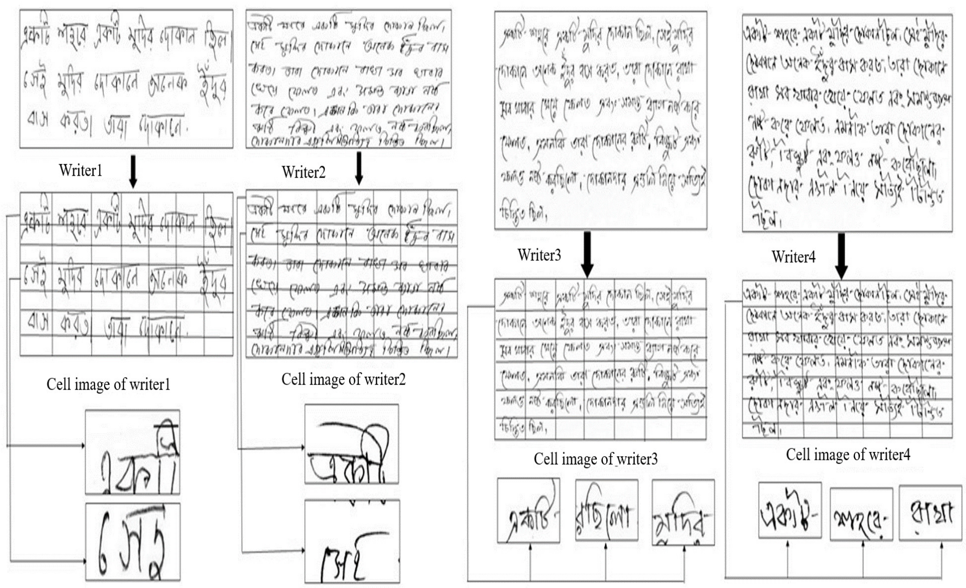



Figure 6.10: The known writer document and the question writer document are from different writers.




(a) Incorrectly classified

(b) Correctly classified

Figure 6.11: Few samples of correctly and incorrectly classified results using SMO classifier

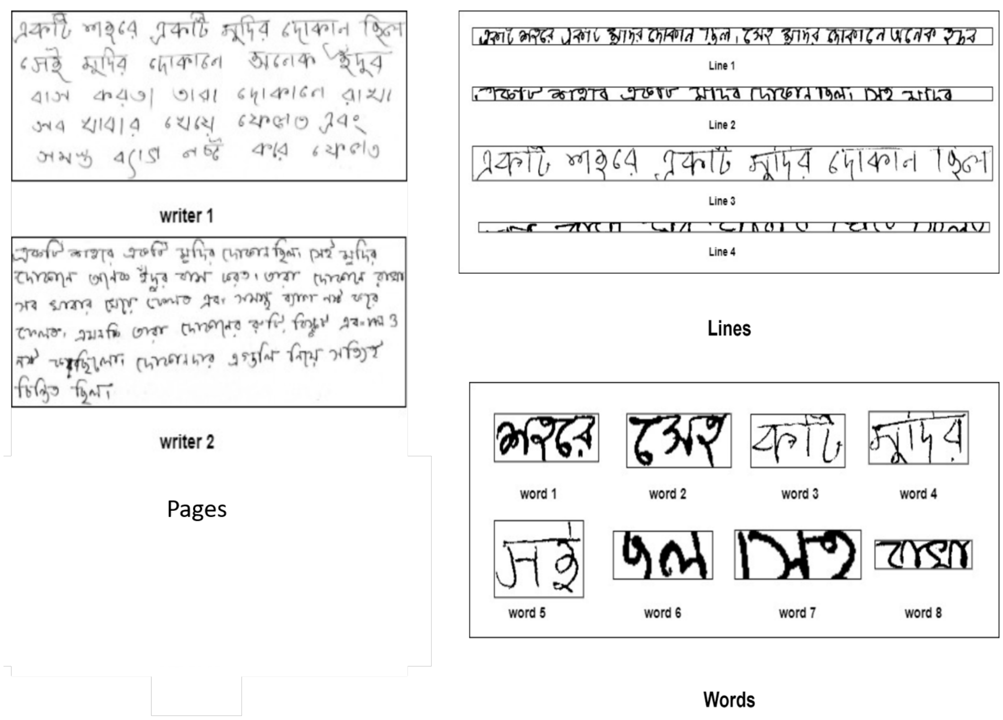



Figure 6.12: Multi-level handwriting Bangla script from different writers generate error

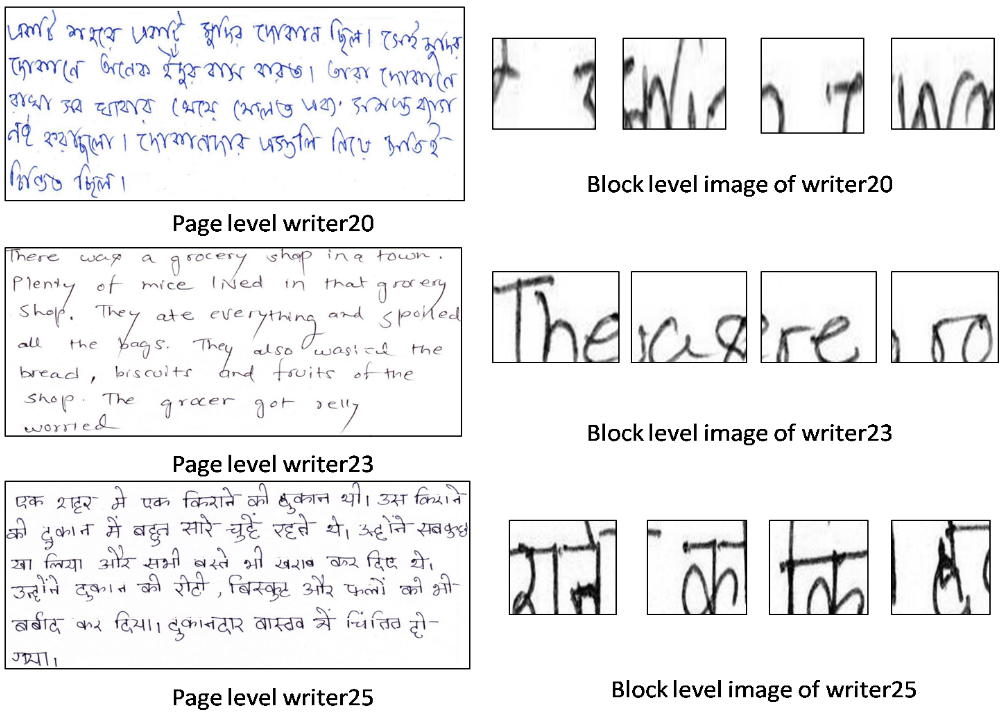



Figure 6.13: Multi-lingual tri-script handwriting Bangla script from different writers generate error

# 7 Conclusions and Scope for Future Research

In this thesis, the challenges and opportunities of writer verification and authentication for Bangla scripts were thoroughly explored and addressed. The main objective was to create a reliable and efficient system capable of accurately verifying and authenticating authors of handwritten Bangla documents.

Throughout the research, diverse methodologies, algorithms, and techniques in pattern recognition, machine learning, and image processing were investigated. Special attention was given to the distinct characteristics of Bangla scripts, which were leveraged to develop an effective writer verification system. The collection of a comprehensive dataset of handwritten samples from numerous writers facilitated extensive experiments and evaluations.

The findings highlighted that the proposed writer verification system achieved impressive accuracy and performance. This success was attributed to the integration of the advanced machine learning models and feature extraction techniques. The thesis also emphasized the significance of dataset quality in determining the overall performance of the verification system.

Moreover, the study's practical implications extended beyond writer verification, with potential applications in forensic document analysis, authorship attribution, and data security. Furthermore, the advancements made in this research offer a foundation for future studies on other Indic scripts, contributing to the improvement of writer verification systems in multiple languages.

Chapter 2 of the thesis conducts a comprehensive survey of offline and online methods for writer identification and verification across different languages and script levels. It evaluates various feature extraction strategies, including macro and micro level features, GST features, point and stroke-based features, and texture-based features, highlighting the superiority of writer identification features over verification features in terms of performance. The chapter showcases the efficacy of different feature extraction and classification algorithms for offline and online handwritten character recognition, writer identity, and verification. Noteworthy features like textual, allographic, local phase quantization, and local binary pattern prove particularly effective for author identification and verification. While recommending further exploration into different scripts and levels, such as line, word, and character segmentation, the chapter equips researchers and practitioners with valuable insights to address writer authentication challenges more effectively.

This research work focuses on two essential types of datasets, Public Dataset, and the Developed Dataset, specifically in the field of document analysis and recognition, with a particular emphasis on writer verification in chapter 3. The survey explores online and offline data acquisition methods, discussing their benefits and complexities. Several public datasets, such as IAM, Firemaker set, CEDAR, IFN/ENIT, AHDB/FTR, HCL2000, ETL1-ETL9, ISI, and CVL, are analyzed, playing a critical role in advancing writer verification research. To address the scarcity of publicly available Bangla script databases, this research introduces two new offline datasets: JUDVLP-BLWVdb and JUDVLP-TLWVdb. The former caters specifically to Bangla language writer verification with 101 native Bengali writers contributing 488 pages, categorized at multiple levels for comprehensive analysis. The latter dataset addresses India's multilingual context, containing samples from 31 writers proficient in Bengali, Hindi, and English, providing a valuable resource for tri-language writer verification. The datasets undergo data preprocessing, ensuring their suitability for further analysis by applying scanning, grayscale conversion, skew correction, and the minimum bounding box algorithm.

In Chapter 4, we present both handcrafted and auto-derived features that show promising results in page-level, block-level, line-level, and word-level writer verification. Handcrafted features encompass Radon Transform, HOG, LBP, GLCM, DWT, Gabor filter, and LPQ. To enhance the efficiency of the writer verification system, a feature selection procedure using Genetic Algorithm (GA) with SVM as the fitness function was implemented, resulting in a reduction of the feature set by approximately 50%. In this chapter, we also delve into auto-derived methods, featuring AlexNet, VGG16, and ResNet34. AlexNet gained significant attention as one of the early deep convolutional neural networks (CNNs), VGG16 is recognized for its simplicity and effectiveness, and ResNet34 introduced the concept of residual learning, addressing challenges such as vanishing gradients in very deep networks. These models have consistently demonstrated strong performance in image-related tasks, showcasing their effectiveness in learning hierarchical features.

The handcrafted feature combination exhibited greater robustness compared to automatically derived features, attributed to limitations in the dataset. Furthermore, in this study, we leverage ViT for writer recognition and achieve superior performance compared to ViT when using tri-level block images of the page. For the writer verification task, the original image is divided into 16 x 16 blocks. In processing, ViT also divides the input image into 16 x 16 blocks. Our analysis reveals that the proposed system, which employs manual block division, outperforms the ViT model.

In Chapter 5, the focus is on the machine learning technologies for the classification of whether the text is written by the same or different writers. The method is tested using the JUDVLP-BLWVdb dataset, extracting various levels of document features from Bangla handwritten pages, blocks, lines, and words of two writers. Different classifiers, including SVM, RBF network, KNN, MLP, SMO, and Simple Logistic, are employed for verification. Ensemble techniques, such as probabilistic voting, along with conventional feature combination approaches, are applied at the page, line, and word levels.

In Chapter 6, promising results are presented in addressing page-level, block-level, line-level, and word-level writer verification on the JUDVLP-BLWVdb dataset, as well as block-level tri-script verification on the JUDVLP-TLWVdb dataset. In the first experiment, the maximum verification accuracy achieved on the newly developed Bangla script dataset was an impressive 94.54%. SVM emerged as a superior classifier, outperforming commonly used classifiers like simple logistic regression and RBF network in the context of writer verification. These findings highlight the effectiveness of the proposed approach in advancing the field of page-level writer verification for Bangla scripts. In the second experiment, the methodology extracts features at different levels from handwritten Bangla pages, lines, and words of two writers. For writer pair verification, three classifiers (SVM, MLP, and Simple Logistic) are employed. Ensemble techniques, including probabilistic voting, and conventional approaches, such as feature combination, are applied on page-level, line-level, and word-level data. However, due to the limited sample size of each writer in the page-level data, the performance of our auto-crafted features does not match that of our proposed model. To enhance the dataset for deep learning processes, we augment data in our line-level and word-level datasets. In future work, we aim to experiment with synthetic datasets derived from the JUDVLP-BLWVdb dataset and explore various CNN architectures to further improve our deep learning evaluation.

In the third experiment, a dataset for a tri-script writer verification system is introduced, achieving a verification accuracy of 91.50% by combining Radon Transform, HOG, LBP, and LPQ features. The overall script-independent performance reaches 91.80%, consistent with the combined performance of all three scripts. Specifically, the Hindi script outperforms Bangla and English in individual script analysis.

The future prospects of this thesis involve continuous improvement and expansion of the proposed Bangla script writer verification system. By exploring advanced machine learning models and feature extraction techniques, the system's accuracy and efficiency can be enhanced. Furthermore, the research can extend to other Indic scripts and multilingual writer verification, making the system applicable to a wide range of languages. Additionally, exciting opportunities lie in forensic document analysis, privacy and security aspects, and integration with human-computer interaction systems. Creating benchmark datasets, addressing adversarial attacks, optimizing for mobile and embedded applications, and conducting user experience studies will contribute significantly to the field. Ultimately, this research aims to facilitate real-world deployment and offer impactful contributions to writer verification and authentication.

# Bibliography


[1] K. Roy, "On the development of an optical character recognition system for indian postal automation."

[2] C. De Stefano, F. Fontanella, D. Impedovo, G. Pirlo, and A. Scotto di Freca, "Handwriting analysis to support neurodegenerative diseases diagnosis: A review," *Pattern Recognition Letters*, vol. 121, pp. 37–45, 2019, graphonomics for e-citizens: e-health, e-society, e-education. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865518301880

[3] H. H. Harralson and L. S. Miller, *Huber and Headrick's handwriting identification: facts and fundamentals*. CRC press, 2017.

[4] K. Breckenridge, *Biometric state*. Cambridge University Press, 2014.

[5] M. G. Galterio, S. A. Shavit, and T. Hayajneh, "A review of facial biometrics security for smart devices," *Computers*, vol. 7, no. 3, p. 37, 2018.

[6] R. Srivastava, R. Tomar, A. Sharma, G. Dhiman, N. Chilamkurti, and B.-G. Kim, "Real-time multimodal biometric authentication of human using face feature analysis." *Computers, Materials & Continua*, vol. 69, no. 1, 2021.

[7] Y. Park, K. Park, K. Lee, H. Song, and Y. Park, "Security analysis and enhancements of an improved multi-factor biometric authentication scheme," *International Journal of Distributed Sensor Networks*, vol. 13, no. 8, p. 1550147717724308, 2017.

[8] P. Gainza, S. Wehrle, A. Van Hall-Beauvais, A. Marchand, A. Scheck, Z. Harteveld, S. Buckley, D. Ni, S. Tan, F. Sverrisson *et al.*, "De novo design of protein interactions with learned surface fingerprints," *Nature*, pp. 1–9, 2023.

[9] S. Q. Smeele, J. C. Senar, L. M. Aplin, and M. B. McElreath, "Evidence for vocal signatures and voice-prints in a wild parrot," *bioRxiv*, pp. 2023–01, 2023.

[10] Z. Shen, S. Li, X. Zhao, and J. Zou, "Increauth: Incremental learning based behavioral biometric authentication on smartphones," *IEEE Internet of Things Journal*, 2023.

[11] H. Kaur and M. Kumar, "Signature identification and verification techniques: state-of-the-art work," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 2, pp. 1027–1045, 2023.

[12] M. A. M. Hasan, J. Shin, and M. Maniruzzaman, "Online kanji characters based writer identification using sequential forward floating selection and support vector machine," *Applied Sciences*, vol. 12, no. 20, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/20/10249

[13] A. Bensefia and T. Paquet, "Writer verification based on a single handwriting word samples," *EURASIP Journal on Image and Video Processing*, vol. 2016, no. 1, p. 34, 2016.

[14] A. Al-Dhaqm, S. Abd Razak, R. A. Ikuesan, V. R. Kebande, and K. Siddique, "A review of mobile forensic investigation process models," *IEEE access*, vol. 8, pp. 173 359–173 375, 2020.

[15] S. Raaijmakers, "Artificial intelligence for law enforcement: challenges and opportunities," *IEEE security & privacy*, vol. 17, no. 5, pp. 74–77, 2019.

[16] A. Bodepudi and M. Reddy, "Cloud-based biometric authentication techniques for secure financial transactions: A review," *International Journal of Information and Cybersecurity*, vol. 4, no. 1, pp. 1–18, 2020.

[17] S. N. Srihari and G. R. Ball, "Comparison of statistical models for writer verification," *Proc. SPIE*, vol. 7247, p. 8, 2009.

[18] A. Gattal, C. Djeddi, F. Abbas, I. Siddiqi, and B. Bouderah, "A new method for writer identification based on historical documents," *Journal of Intelligent Systems*, vol. 32, no. 1, p. 20220244, 2023.

[19] B. Q. Ahmed, Y. F. Hassan, and A. S. Elsayed, "Offline text-independent writer identification using a codebook with structural features," *Plos one*, vol. 18, no. 4, p. e0284680, 2023.

[20] R. Johinke, R. Cummings, and F. Di Lauro, "Reclaiming the technology of higher education for teaching digital writing in a post—pandemic world," *Journal of University Teaching & Learning Practice*, vol. 20, no. 2, p. 01, 2023.

[21] P. Bhowal, D. Banerjee, S. Malakar, and R. Sarkar, "A two-tier ensemble approach for writer dependent online signature verification," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 1, pp. 21–40, Jan 2022. [Online]. Available: https://doi.org/10.1007/s12652-020-02872-5

[22] E. Zois and V. Anastassopoulos, "Morphological waveform coding for writer identification," *Pattern Recognition*, vol. 33, no. 3, pp. 385 – 398, 2000.

[23] A. Bensefia, A. Nosary, T. Paquet, and L. Heutte, "Writer identification by writer's invariants," in *Proceedings of International Workshop on Frontiers in Handwriting Recognition*, 2002, pp. 274–279.

[24] F. E. Batool, M. Attique, M. Sharif, K. Javed, M. Nazir, A. A. Abbasi, Z. Iqbal, and N. Riaz, "Offline signature verification system: a novel technique of fusion of glcm and geometric features using svm," *Multimedia Tools and Applications*, 2020. [Online]. Available: https://doi.org/10.1007/s11042-020-08851-4

[25] G. Abdeljalil, C. Djeddi, I. Siddiqi, and S. Al-Maadeed, "Writer identification on historical documents using oriented basic image features," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2018, pp. 369–373.

[26] C. Adak, B. B. Chaudhuri, and M. Blumenstein, "An empirical study on writer identification and verification from intra-variable individual handwriting," *IEEE Access*, vol. 7, pp. 24 738–24 758, 2019.

[27] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on iot security: Application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82 721–82 743, 2019.

[28] Y. Kuan, A. Goh, C. Ngo, and B. Teoh, "Cryptographic keys from dynamic hand-signatures with biometric secrecy preservation and replaceability," in *Proceedings of International Workshop on Automatic Identification Advanced Technologies*, 2005, pp. 27–32.

[29] S. Awaida and S. Mahmoud, "State of the art in off-line writer identification of handwritten text and survey of writer identification of arabic text," *Educational Research and Reviews*, vol. 7, 2012.

[30] R. Welekar and V. S. D. Rao, "Survey on existing techniques for writer verification," *COMPUSOFT: An International Journal of Advanced Computer Technology*, vol. 3, no. 5, 2015.

[31] H. Girdher and H. Sharma, "A survey on writer identification system for indic scripts," *Available at SSRN 3538594*, 2020.

[32] C. Halder, S. Obaidullah, K. Roy *et al.*, "Offline writer identification and verification—a state-of-the-art," in *Information systems design and intelligent applications*. Springer, 2016, pp. 153–163.

[33] P. K. Singh, R. Sarkar, and M. Nasipuri, "A comprehensive survey on bangla handwritten numeral recognition," *International Journal of Applied Pattern Recognition*, vol. 5, no. 1, pp. 55–71, 2018.

[34] V. Aubin, M. Mora, and M. Santos Peñas, "Off-line writer verification based on simple graphemes," *Pattern Recognition*, vol. 79, pp. 414–426, 02 2018.

[35] M. N. Abdi and M. Khemakhem, "A model-based approach to offline text-independent arabic writer identification and verification," *Pattern Recognition*, vol. 48, no. 5, pp. 1890 – 1903, 2015.

[36] S. Fiel and R. Sablatnig, "Writer identification and retrieval using a convolutional neural network," in *Proceedings of International Conference on Computer Analysis of Images and Patterns*, 2015, pp. 26–37.

[37] W. Yang, L. Jin, and M. Liu, "Chinese character-level writer identification using path signature feature, dropstroke and deep cnn," in *International Conference on Document Analysis and Recognition*. IEEE, 2015, pp. 546–550.

[38] J. Paul, A. Sarkar, N. Das, and K. Roy, "Hog and lbp based writer verification," in *Proceedings of International Conference on Frontiers in Computing and Systems*, 2021, pp. 3–12.

[39] J. Paul and A. Sarkar, "Handwritten bangla numeral recognition using convolutional neural networks," in *Proceedings of International Conference on Electronics, Materials Engineering Nano-Technology*, 2018, pp. 1–4.

[40] T. Bahram, "A texture-based approach for offline writer identification," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, Part A, pp. 5204–5222, 2022.

[41] Y. Hannad, I. Siddiqi, and M. E. Y. E. Kettani, *Arabic Writer Identification Using Local Binary Patterns (LBP) of Handwritten Fragments*. Springer International, 2015, pp. 237–244.

[42] 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, 2007.

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

[44] S. M. Obaidullah, N. Das, and R. Kaushik, "Convolution based technique for indic script identification from handwritten document images," *International Journal of Image, Graphics and Signal Processing*, vol. 7, no. 5, p. 49, 2015.

[45] V. Venugopal and S. Sundaram, "Online writer identification with sparse coding-based descriptors," *Information Forensics and Security*, vol. 13, no. 10, pp. 2538–2552, 2018.

[46] A. Al-Dmour and R. A. Zitar, "Arabic writer identification based on hybrid spectral statistical measures," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 19, no. 4, pp. 307–332, 2007.

[47] M. Y. Tsai and L. S. Lan, "Online writer identification using the point distribution model," in *Proceedings of International Conference on Systems, Man and Cybernetics*, vol. 2, 2005, pp. 1264–1268.

[48] H. Kameya, S. Mori, and R. Oka, "Figure-based writer verification by matching between an arbitrary part of registered sequence and an input sequence extracted from on-line handwritten figures," in *Proceedings of International Conference on Document Analysis and Recognition*, 2003, pp. 985–989.

[49] W. Yang, L. Jin, and M. Liu, "Deepwriterid: An end-to-end online text-independent writer identification system," *IEEE Intelligent Systems*, vol. 31, no. 2, pp. 45–53, 2016.

[50] 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, 2004.

[51] A. Seropian, M. Grimaldi, and N. Vincent, "Writer identification based on the fractal construction of a reference base," in *Proceedings of International Conference on Document Analysis and Recognition*, 2003, pp. 1163–1167.

[52] S. Al-Maadeed, A. Hassaine, A. Bouridane, and M. A. Tahir, "Novel geometric features for off-line writer identification," *Pattern Analysis and Applications*, vol. 19, no. 3, pp. 699–708, Aug 2016.

[53] W. Jin, Y. Wang, and T. Tan, *Text-Independent Writer Identification Based on Fusion of Dynamic and Static Features*. Springer Berlin Heidelberg, 2005, pp. 197–204.

[54] A. Schlapbach, M. Liwicki, and H. Bunke, "A writer identification system for on-line whiteboard data," *Pattern Recognition*, vol. 41, no. 7, pp. 2381 – 2397, 2008.

[55] A. Schlapbach and H. Bunke, "A writer identification and verification system using HMM based recognizers," *Pattern Analysis and Applications*, vol. 10, no. 1, pp. 33–43, 2007.

[56] B. B. Chaudhuri and S. Bera, "Handwritten text line identification in indian scripts," in *International Conference on Document Analysis and Recognition*. IEEE, 2009, pp. 636–640.

[57] S. Goyal and A. K. Bathla, "Method for Line Segmentation in Handwritten Documents with Touching and Broken Parts in Devanagari Script," *International Journal of Computer Applications*, vol. 102, no. 12, pp. 22–27, 2014.

[58] A. Sulaiman, K. Omar, M. F. Nasrudin, and A. Arram, "Length Independent Writer Identification Based on the Fusion of Deep and Hand-Crafted Descriptors," *IEEE Access*, vol. 7, pp. 91 772–91 784, 2019.

[59] J. Chapran, "Biometric writer identification: feature analysis and classification," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 20, no. 04, pp. 483–503, 2006.

[60] B. Zhang and S. N. Srihari, "Analysis of handwriting individuality using word features," in *Conference on Document Analysis and Recognition*, 2003, pp. 1142–.

[61] P. Zhang, "Rstc: a new residual swin transformer for offline word-level writer identification," *IEEE Access*, vol. 10, pp. 57 452–57 460, 2022.

[62] H. Kaur, S. Bansal, M. Kumar, A. Mittal, and K. Kumar, "Worddeepnet: handwritten gurumukhi word recognition using convolutional neural network," *Multimedia Tools and Applications*, 2023. [Online]. Available: https://doi.org/10.1007/s11042-023-15527-2

[63] U. Pal, R. K. Roy, and F. Kimura, "Handwritten Street Name Recognition for Indian Postal Automation," in *Proceedings of International Conference on Document Analysis and Recognition*, 2011, pp. 483–487.

[64] U. Pal and R. K. Roy and F. Kimura, "Multi-lingual City Name Recognition for Indian Postal Automation," in *Proceedings of International Conference on Frontiers in Handwriting Recognition*, 2012, pp. 169–173.

[65] T. K. Bhowmik, U. Roy, and S. K. Parui, "Lexicon Reduction Technique for Bangla Handwritten Word Recognition," in *Proceedings of International Conference on Document Analysis Systems*, 2012, pp. 195–199.

[66] C. Halder and K. Roy, "Word & character segmentation for bangla handwriting analysis & recognition," in *Proceedings of International Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*, 2011, pp. 243–246.

[67] S. Chanda, S. Pal, K. Franke, and U. Pal, "Two-stage Approach for Word-wise Script Identification," in *Proceedings of International Conference on Document Analysis and Recognition*, 2009, pp. 926–930.

[68] M. Hangarge, C. Veershetty, R. Pardeshi, and B. Dhandra, “Gabor wavelets based word retrieval from kannada documents,” *Procedia Computer Science*, vol. 79, no. 3, pp. 441–448, 2016.

[69] M. Gargouri, S. Kanoun, and J. M. Ogier, “Text-Independent Writer Identification on Online Arabic Handwriting,” in *Proceedings of International Conference on Document Analysis and Recognition*, 2013, pp. 428–432.

[70] G. A. Fink, S. Vajda, U. Bhattacharya, S. K. Parui, and B. B. Chaudhuri, “Online Bangla Word Recognition Using Sub-Stroke Level Features and Hidden Markov Models,” in *Proceedings of International Conference on Frontiers in Handwriting Recognition*, 2010, pp. 393–398.

[71] V. Aubin, M. Mora, and M. Santos, “A new approach for writer verification based on segments of handwritten graphemes,” *Logic Journal of the IGPL*, vol. 30, no. 6, pp. 965–978, 2022.

[72] V. Pervouchine and G. Leedham, “Extraction and analysis of forensic document examiner features used for writer identification,” *Pattern Recognition*, vol. 40, no. 3, pp. 1004 – 1013, 2007.

[73] Pervouchine, V. and Leedham, G., “Extraction and Analysis of Document Examiner Features from Vector Skeletons of Grapheme ‘th’,” in *International Conference on Document Analysis Systems VII*. Springer, 2006, pp. 196–207.

[74] M. Okawa and K. Yoshida, “User generic model for writer verification using multiband image scanner,” in *Proceedings of International Conference on Technologies for Homeland Security*, 2013, pp. 375–380.

[75] X. Li and X. Ding, “Improving semi-text-independent method of writer verification using difference vector,” in *Proceedings of International Conference on Document Recognition and Retrieval XVI*, vol. 7247. SPIE, 2009, pp. 233–240.

[76] C. Halder, J. Paul, and K. Roy, “Individuality of bangla numerals,” in *International Conference on Intelligent Systems Design and Applications*. IEEE, 2012.

[77] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The weka data mining software: An update,” *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, 2009.

[78] C. Halder, S. M.Obaidullah, J. Paul, and K. Roy, “Writer Verification on Bangla Handwritten Characters,” in *Proceedings of International Conference on Advanced Computing and Systems for Security*. Springer, 2015, pp. 53–68.

[79] C. Halder, K. Thakur, S. Phadikar, and K. Roy, “Writer Identification from Handwritten Devanagari Script,” in *Proceedings of International Conference on Information Systems Design and Intelligent Applications*, 2015, pp. 497–505.

[80] Y. Nakamura and M. Kidode, “Individuality analysis of online Kanji handwriting,” in *Proceedings of International Conference on Document Analysis and Recognition*, 2005, pp. 620–624.

[81] S. Sen, M. Mitra, S. Chowdhury, R. Sarkar, and K. Roy, “Quad-Tree Based Image Segmentation and Feature Extraction to Recognize Online Handwritten Bangla Characters,” in *Proceedings of International Conference on Artificial Neural Networks in Pattern Recognition*. Springer, 2016, pp. 246–256.

[82] A. Namboodiri and S. Gupta, “Text Independent Writer Identification from Online Handwriting,” in *Proceedings of International Workshop on Frontiers in Handwriting Recognition*. Suvisoft, 2006.

[83] V. Kumar and S. Sundaram, “Utilization of information from cnn feature maps for offline word-level writer identification,” *Available at SSRN 4329693*, 2022.

[84] C. Adak, S. Marinai, B. B. Chaudhuri, and M. Blumenstein, “Offline bengali writer verification by pdf-cnn and siamese net,” in *International Workshop on Document Analysis Systems (DAS)*. IEEE, 2018, pp. 381–386.

[85] S. N. Srihari, C. I. Tomai, B. Zhang, and S. Lee, “Individuality of numerals,” in *Proceedings of International Conference on Document Analysis and Recognition*, 2003, pp. 1096–1100.

[86] K. Tselios, E. N. Zois, A. Nassiopoulos, S. Karabetsos, and G. Economou, “Automated Off-Line Writer Verification Using Short Sentences and Grid Features,” in *Proceedings of International Conference on Automated Forensic Handwriting Analysis*, 2011.

[87] S. N. Srihari, S. H. Cha, H. Arora, and S. Lee, “Individuality of handwriting,” *Journal of forensic science*, vol. 47, pp. 856–72, 2002.

[88] 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, 2013.

[89] S. Kore and S. Apte, “Writer Verification Using Spatial Domain Features under Different Ink Width Conditions,” *Computing Science and Engineering*, vol. 10, pp. 39–50, 2016.

[90] C. Adak, B. B. Chaudhuri, and M. Blumenstein, "An empirical study on writer identification and verification from intra-variable individual handwriting," *IEEE Access*, vol. 7, no. 7, pp. 24 738–24 758, 2019.

[91] A. Bensefia, T. Paquet, and L. Heutte, "A writer identification and verification system," *Pattern Recognition Letters*, vol. 26, pp. 2080–2092, 2005.

[92] U.-V. Marti and H. Bunke, "The iam-database: an english sentence database for offline handwriting recognition," *International Journal on Document Analysis and Recognition*, vol. 5, no. 1, pp. 39–46, 2002.

[93] S. He and L. Schomaker, "Fragnet: Writer identification using deep fragment networks," *IEEE Transactions on Information Forensics and Security*, vol. 15, no. 3, pp. 3013–3022, 2020.

[94] S. Singh and M. Hewitt, "Cursive digit and character recognition in cedar database," in *Proceedings of International Conference on Pattern Recognition*, vol. 2, 2000, pp. 569–572 vol.2.

[95] G. R. Ball, H. Kasiviswanathan, S. N. Srihari, and A. Narayanan, "Analysis of line structure in handwritten documents using the hough transform," *Proc. SPIE*, vol. 7534, p. 9, 2010.

[96] R. El-Hajj, L. Likforman-Sulem, and C. Mokbel, "Arabic handwriting recognition using baseline dependant features and hidden markov modeling," in *Proceedings of International Conference on Document Analysis and Recognition*, 2005, pp. 893–897.

[97] M. Pechwitz, S. S. Maddouri, V. Märgner, N. Ellouze, and H. Amiri, "IFN\ENIT - database of handwritten Arabic words," in *Proceedings of International Conference on CIFED*, 2002, pp. 129–136.

[98] J. Ramdan, K. Omar, M. Faidzul, and A. Mady, "Arabic handwriting data base for text recognition," *Procedia Technology*, vol. 11, pp. 580–584, 2013.

[99] H. Zhang, J. Guo, G. Chen, and C. Li, "HCL2000-A large-scale handwritten Chinese character database for handwritten character recognition," in *Proceedings of International Conference on Document Analysis and Recognition*, 2009, pp. 286–290.

[100] Y. H. S. Taiichi and Y. Kazuhiko, "On the data base ETL9 of handprinted characters in JIS Chinese characters and its analysis," *Trans. IEICE*, vol. 68, no. 4, pp. 757–764, 1985.

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

[102] F. Kleber, S. Fiel, M. Diem, and R. Sablatnig, "CVL-Database: An Off-Line Database for Writer Retrieval, Writer Identification and Word Spotting," in *Proceedings of International Conference on Document Analysis and Recognition*, 2013, pp. 560–564.

[103] Y. Tang and X. Wu, "Text-independent writer identification via cnn features and joint bayesian," in *Proceedings of International Conference on Frontiers in Handwriting Recognition*, 2016, pp. 566–571.

[104] L. Xing and Y. Qiao, "Deepwriter: A multi-stream deep cnn for text-independent writer identification," in *Proceedings of International Conference on Frontiers in Handwriting Recognition*, 2016, pp. 584–589.

[105] H. T. Nguyen, C. T. Nguyen, T. Ino, B. Indurkhya, and M. Nakagawa, "Text-independent writer identification using convolutional neural network," *CoRR*, vol. abs/2009.04877, 2020.

[106] C. L. Liu, F. Yin, D. H. Wang, and Q. F. Wang, "CASIA Online and Offline Chinese Handwriting Databases," in *Proceedings of International Conference on Document Analysis and Recognition*, 2011, pp. 37–41.

[107] T. K. Bhowmik, U. Bhattacharya, and S. K. Parui, "Recognition of Bangla Handwritten Characters Using an MLP Classifier Based on Stroke Features," in *Proceedings of International Conference on Neural Information Processing*, N. R. Pal, N. Kasabov, R. K. Mudi, S. Pal, and S. K. Parui, Eds. Springer Berlin Heidelberg, 2004, pp. 814–819.

[108] R. Sarkar, N. Das, S. Basu, M. Kundu, M. Nasipuri, and D. K. Basu, "CMATERdb1: a database of unconstrained handwritten bangla and bangla–english mixed script document image," *document analysis and recognition*, vol. 15, no. 1, pp. 71–83, 2012.

[109] S. Bhowmik, S. Malakar, R. Sarkar, S. Basu, M. Kundu, and M. Nasipuri, "Off-line bangla handwritten word recognition: a holistic approach," *Neural Computing and Applications*, vol. 31, pp. 5783–5798, 2019.

[110] S. Mozaffari, H. El Abed, V. Märgner, K. Faez, and A. Amirshahi, "Ifn/farsi-database: a database of farsi handwritten city names," in *International Conference on Frontiers in Handwriting Recognition*, 2008.

[111] C. Viard-Gaudin, P. M. Lallican, S. Knerr, and P. Binter, "The IRESTE On/Off (IRONOFF) dual handwriting database," in *Proceedings of International Conference on Document Analysis and Recognition*, 1999, pp. 455–458.

[112] D. H. Kim, Y. S. Hwang, S. T. Park, E. J. Kim, S. H. Paek, and S. Y. Bang, "Handwritten Korean character image database PE92," in *Proceedings of International Conference on Document Analysis and Recognition*, 1993, pp. 470–473.

[113] V. Christlein, D. Bernecker, A. Maier, and E. Angelopoulou, "Offline Writer Identification Using Convolutional Neural Network Activation Features," in *Proceedings of International Conference on Pattern Recognition*, J. Gall, P. Gehler, and B. Leibe, Eds., 2015, pp. 540–552.

[114] V. Christlein and A. Maier, "Encoding cnn activations for writer recognition," in *international workshop on document analysis systems (DAS)*. IEEE, 2018, pp. 169–174.

[115] S. Thadchanamoorthy, N. Kodikara, H. Premaretne, U. Pal, and F. Kimura, "Tamil handwritten city name database development and recognition for postal automation," in *International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 793–797.

[116] C. Adak, B. B. Chaudhuri, and M. Blumenstein, "An empirical study on writer identification and verification from intra-variable individual handwriting," *IEEE Access*, vol. 7, pp. 24 738–24 758, 2019.

[117] J. Miller, R. Patterson, D. Gantz, C. Saunders, M. Walch, and J. Buscaglia, "A set of handwriting features for use in automated writer identification," *Journal of Forensic Sciences*, vol. 62, 2017.

[118] M. S. Obaidullah, C. Halder, C. K. Santosh, N. Das, and K. Roy, "Phdindic_11:page-level handwritten document image dataset of 11 official indic scripts for script identification," *Multimedia Tools and Applications*, vol. 77, pp. 1643–1678, 2018.

[119] S. Biswas and A. K. Das, "Writer identification of bangla handwritings by radon transform projection profile," in *Document Analysis Systems*, 2012, pp. 215–219.

[120] S. Bilan, R. Motornyuk, S. Bilan, and O. Galan, "User identification using images of the handwritten characters based on cellular automata and radon transform," *Biometric Identification Technologies Based on Modern Data Mining Methods*, pp. 91–103, 2021.

[121] S. Karanwal, "Robust local binary pattern for face recognition in different challenges," *Multimedia Tools and Applications*, vol. 81, no. 20, pp. 29 405–29 421, 2022.

[122] D. Bertolini, L. S. Oliveira, and R. Sabourin, "Multi-script writer identification using dissimilarity," in *Pattern Recognition*, 2016, pp. 3025–3030.

[123] C.-K. Tran, P. Khamphoui *et al.*, "Face recognition technology using the fusion of local descriptors," *Annals of Computer Science and Information Systems*, vol. 34, pp. 227–231, 2022.

[124] A. Bhunia, A. Alaei, and P. Roy, "Signature verification approach using fusion of hybrid texture features," *Neural Computing and Applications*, 2019.

[125] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, 1973.

[126] K. Franke, O. Bunnemeyer, and T. Sy, "Ink texture analysis for writer identification," in *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition*, 2002, pp. 268–273.

[127] S. Arivazhagan, L. Ganesan, and S. P. Priyal, "Texture classification using gabor wavelets based rotation invariant features," *Pattern Recognition Letters*, vol. 27, no. 16, pp. 1976–1982, 2006.

[128] P. C. L. da Silva, J. P. da Silva, and A. R. G. Garcia, "Daubechies wavelets as basis functions for the vectorial beam propagation method," *Journal of Electromagnetic Waves and Applications*, vol. 33, no. 8, pp. 1027–1041, 2019.

[129] R. Hammouche, A. Attia, S. Akhrouf, and Z. Akhtar, "Gabor filter bank with deep autoencoder based face recognition system," *Expert Systems with Applications*, vol. 197, p. 116743, 2022.

[130] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection," *Pattern Recognition Letters*, vol. 10, no. 5, pp. 335 – 347, 1989.

[131] O. Babatunde, "Zernike moments and genetic algorithm: Tutorial and application," *British Journal of Mathematics and Computer Science*, vol. 4, pp. 2217–2236, 01 2014.

[132] S. Lu, Z. Lu, and Y.-D. Zhang, "Pathological brain detection based on alexnet and transfer learning," *Journal of Computational Science*, vol. 30, pp. 41–47, 2019.

[133] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[134] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[135] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[136] A. Jain, P. Flynn, and A. Ross, *Handbook of Biometrics*. Springer, 2008.

[137] H. Hassen and S. Al-Maadeed, "Arabic handwriting recognition using sequential minimal optimization," in *Arabic Script Analysis and Recognition*, 2017, pp. 79–84.

[138] A. Jammi and R. E. G., "Writer identification and recognition using radial basis function," *Computer Science and Information Technologies*, 2010.

[139] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)," *Ann. Statist.*, vol. 28, no. 2, pp. 337–407, 2000.

[140] S. Gazzah and N. Essoukri Ben Amara, "Writer identification using modular mlp classifier and genetic algorithm for optimal features selection," in *Advances in Neural Networks - ISNN 2006*, J. Wang, Z. Yi, J. M. Zurada, B.-L. Lu, and H. Yin, Eds. Springer Berlin Heidelberg, 2006, pp. 271–276.

[141] Y. Guerbai, Y. Chibani, and B. Hadjadji, "The effective use of the one-class svm classifier for handwritten signature verification based on writer-independent parameters," *Pattern Recognition*, vol. 48, no. 1, pp. 103–113, 2015.

[142] S. Dargan, M. Kumar, A. Garg, and K. Thakur, "Writer identification system for pre-segmented offline handwritten devanagari characters using k-nn and svm," *Soft Computing*, vol. 24, no. 13, pp. 10 111–10 122, Jul 2020.

[143] F. R. Dos Santos Nascimento, S. Smith, and M. Da Costa Abreu, "Exploring medieval manuscripts writer predictability: A study on scribe and letter identification," *Digital Studies/le champ numérique*, vol. 12, no. 1, 2022.

[144] N. Das, R. Sarkar, S. Basu, M. Kundu, M. Nasipuri, and D. K. Basu, "A genetic algorithm based region sampling for selection of local features in handwritten digit recognition application," *Applied Soft Computing*, vol. 12, no. 5, pp. 1592–1606, 2012.

[145] C. Halder and K. Roy, "Individuality of isolated bangla characters," in *Proc. of ICDCCom*, 2014, pp. 1–6.

[146] R. Hanusiak, L. Soares de Oliveira, E. Justino, and R. Sabourin, "Writer verification using texture-based features," *Document Analysis and Recognition*, vol. 15, pp. 1–14, 2011.

[147] C. Halder, S. M. Obaidullah, J. Paul, and K. Roy, *Writer Verification on Bangla Handwritten Characters*. Springer India, 2016, pp. 53–68. [Online]. Available: https://doi.org/10.1007/978-81-322-2653-6_4

[148] D. Chawki and S. Labiba, "A texture based approach for arabic writer identification and verification," in *Machine and Web Intelligence*, 2010, pp. 115–120.