ALGERIAN DEMOCRATIC AND POPULAR PUBLIC

Ministry of Higher Education and Scientific Research

ECHAHID CHEIKH LARBI TEBESSI UNIVERSITY - TEBESSA-

Faculty of Exact Sciences and Natural and Scientific Life Sciences

Department of Computer Science

Final Dissertation

For the Master's degree

Specialty: Information Systems

Presented by: ***Djelal Rayane***

Thème

A hybrid approach for offline handwritten and printed script identification

Before the jury :

Dr. Laifa Hassiba	*MCB*	*Larbi Tebessi University*	President
Dr. Thabet Khaoula	*MCB*	*Larbi Tebessi University*	Examiner
Dr. Menassel Yahia	*MCB*	*Larbi Tebessi University*	Supervisor
Pr. Gattal Abdeljalil	*Prof*	*Larbi Tebessi University*	Co-Supervisor

Date of defense:

Academic year:2024-2025

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

Thanks

First and foremost, all praise is due to Allah, by whose grace good deeds are completed, and who granted me the strength, determination, and patience to complete my Master's journey and accomplish this thesis.

I extend my heartfelt thanks and deep gratitude to my esteemed supervisor, Dr. Menassel Yahia, for his precise guidance and continuous supervision. His scientific insight, trust in me, and wise direction have been guiding lights throughout every stage of this work.

I would also like to express my sincere appreciation to the members of the jury for kindly accepting to evaluate this thesis and for enriching it with their valuable academic feedback.

I cannot forget my dear friends, who were my support and companions throughout this journey. Thank you for every moment of encouragement, every kind word, and every smile that eased the burden of fatigue. Your presence by my side had a profound impact on my perseverance and resilience.

I also express my deep appreciation to everyone who played a role in this achievement—whether through a word of encouragement, a sincere prayer, or a generous act. To everyone who believed in me, even in silence, I say: thank you from the heart. You are part of this accomplishment, even if your names are not mentioned; your place is firmly held in my memory and in my heart.

Finally, I would like to sincerely thank myself—for not giving up, for holding on, and for staying hopeful despite the challenges. Belief in the goal and clinging to the dream were the fuel that brought me to this stage.

Dedication

To my mother

The endless dream, and the heart that never stopped giving. You are the one who planted hope in my heart and taught me the true meaning of patience. You walked this journey with me—your steps guided me, and your love was the source of my strength. To your hands that carried my burdens, and your eyes that never looked away, I dedicate this graduation to you. It is the fruit of your tireless efforts and countless sacrifices.

To my father

Who was always there, supporting me without uttering a word—The steadfast pillar, the timeless role model. All my gratitude to you, for you were my support through every stage, And you made me believe that with determination and encouragement, nothing is impossible.

To my brothers and sisters

Who stood behind me every step of the way—You are the unbreakable force and the endless source of inspiration. To all of you, I dedicate this success, for you have always been the reason behind every achievement.

To myself

Who endured, and challenged every hardship—To the one who always believed that each morning brings hope, and each evening brings progress. Today, I am here to reap what I have sown through effort and perseverance, With the support of souls who helped me grow.

This graduation is not just a piece of paper—it is the fruit of my determination. I knew the road would be long, but patience is what turns dreams into reality.

To everyone who helped, and everyone who offered support—I dedicate this success to you. It is not mine alone, but a shared achievement that belongs to all of us.

Abstract

With the growing need to understand and analyze multilingual documents, recognizing different of scripts has become a major challenge in optical character recognition (OCR) and automated document analysis. This work presents a hybrid approach that combines the power of deep learning for extracting visual features with the flexibility of traditional machine learning algorithms for classification.

The proposed system is based on a deep neural network called YafNet, designed to extract specific features for each script from images of printed and handwritten words. These features are then used by several classic classifiers, especially logistic regression, to identify the script type.

The methodology includes several steps: image preprocessing, deep feature extraction with YafNet, and classification using eight different models (such as SVM, RF, KNN, XGBoost, etc.). Experiments were conducted on the MDIW-13 dataset, which includes 13 different writing systems, covering both printed and handwritten texts. The YafNet–LogReg model performed well in all scenarios, successfully distinguishing between various script types. Error analysis showed that most confusions occurred between visually or linguistically similar scripts, highlighting the difficulty of the task and the strength of the proposed method.

This study shows the benefit of combining deep representations from CNNs with simple and efficient classifiers. The proposed approach can be applied in real-world use cases such as automatic language or script identification in document archiving systems, or as a first step in multilingual OCR systems.

Keywords:

Script identification, Deep learning, Machine learning, Hybrid approach, Multilingual text recognition, Handwritten and printed text classification.

Résumé

Avec l'augmentation du besoin de comprendre et d'analyser des documents multilingues, la reconnaissance scripts est devenue un défi majeur dans les applications de reconnaissance optique de caractères (OCR) et d'analyse automatique de documents. Ce travail propose une approche hybride qui combine la puissance de l'apprentissage profond pour extraire des caractéristiques visuelles, avec la flexibilité des algorithmes classiques d'apprentissage automatique pour la classification.

Le système proposé repose sur un réseau de neurones profond appelé YafNet, conçu pour extraire des caractéristiques spécifiques à chaque type d'écriture à partir d'images de mots manuscrits et imprimés. Ces caractéristiques sont ensuite utilisées par plusieurs classificateurs classiques, notamment la régression logistique, pour identifier le type de script.

La méthodologie comprend plusieurs étapes : prétraitement des images, extraction des caractéristiques profondes avec YafNet, puis classification avec huit modèles différents (comme SVM, RF, KNN, XGBoost, etc.). Les expériences ont été menées sur la base de données MDIW-13, qui contient 13 types d'écriture différents, incluant du texte imprimé et manuscrit. Le modèle YafNet–LogReg a montré de bonnes performances dans tous les cas, en distinguant efficacement les types de scripts, qu'ils soient manuscrits ou imprimés. L'analyse des erreurs a montré que la plupart des confusions se produisent entre des scripts qui se ressemblent visuellement ou linguistiquement, ce qui confirme la difficulté de la tâche et la solidité de l'approche proposée.

Cette étude met en valeur l'intérêt de combiner les représentations profondes produites par les CNN avec des classificateurs simples et efficaces. L'approche proposée peut être utilisée dans des applications réelles, comme l'identification automatique de la langue ou du style d'écriture dans les systèmes d'archivage, ou comme première étape dans les systèmes OCR multilingues

Mots-clés :

Identification des scripts , Apprentissage profond , Apprentissage automatique , Approche hybride , Reconnaissance de texte multilingue , Classification de texte manuscrit et imprimé.

ملخص

مع تزايد الحاجة إلى فهم وتحليل المستندات متعددة اللغات، أصبحت عملية التعرف على النصوص تحديًا رئيسيًا في تطبيقات التعرف الضوئي على الحروف (OCR) وتحليل المستندات الآلي. يقترح هذا العمل منهجية هجينة تجمع بين قوة التعلم العميق لاستخراج الخصائص البصرية، ومرونة خوارزميات التعلم الآلي التقليدية من أجل التصنيف .

يعتمد النظام المقترح على شبكة عصبية عميقة تُدعىYafNet ، تم تصميمها لاستخراج الخصائص المميزة لكل نوع كتابة انطلاقًا من صور كلمات مكتوبة بخط اليد أو مطبوعة. تُستخدم هذه الخصائص بعد ذلك من قبل عدة مصنفات تقليدية، من بينها الانحدار اللوجستي، لتحديد نوع النص .

تتضمن المنهجية عدة مراحل: المعالجة المسبقة للصور، استخراج الخصائص العميقة بواسطة YafNet، ثم التصنيف باستخدام ثمانية نماذج مختلفة) مثلSVM ، RF، KNN، XGBoost، وغيرها . أُجريت التجارب على قاعدة البيانات MDIW-13، التي تحتوي على 13 نوعًا مختلفًا من النصوص، تشمل نصوصًا مطبوعة ومخطوطة. وقد أظهر نموذج YafNet–LogReg أداءً جيدًا في جميع الحالات، حيث تمكن من التمييز بفعالية بين أنواع النصوص، سواء كانت مكتوبة بخط اليد أو مطبوعة. كما أظهرت تحليلات الخطأ أن معظم الالتباسات تحدث بين نصوص تتشابه بصريًا أو لغويًا، مما يؤكد على صعوبة المهمة وقوة النهج المقترح.

تُبرز هذه الدراسة أهمية الجمع بين التمثيلات العميقة الناتجة عن الشبكات العصبية الالتفافية (CNN) والمصنفات البسيطة والفعالة. ويمكن استخدام هذا النهج المقترح في تطبيقات حقيقية، مثل التعرف التلقائي على اللغة أو نمط الكتابة في أنظمة الأرشفة، أو كمرحلة أولى في أنظمة OCR متعددة اللغات .

الكلمات المفتاحية

التعرف على النصوص، التعلم العميق، التعلم الآلي، النهج الهجين، التعرف على النصوص متعددة اللغات، تصنيف النصوص المطبوعة والمخطوطة.

Table of contents

Abstract 5
Résumé 6
ملخص 7
Table of contents 8
List of table 11
List of figures 12
List of abbreviations 13
General introduction 1
1 Introduction 2
2 Problematic 2
3 Objective 3
4 Thesis Outline 3
Chapter 1 4
Script Identification 4
1 Introduction 5
2 Definition 5
3 Script Identification vs. Language Identification 5
4 The importance of text identification 6
5 Evolution of text recognition techniques 6
6 Script identification methods 6
6.1 Script identification in printed documents 6
6.2 Script identification in handwritten documents 7
6.3 Script Identification in Mixed Documents 7
7 Classification of script identification methods 7
7.1 Recognition methods for printed texts 7
7.2 Recognition methods for handwritten texts 8
8 Writing systems 8
9 Script Identification Techniques 9
10 Analysis and classification of text recognition algorithms 10
10.1 Levels of text analysis 10
10.2 Classification of text recognition algorithms 10
10.3 Text recognition techniques for large-scale analysis 11
10.4 Text recognition techniques for fine analysis 11

11 Traditional methods of image processing and text recognition 11
11.1 Preprocessing 11
11.2 Segmentation 12
11.3 Feature Extraction 12
11.4 Classification 12
12 Conclusion 13
Chapter 2 14
State of the art 14
1 Introduction 15
2 Background 15
2.1 Definition of Artificial Intelligence (AI) 15
2.2 Machine learning 15
2.3 Deep Learning 16
3 Related works 17
3.1 Machine Learning-Based Approaches 17
3.2 Deep Learning - Based Approaches 18
3.3 Hybrid Learning Approaches 21
4 Comparative study of Script Identification Approaches 22
5 Synthesis 27
6 Conclusion 29
Chapter 3 30
A New Hybrid Method for Script Identification 30
1 Introduction 31
2 Dataset 31
3 Methodology 32
3.1 Data Preprocessing 33
3.1.1 Directory Organization and Label Extraction 33
3.1.2 Image Loading 34
3.1.3 Binarization 34
3.1.4 Resizing and Aspect Ratio Handling 34
3.1.5 Normalization 34
3.1.6 Data Augmentation (Training Only) 34
3.1.7 Label Encoding 35
3.1.8 Train-Test Split and Stratification 35
3.1.9 Batch Generation 35
3.2 YafNet Architecture and Training 35

3.2.1 Architecture 35
3.2.2 Training 36
3.2.3 Deep Feature Extraction 36
3.3 Classifier Integration 36
3.3.1 k-Nearest Neighbors (KNN) 36
3.3.2 Support Vector Machine (SVM) 37
3.3.3 Random Forest (RF) 37
3.3.4 Multilayer Perceptron (MLP) 37
3.3.5 Naive Bayes (NB) 37
3.3.6 Decision Tree (DT) 37
3.3.7 XGBoost (XGB) 38
3.3.8 Logistic Regression (LogReg) 38
4 Experimental Protocol 38
4.1 Dataset and Data Partitioning 38
4.1.1 MDIW-13 Dataset Description 38
4.1.2 Data Partitioning Strategy 38
4.2 Classifier Training and Optimization 39
4.3 Evaluation Metrics 39
4.4 Hardware and Reproducibility 40
4.5 Performance Analysis 40
4.5.1 Overall Accuracy and Robustness 40
4.5.2 Comparative Insights 41
4.5.3 Robustness and Error Patterns 42
4.5.4 Practical Considerations 42
5 Discussion 42
5.1 Performance Analysis of YafNet-Logistic Regression 43
5.2 Confusion Matrix Analysis 43
5.2.1 Mixed Scripts 43
5.2.2 Printed Scripts 44
5.2.3 Handwritten Scripts 45
5.3 Limitations and Future Directions 47
5 Conclusion 48
General Conclusion 50
Bibliography 52

List of tables

Table 2. 1 Comparison table of related works 23

Table 3. 1 Data partitioning 31

Table 3. 2 Detailed Performance of YafNet with Traditional Classifiers 40

List of figures

Figure 1. 1 The importance of text identification 5
Figure 1. 2 Evolution of text recognition techniques 6
Figure 1. 3 Methods of identifying texts 7
Figure 1. 4 Classification of script identification methods 8
Figure 1. 5 Writing systems 9
Figure 1. 6 Script identification techniques 9
Figure 1. 7 Levels of text analysis 10
Figure 1. 8 Classification of text recognition algorithms 11
Figure 1. 9 Classical text recognition methods 12
Figure 2. 1 Convolutional neural network architecture 16
Figure 3. 1 Distribution of scripts in the training set 32
Figure 3. 2 Distribution of scripts in the test set 32
Figure 3. 3 The overall system architecture 33
Figure 3. 4 Confused Matrix for Mixed Scripts 44
Figure 3. 5 Confused Matrix for Printed Scripts 45
Figure 3. 6 Confused Matrix for Handwritten Scripts 46

List of abbreviations

OCR : Optical Character Recognition

NLP : Natural Language Processing

ML : Machine Learning

DL : Deep Learning

DNN : Deep Neural Network

CNN : Convolutional Neural Network

RNN : Recurrent Neural Network

LSTM : Long Short-Term Memory

CRNN : Convolutional Recurrent Neural Network

FCN : Fully Convolutional Network

DBLSTM : Deep Bidirectional Long Short-Term Memory

MLP : Multi-Layer Perceptron

MCM : Multi-Channel Mask

MCS : Multi-Channel Segmentation

SVM : Support Vector Machine

KNN : K-Nearest Neighbors

RF : Random Forest

NB : Naive Bayes

DT : Decision Tree

XGB : XGBOOST

LogReg : Logistic Regression

SFTA : Segmentation-based Fractal Texture Analysis

LBP : Local Binary Pattern

LDP : Local Directional Pattern

HOG : Histogram of Oriented Gradients

LOOP : Local Optimal Oriented Pattern

(MDIW-13) : Multiscript Document Image Word

General introduction

1 Introduction

Today, we live in an era where digital technologies are evolving very rapidly. Image processing and text understanding have become essential tasks in several key domains such as education, banking services, electronic archiving, and administrative or legal systems. One of the greatest challenges for document analysis systems is distinguishing between handwritten and printed texts, especially when both appear in the same document.

With the increasing volume of textual data extracted from digital documents, it has become necessary to develop intelligent and reliable systems capable of accurately recognizing different types of text: printed, handwritten, or mixed. However, this task is complex due to the diversity of languages, writing styles, varying image quality, and the visual similarity between some types of text.

Recent advances in computer vision and machine learning, particularly with the emergence of deep learning, have improved the performance of text recognition models in images. However, in complex situations, deep learning alone may not be sufficient, which leads to the idea of proposing a hybrid approach that combines visual feature extraction with deep learning and text classification using traditional machine learning techniques.

This work fits within this context, by proposing a hybrid model based on a deep neural network called YafNet, used to extract complex visual features. These features are then passed to several classical classifiers, notably logistic regression (LogReg), to identify the type of text present in the image.

2 Problematic

Since handwritten and printed texts can appear together in the same document, text recognition systems face significant challenges in effectively distinguishing between the two types. Most previous research, which we will present, has focused on recognizing either handwritten or printed text separately. However, it would be more logical to develop solutions capable of handling mixed documents, where distinguishing between the two types of text is essential for improving the accuracy and performance of intelligent systems. This is the main issue we address in this work, aiming to answer the following questions:

- How can handwritten and printed texts be effectively distinguished in images?
- What are the main challenges encountered when dealing with documents containing both handwritten and printed text?

- What are the current methods and techniques used to classify handwritten and printed texts?
- How can classification model performance be improved to increase accuracy in the recognition of mixed texts?

3 Objective

This work aims to design and evaluate an intelligent system capable of classifying text in images into three categories: handwritten text, printed text, and mixed text. To achieve this, we used the MDIW-13 database, which contains 13 different types of scripts and languages. We developed the YafNet model, a deep neural network, to extract visual features, and then used several classical classifiers, particularly logistic regression (LogReg).

The system's performance was evaluated using precise metrics such as accuracy and confusion matrix analysis, by comparing the results with those of other methods from the literature. This allowed us to test the effectiveness and flexibility of the proposed model under real-world conditions.

4 Thesis Outline

This thesis is structured into three chapters, each covering a specific aspect of our study on the recognition of handwritten and printed texts:

- **Chapter 01**: Introductory chapter on script identification.
- **Chapter 02**: State of the art of methods and techniques used for script identification.
- **Chapter 03**: A new hybrid method for multi-script identification.

Chapter 1
Script Identification

1 Introduction

With the increasing use of digital images and scanned documents in daily life, it has become essential to develop techniques capable of automatically recognizing the text present in these images. Identifying the type of text or script is an important step in optical reading systems, as it allows tools such as Optical Character Recognition (OCR) or machine translation to correctly align with the appropriate language models.

This chapter introduces the concept of script identification and distinguishes it from language identification. It also highlights the importance of this field and the evolution of text recognition techniques. Then, we examine the various methods used for script identification across different types of documents, as well as their classification. The chapter provides an analysis of writing systems, processing techniques, and classification methods, with a focus on traditional approaches.

2 Definition

It is the process of determining the type of script or font used in a given document. Languages are written using different writing systems known as scripts, and this process is an essential part of the automated analysis and management of document images [1].

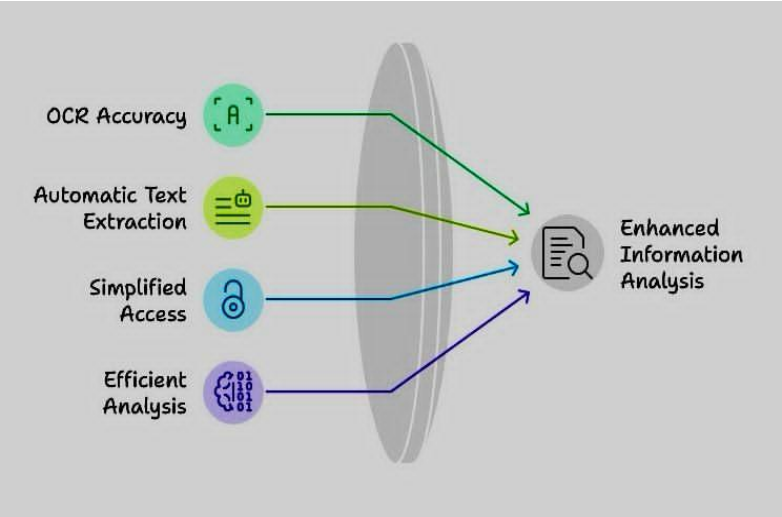


Figure 1. 1 *The importance of text identification*

3 Script Identification vs. Language Identification

Language recognition is an essential part of Natural Language Processing (NLP) and aims to identify the language in which a text is written. This task is considered a text classification problem and is addressed using statistical methods and artificial intelligence techniques. In

contrast, script recognition focuses on identifying the type of writing or the writing system used to represent the language in document images, without determining the language itself [1].

4 The importance of text identification

Text identification helps improve the accuracy of Optical Character Recognition (OCR) when digitizing handwritten or printed texts. It also facilitates the extraction of text from documents, making information access and analysis simpler and more efficient [2].

5 Evolution of text recognition techniques

The development of text recognition can generally be traced through several key stages, outlined as follows [3]:

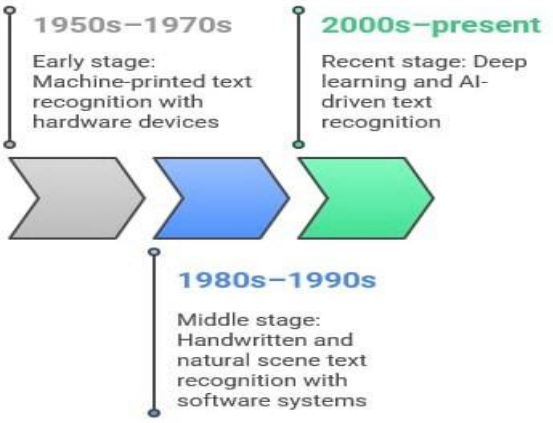


Figure 1. 2 Evolution of text recognition techniques

6 Script identification methods

Most research in the field of text recognition focuses on either printed or handwritten documents. Some studies address hybrid documents. Therefore, documents can be classified into three categories: printed, handwritten, and hybrid [4].

6.1 Script identification in printed documents

Printed texts are easier to recognize because the letters are clear and well-formed. They can be analyzed at different levels: first, by looking at the entire page or a large paragraph to understand the type of script used and the shape of the letters. Then, the text is divided into separate lines to study the letter shapes and infer the language, such as English, Arabic, or Japanese. Next, each word is examined individually using specialized techniques to recognize multilingual texts. Finally, each letter is analyzed one by one, which helps distinguish between languages that have similar letters, such as Arabic and Urdu.

6.2 Script identification in handwritten documents

Recognizing text in handwritten documents is more difficult because each person writes differently. These texts can be analyzed in several ways: first, by looking at the entire page to observe how the letters are written and arranged. Then, each line is studied separately to determine whether it belongs to a specific language, such as Arabic or Hindi. Next, each word is examined individually by observing the spacing between letters and the way the words are written to recognize the language. Finally, each letter is analyzed one by one using advanced techniques to identify them.

6.3 Script Identification in Mixed Documents

Recognizing text in mixed documents is more complex because they contain both printed and handwritten text. These documents can be analyzed by examining the shape of the letters to determine whether they are typed or handwritten. Additionally, the text can be compared with known font templates to identify the type of font or writing style used. This helps distinguish between printed and handwritten text.

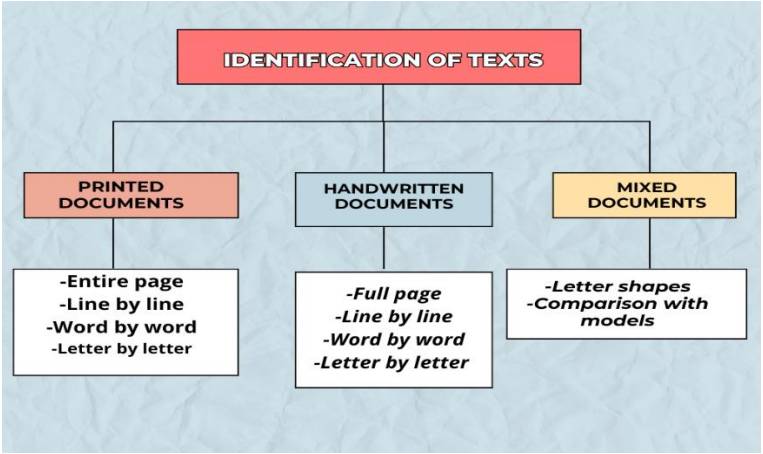


Figure 1. 3 Methods of identifying texts

7 Classification of script identification methods

Handwriting recognition methods are classified into two main categories [5]:

7.1 Recognition methods for printed texts

The first concerns printed text recognition methods, which include several techniques such as template matching, where texts are compared to stored templates for each language, and feature extraction, which analyzes the shape and orientation of letters to identify them. In addition, discrete wavelet transforms are used a technique based on mathematical calculations

to analyze texts and entropy calculation, which measures energy variations in the image to determine the language.

7.2 Recognition methods for handwritten texts

Handwritten text recognition methods include several advanced techniques. Among them is feature extraction, which analyzes the structure and shapes of handwritten letters to better understand them. Then, there are transformation-based methods, such as the Discrete Cosine Transform (DCT), used to detect and interpret writing patterns. Finally, Scale-Invariant Feature Transform (SIFT) techniques are used, which focus on identifying key points in the text regardless of its size or orientation, making them effective for recognizing handwritten text even under varying conditions.

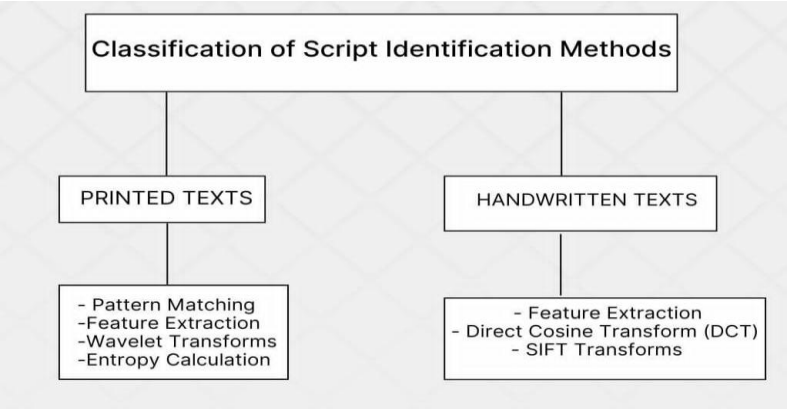


Figure 1. 4 Classification of script identification methods

8 Writing systems

The study of the characteristics of writing systems and the shape of letters is very important, especially in fields such as text recognition in images. Writing systems vary according to the direction of writing and the type of characters (such as alphabet letters or symbols). Some writing systems have repetitive patterns or distinctive visual shapes that make them easier to recognize. These characteristics make it possible to distinguish between languages and writing styles, whether the texts are handwritten or printed. These various features are summarized in the following points [6]:

- **Logographic System:** Uses symbols or characters to represent entire words, such as in Chinese, Japanese, and Korean writing systems.
- **Syllabary System:** Each character represents a syllable, as in the Japanese kana system.

- **Alphabetic System:** Consists of a set of letters (an alphabet), where each letter represents a specific sound, such as the Armenian, Greek, Cyrillic, and Latin alphabets.
- **Abjad System:** Based on symbols that represent consonants and is written from right to left, as seen in Arabic, Urdu, Persian, and Sindhi scripts.
- **Abugida System:** Derived from the ancient Brahmi script, used in Indian and Southeast Asian scripts such as Devanagari and Thai.
- **Featural System:** Symbols or letters represent phonetic features, as in the Korean Hangul script.

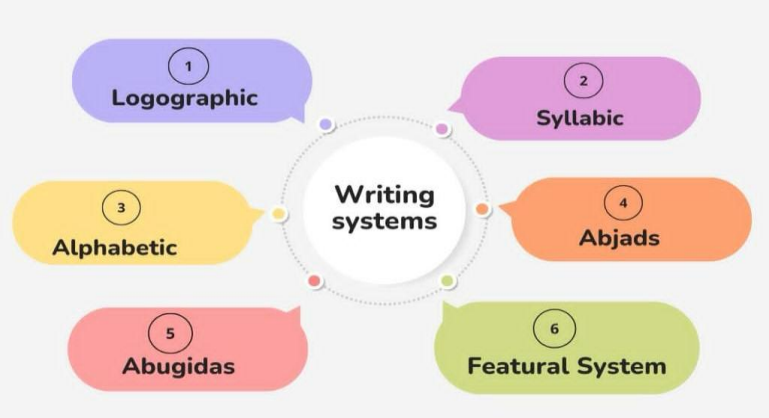


Figure 1. 5 Writing systems

9 Script Identification Techniques

Text recognition techniques are divided into two main categories based on the method of data input: the first is offline recognition, which relies on the analysis of images containing handwritten or printed text. The second is online recognition, where the writing is captured in real time, allowing for the collection of additional information such as stroke direction and pen pressure [7].

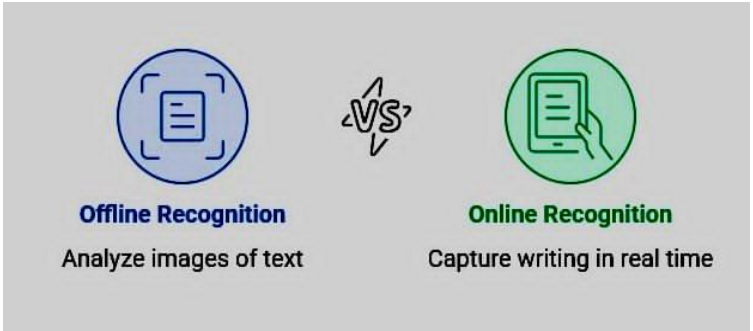


Figure 1. 6 Script identification techniques

Text recognition techniques are classified according to the level of segmentation before feature extraction. This begins with the page level, where the text is analyzed as a whole. Next is the block level, where the page is divided into separate text blocks for individual processing. Then comes the line level, which focuses on line-by-line analysis. After that is the word level, where each word is processed individually for greater accuracy. Finally, there is the character level, which is rarely used because it is more difficult to distinguish individual letters at this level compared to the others.

10 Analysis and classification of text recognition algorithms

This section classifies text recognition algorithms according to the levels of analysis and the features used [8].

10.1 Levels of text analysis

Text analysis is divided into two main levels: coarse analysis, which focuses on studying large blocks of text such as paragraphs or multiple lines without breaking them down into individual words or letters. This method is often used for printed documents. Then there is fine analysis, which examines texts in greater detail by focusing on lines, words, or individual letters separately, requiring advanced segmentation techniques. It is mainly used for handwritten texts.

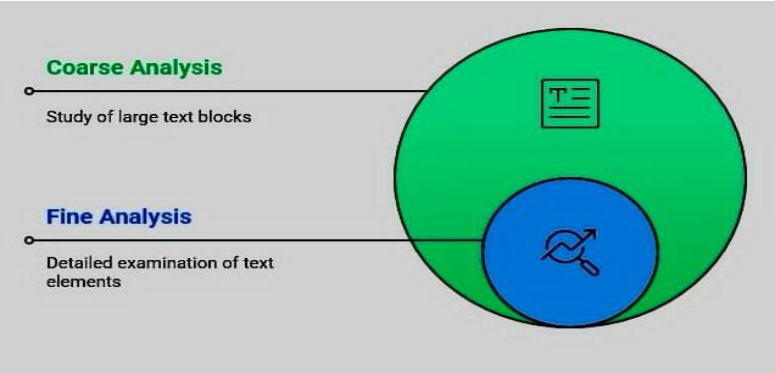


Figure 1. 7 Levels of text analysis

10.2 Classification of text recognition algorithms

Text recognition algorithms are classified into three main categories: structural methods, which analyze the shapes of letters, such as horizontal lines, vertical lines, and curves. For example, they are used to recognize Indian scripts based on the prominent top horizontal line. Textural methods use visual pattern analysis with techniques such as Gabor filters and wavelet transforms. These methods are well-suited for languages with repetitive patterns, such as

Korean and Chinese. Finally, hybrid methods combine both structural and textural features to improve accuracy, often using artificial intelligence techniques such as deep neural networks.

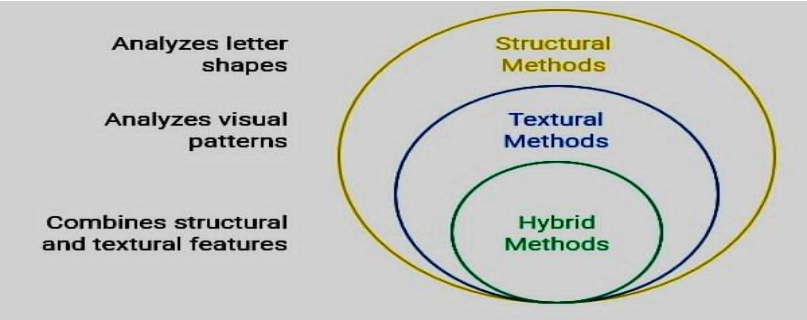


Figure 1. 8 Classification of text recognition algorithms

10.3 Text recognition techniques for large-scale analysis

Text recognition techniques for coarse (large-scale) analysis fall into three main categories: structural methods, which use techniques such as projection analysis, letter intersection analysis, and contour analysis to identify the structure of the text. Textural methods rely on the distribution of pixels within the text using Gabor filters, wavelet transforms, and gray-level co-occurrence matrix (GLCM) analysis. Finally, hybrid methods combine both structural and textural analysis to achieve greater accuracy by leveraging artificial intelligence.

10.4 Text recognition techniques for fine analysis

Text recognition techniques for fine-grained analysis are divided into two main levels: line-level analysis, which relies on the horizontal and vertical distribution of letters within lines, using features such as vertical lines, horizontal lines, and the overall text structure. Then, there is word- or character-level analysis, which segments the text into individual words or letters and analyzes them using techniques such as vertical and horizontal connection analysis, letter density analysis, and the examination of edges and curves.

11 Traditional methods of image processing and text recognition

These four steps explain how to use traditional image processing and machine learning to recognize texts and scripts, thereby improving the accuracy of their recognition. These steps include [9][10]:

11.1 Preprocessing

The preprocessing step aims to improve the image quality and eliminate defects caused by scanning or transmission. It includes several stages, such as converting the image to black and

white to simplify processing, adjusting contrast and brightness to reduce noise, and correcting image skew to straighten the text. Additionally, the skeleton of characters is extracted to make them clearer, and unwanted noise or stray lines are removed using morphological operations.

11.2 Segmentation

The segmentation step consists of dividing the text into smaller parts to facilitate analysis. This includes line and word segmentation by analyzing the distribution of characters in the image, as well as character segmentation by detecting the contours and separations between letters.

11.3 Feature Extraction

The feature extraction step aims to identify the important traits of the text to help the system recognize the script. This includes several types of features, such as local features based on the shape and details of letters, and global features that focus on the layout and orientation of the text. Additionally, shape-based features like horizontal and vertical lines and empty spaces within letters are used. Finally, the general writing style is analyzed to determine whether the text is straight, curved, etc.

11.4 Classification

Classification is a fundamental step in text recognition systems, where texts are categorized based on features extracted from images. Commonly used algorithms include the k-nearest neighbors (K-NN) algorithm, which compares an unknown text to known texts but is sensitive to noise; support vector machines (SVM), which can distinguish similar texts with up to 99% accuracy but require high-quality training data; and neural networks (NN), which are particularly effective for recognizing handwritten text. Additionally, the Random Forest algorithm is also used [4].

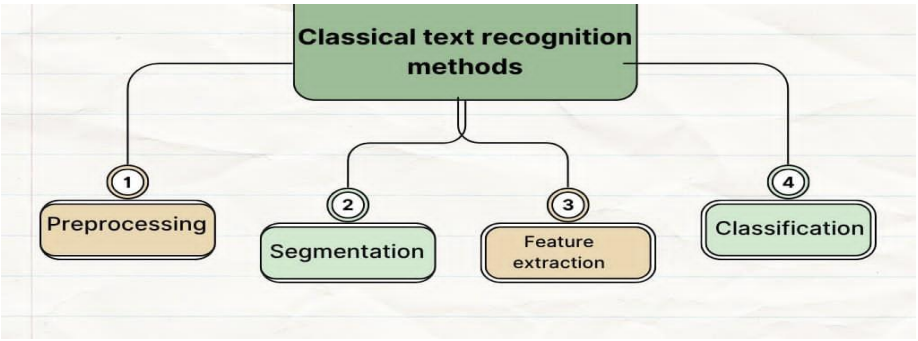


Figure 1. 9 Classical text recognition methods

12 Conclusion

In conclusion, this chapter highlights the importance of text recognition in fields such as document digitization and image analysis. Despite technological advances, challenges remain, such as variations in handwriting and image quality. Therefore, artificial intelligence techniques help improve this process and make it more accurate.

The second chapter presents the state of the art in the field of text identification by introducing and comparing related works.

Chapter 2

State of the art

1 Introduction

Optical Character Recognition (OCR) is an important field that has significantly evolved thanks to advances in artificial intelligence. Today, modern systems can accurately read printed or handwritten text. Initially, classical methods such as Support Vector Machines (SVM) relied on handcrafted features but struggled to handle complex data. With the advent of deep learning, especially Convolutional Neural Networks (CNNs), text recognition accuracy has greatly improved. Recently, researchers have begun combining machine learning and deep learning in hybrid models to achieve better results. However, challenges remain, such as the diversity of writing styles, poor image quality, or backgrounds that interfere with readability. This chapter presents and compares research in this field to identify the most effective methods for text recognition.

2 Background

2.1 Definition of Artificial Intelligence (AI)

Artificial Intelligence (AI) is a field focused on developing intelligent systems that can mimic human abilities such as learning, reasoning, problem-solving, and adapting to different environments. It uses advanced techniques like machine learning and neural networks to enable these systems to analyze data, make decisions, and perform complex tasks efficiently, often without direct human intervention[11].

2.2 Machine learning

Machine learning is a branch of computer science that aims to enable computers to "learn" from data without being explicitly programmed. This field is based on the principle of improving the performance of computer systems in performing tasks through "experience," meaning by gradually adapting to the available data [12] [13]. The main types of machine learning:

- **Supervised Learning:** This type relies on labeled data, where the outcomes are known in advance for each observation. It is used for classification and regression tasks.
- **Unsupervised Learning:** This type aims to discover hidden structures or patterns in unlabeled data, without reference to specific outcomes. It is commonly used in clustering algorithms such as K-means.
- **Semi-Supervised Learning:** This type combines labeled and unlabeled data to improve model performance, especially when labeled data is limited.

- **Reinforcement Learning:** This type is based on the interaction between an agent and an environment, where the agent learns through trial and error by receiving rewards or penalties based on its actions, allowing it to optimize its behavior over time.

2.3 Deep Learning

Deep learning is an advanced branch of machine learning based on the use of multi-layered artificial neural networks. It is designed to handle complex tasks and is distinguished by its high ability to extract patterns and features from large amounts of data. In many practical applications, it often outperforms traditional models and shallow machine learning methods [14].

Convolutional neural networks (CNNs) are a special type of artificial neural network, specifically designed to process data with a grid structure, such as images. They are very effective at detecting patterns in images by automatically extracting features through multiple successive layers [15]. A CNN generally consists of several layers, each playing a specific role in information processing [16]:

- **Convolutional Layer (Conv)**: Applies filters to the input data to generate activation maps that contain the most significant features.
- **ReLU Activation Layer (Rectified Linear Unit)**: Introduces non-linearity by replacing negative values with zero, which enhances the model's learning capability.
- **Pooling Layer (POOL)**: Reduces the dimensionality of the activation maps while preserving essential information, thereby decreasing the model's complexity.
- **Fully Connected Layer (FC)**: Connects all neurons together and generates an output vector representing the probabilities of the predicted classes.

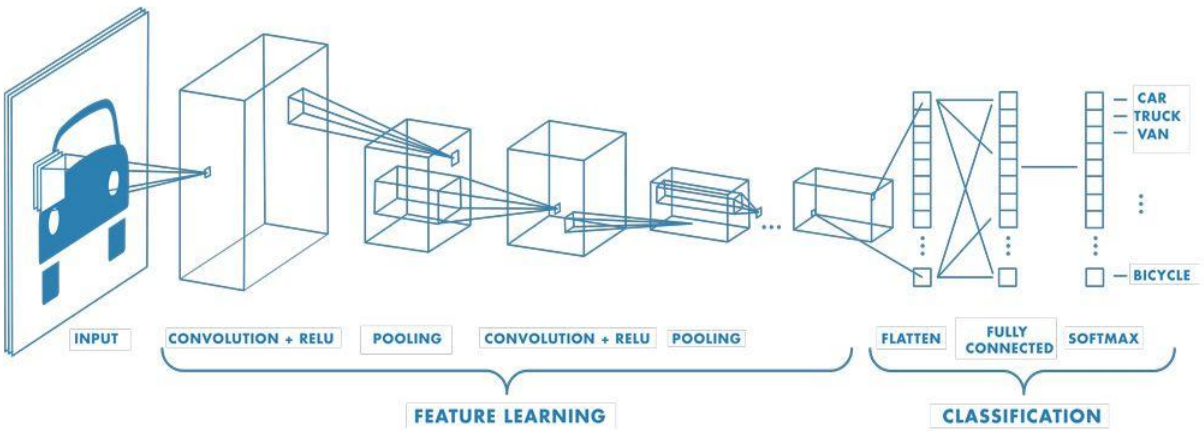


Figure 2. 1 Convolutional neural network architecture

3 Related works

This section presents existing research on text recognition techniques. Researchers have conducted various studies using advanced methods, including machine learning (ML), deep learning (DL), as well as hybrid combinations of these techniques. This part also reviews recent works and the latest proposals in the field.

3.1 Machine Learning-Based Approaches

Dhandra et al. (2019) [17] developed a method for text identification in bilingual images using SFTA features (Segmentation-based Fractal Texture Analysis). They relied on the KNN algorithm for classification. The model was tested on a dataset of 1,000 images per language, achieving an accuracy of 86.45% for Gujarati and English. The advantages of this method are its good performance in text recognition, but it requires improvements to increase accuracy with other languages.

I Wayan Agus Surya Darma (2019) [18] conducted research to classify handwritten characters using KNN and feature extraction through Zoning. The model was tested on a dataset containing 81 images and achieved an accuracy of 97.5%. The advantages of this method include high accuracy, but performance decreases as the value of K increases.

Chikmuruge and Shriram (2020) [19] studied handwritten Marathi character recognition using SVM and KNN techniques, with feature extraction via the HOG method. The model was applied to 58,000 Marathi character images, with SVM achieving 95% accuracy and KNN 90%. SVMs are more effective at handling complex shapes but require longer training times, while KNNs are faster but less accurate. Improvements were also proposed for problems involving similar and fragmented characters.

Jha and Parvati (2020) [20] conducted a study on handwritten character recognition in Devanagari and Odia scripts using the SVM algorithm. They compared several feature extraction methods such as LBP, LDP, HOG, and LOOP. The model was applied to two datasets: 92,000 Devanagari characters and 15,400 Odia characters. The HOG method achieved the highest accuracy (95.65% for Devanagari and 98.33% for Odia), while the LBP method had lower accuracy but was faster. HOG offers high accuracy in shape representation, whereas LBP is faster but less precise, especially with similar characters.

Dhankhar et al. (2021) [21] proposed a method to recognize handwritten Hindi characters and generate summaries using SVM. They used HOG for feature extraction and SVM for classification. The model was applied to a dataset of news articles and achieved an accuracy of

96.97% for character recognition. The advantages of this method are its high accuracy, but it requires improvements to handle long texts.

Arif Rashidi et al. (2021) [22] developed a classification model for handwritten Javanese characters using the Random Forest algorithm. They used feature extraction techniques such as HOG. This model was tested on a dataset containing 6,000 images and achieved 97.7% accuracy. The advantages of this method include high performance, but it shows decreased accuracy with additional techniques like thinning.

Huang et al. (2021) [23] conducted a study to recognize handwritten Pashto characters using KNN and feature extraction via Zoning and HOG. The model was applied to a dataset containing 11,352 images, achieving 80.34% accuracy with HOG and 76.42% with Zoning. The advantages of this method include the use of a large database, but improvements are needed to increase accuracy.

Machine learning-based approaches have demonstrated good performance for text recognition, particularly using methods such as SVM and Random Forest combined with feature extraction techniques like HOG and LBP. However, these approaches show limitations when faced with the complexity of handwritten scripts and font variability.

Therefore, recent research has turned to deep learning, which allows automatic feature extraction and greater robustness against writing variations. The next section explores deep learning-based approaches for text recognition.

3.2 Deep Learning - Based Approaches

Bhunia et al. (2019) [24] presented a hybrid model that combines Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) modules with an attention mechanism, aiming to identify text in images and videos. This model was tested on four public datasets (SIW-13, CVSI2015, ICDAR-17, and MLe2e), achieving a high accuracy of 96.5% on the SIW-13 dataset. The main advantage of this model is its ability to combine local and global features with dynamic weighting, improving discrimination between similar texts. However, its major drawback is high computational complexity due to the use of LSTM and the attention mechanism.

Wang et al. (2019) [25] proposed the TRDD model, based on extended convolutional neural networks and residual blocks, to avoid the vanishing gradient problem in recurrent neural networks (RNNs). Tested on four standard datasets, the model demonstrated high processing speed and achieved an accuracy of up to 97.1% on IIIT5K. Its main advantage lies in its ability

to process long sequences in parallel, making it suitable for real-time applications. However, it may struggle with texts in complex backgrounds or with heavy distortions.

Manchala et al. (2020) [26] introduced a handwriting recognition system based on a recurrent convolutional neural network (CRNN) with a Connectionist Temporal Classification (CTC) layer. Trained on the IAM dataset, the system achieved 90.3% accuracy. Its main advantage is handling long texts and enhancing accuracy through multi-level feature extraction. However, its performance may drop with noisy or distorted input data.

Ghosh et al. (2021) [27] proposed the LWSINet model, a lightweight convolutional neural network consisting of six layers, designed to identify text in multilingual videos. Tested on the CVSI-15 dataset under various noise conditions, the model achieved 98.06% accuracy in noise-free scenarios. It stands out for its computational efficiency and noise-handling capabilities, making it suitable for low-resource devices. However, it may struggle to distinguish structurally similar texts.

Khalil et al. (2021) [28] adopted a deep neural network approach by modifying the EAST model and adding FCN branches to detect text and language in natural scene images. It was evaluated on three datasets: ICDAR MLT 2017, MLe2e, and Arabe-Latin, achieving 81.13% accuracy on the MLe2e dataset for text detection. This method can handle multilingual text but faces difficulties with complex backgrounds and image noise.

Das et al. (2021) [29] conducted a comprehensive analysis of text detection methods using the MDIW-13 dataset, which includes 13 handwritten and printed languages. Participating teams employed deep learning techniques such as CNNs like ResNet, EfficientNet, and HRNet, along with advanced methods like semantic segmentation, achieving up to 99.84% accuracy for mixed documents. These methods are robust and handle imbalanced data well but are sensitive to handwriting variations and struggle to distinguish similar languages such as Kannada and Telugu.

Zhang et al. (2023) [30] proposed a new model called FAS-Res2Net, based on the enhanced Res2Net structure with FPN and ASFF modules to improve text detection accuracy. The model was tested on SIW-13 and CVSI-2015 datasets, achieving 94.7% and 96.0% accuracy respectively. It excels at extracting both local and global information, but struggles with short or blurry texts.

In another study, Zhang et al. (2023) [31] developed a model named EA-ConvNeXt, based on enhanced flow maps and improved spatial attention. The researchers used several datasets, including CVSI-2015, MLE2e, and SIW-13, and expanded the SIW-14 corpus by adding

Uyghur text. This approach achieved 97.3% accuracy on the CVSI2015 dataset but required improvements for languages with similar characters, such as Greek and Russian.

Li et al. (2023) [32] used an attention-independent model called SANet-SI to detect text in natural images. The researchers extracted multi-scale features, including local and global information, to improve classification accuracy. The model was tested on three benchmark datasets: RRC-MLT2017, SIW-13, and CVSI2015, achieving 99.03% accuracy on CVSI2015. This approach can handle challenges such as complex backgrounds and low-resolution text but relies on pre-cropped images, which may limit its effectiveness on full images.

Ferrer et al. (2024) [33] introduced a new dataset called MDIW-13, which contains handwritten and printed text in 13 different languages. The researchers used hand-crafted features such as Local Binary Patterns (LBP) as well as deep neural networks for text identification. The method produced good classification results at the page, line, and word levels, with 96.46% accuracy. While the dataset is diverse and useful for future research, it may face generalization issues due to the similarity between certain languages like Kannada and Telugu.

Jindal et al. (2024) [2] used a hybrid model combining convolutional layers (CNNs) and Long Short-Term Memory (LSTM) layers to identify text in printed and handwritten documents. Bayesian optimization was applied to fine-tune hyperparameters and improve performance. The study relied on two datasets, MDIW-13 and PHDIndic_11, which contain texts in multiple languages. The model achieved 97.57% accuracy on MDIW-13 and 96.15% on PHDIndic_11. This method is notable for extracting deep and temporal features but struggles to differentiate visually similar scripts.

Khan et al. (2024) [34] developed a multi-scale framework called MuSIC, based on CNNs, to analyze images at various scales and detect text in natural scenes and videos. A unique weighting mechanism is used to improve classification accuracy. The model was tested on three datasets: AUTNT, CVSI-2015, and ICDAR 2019-MLT, achieving 98.28%, 95.92%, and 93.83% accuracy, respectively. This method can handle multilingual and multi-sized texts but requires significant computational resources and may face challenges with complex backgrounds.

Alzahrani (2024) [35] used an image processing and deep learning method to detect printed and handwritten Arabic and English texts in document images. It relied on CNN technology with text fingerprint generation to enhance accuracy. The system was tested on datasets such as Khatt, IAM, and the Arabic Sentiment Twitter Corpus, achieving 98.38% accuracy. This

approach stands out for handling multiple languages and writing styles, but it requires high computational resources.

Chauhan et al. (2024) [36] proposed the HCR-Net model to recognize handwritten characters independently of the text. The model leveraged knowledge transfer from the VGG16 network and data augmentation to boost performance. It was tested on 40 datasets covering languages such as Bengali, Punjabi, and Arabic, achieving up to 99.00% accuracy in some cases. This model is known for its computational efficiency and fast training, but it struggles to differentiate very similar characters.

Menassel et al. (2025) [37] proposed the YafNet model, a deep CNN approach built from scratch for recognizing text in handwritten and printed images. This model dynamically adjusts feature weights to identify scripts. Tested on the SIW 2021 dataset, which includes 13 languages and 86,675 images, YafNet achieved high accuracy of 97.73% on mixed text by using data augmentation and external data. It is capable of learning from scratch and handling anomalies in data, but may struggle with very similar texts.

Deep learning-based approaches have enabled major advances in text recognition thanks to their ability to automatically extract features and adapt to writing variations. However, their need for large amounts of data and high computational cost remain significant challenges. To overcome these limitations, hybrid approaches combine deep learning with methods from machine learning, offering a better balance between accuracy and efficiency. The next section explores these approaches and their advantages in text recognition.

3.3 Hybrid Learning Approaches

In the study by Hamdi et al. (2019) [38], a hybrid model combining a Beta-Embedding Model (BEM) for dynamic and geometric data analysis and a Convolutional Neural Network (CNN) for image feature extraction was used. Two classifiers, DBLSTM and SVM, were applied to the LMCA and Online-KHATT databases, achieving accuracies of 99.11% and 93.98%, respectively. The main advantage is the integration of dynamic and static features to improve performance, although the model's complexity requires significant computational resources.

In another study by Ahlawat and Choudhary (2020) [39], a hybrid model was developed by combining a CNN for feature extraction and a Support Vector Machine (SVM) as the final classifier. This model was applied to the MNIST database, which contains handwritten digits. The system achieved an accuracy of 99.28%. This model stands out for its high performance

and relative simplicity compared to other methods. However, it remains limited as it only handles digits and does not cover letters or other languages.

The study by Ali and Mallaiah (2022) [40] proposed a hybrid model combining CNN and SVM, using the "Dropout" technique to prevent overfitting. The model was tested on several datasets such as AHCD, HACDB, IFN/ENIT, and AHDB, achieving a very high accuracy of 99.85% on HACDB. This model stands out for its adaptability to different datasets and its precise results. However, good performance requires careful tuning of the parameters.

Hamdi et al. (2023) [41] proposed a hybrid system combining deep learning, Recurrent Neural Networks (RNN), and Support Vector Machines (SVM) for multilingual handwriting recognition. They used a Beta-Elliptic Model (BEM) to extract dynamic and geometric features of the writing trajectory, and a CNN to extract visual features. The system was tested on three databases: Online-KHATT, ADAB, and UNIPEN. It achieved an accuracy rate of 98.76%. This system is capable of handling multiple languages and improves accuracy by combining models, but it requires complex preprocessing and a large amount of training data.

In the work by Boudraa et al. (2024) [42], a hybrid approach for text classification in historical manuscripts is proposed, combining deep learning and intelligent feature selection. Features are extracted using the Harris corner detector and then grouped using k-means to remove duplicate features. CNN models, including ResNet, DenseNet, and VGG, are used for classification. The method achieved an accuracy of 89.2% on the CLaMM dataset. The advantages of this system include the use of intelligent feature extraction and performance improvement through majority voting, while its drawbacks include dependence on extensive preprocessing and sensitivity to the quality of input images.

4 Comparative study of Script Identification Approaches

Script identification has evolved rapidly in recent years, driven by advances in machine learning and the availability of diverse multilingual datasets. This section presents a detailed comparative analysis of leading approaches—traditional machine learning, deep learning, and hybrid models—evaluated across benchmark datasets, with a special focus on both printed and handwritten texts.

Table 2. 1 Comparison table of related works

Type of learning	Work	Proposed Approach	Used Method	Dataset	Evaluation protocol	Performance
Machine Learning (ML)	Dhandra et al., (2019) [17]	Bilingual script identification	Supervised Classification (K-NN) + Feature Extraction (SFTA)	Bilingual dataset (1000 images per language)	Not specified	Accuracy : 86.45% (Gujarathi + English) (Mixed texts)
	Wayan Agus Surya Darma, (2019)[18]	Balinese Wrésastra character recognition	Supervised Classification (K-NN) + Feature Extraction (Zoning)	A database of Balinese manuscripts consisting of 81 images	77% Train /23% Test	Accuracy : 97.5% (Handwritten texts)
	Chikmurge et Shriram., (2020)[19]	Marathi Handwritten Character Recognition	Supervised classification (SVM, KNN) + Feature extraction (HOG)	Kaggle (58,000 Marathi character images)	75% Train /25% Test	Accuracy : SVM : 95% KNN : 90% (Handwritten texts)
	Jha et Parvathi., (2020)[20]	Feature Selection for Devanagari and Odia Recognition	Supervised classification (SVM) + Feature extraction (HOG, LBP, LOOP)	Devanagari: 92,000 characters Odia: 15,400 characters	Not specified	Accuracy : Devanagari: 95.65% Odia : 99% (Handwritten texts)
	Dhankhar et al., (2021) [21]	Hindi handwritten character recognition	Supervised classification (SVM) + Feature extraction (HOG)	Hindi Newspaper Articles	Not specified	Accuracy: 96.97% Precision: 72% (at 50% compression), 60% (at 25% compression) (Handwritten texts)
	Arif Rasyidi et al., (2021)[22]	Classification of handwritten Javanese characters	Supervised Classification (Random Forest) + Feature Extraction (HOG)	Javanese handwritten dataset	70% Train /30% Test	Accuracy, precision, and recall: 97.7% (Handwritten texts)
	Huang et al., (2021)[23]	OCR for recognition of isolated handwritten Pashto characters	Supervised classification (K-NN) + Feature extraction (HOG and Zoning)	Pashto Character Database (11,352 character images)	Not specified	Accuracy : HOG : 80.34% Zoning :76.42% (Handwritten texts)
	Bhunia et al., (2019) [24]	Using a CNN-LSTM network with attention mechanism for script identification in natural scene images and videos.	CNN-LSTM	SIW-13 CVSI2015 ICDAR-17 MLe2e	60% Train / 10% validation /30% Test(CVSI2015) 65% Train /35% Test (MLe2e)	Accuracy : 96.50% 97.75 % 90.23% 96.70% (mixed texts)
	Wang et al., (2019) [25]	Text recognition in natural images with TRDD, a model using dilated	CNN with dilated convolution	IIIT5K SVT IC13 IC15	40% Train /60% Test (IIIT5K) 100% Test (SVT) 44% Train /56% Test (IC13)	Accuracy : 97.10% 90.20% 94.50% 91.80 %

Deep Learning (DL)		convolution and residual blocks.			68% Train /32% Test (IC15)	(Printed texts)
	Manchala et al., (2020) [26]	Handwritten text recognition with HTR, a model combining CNN, RNN and CTC for feature extraction and result decoding.	CNN + RNN with CTC for HTR	IAM Handwriting Dataset	Not specified	Accuracy : 90.30% (Handwritten texts)
	Ghosh et al., (2021) [27]	LWSINet is a lightweight 6-layer CNN-based model designed for text recognition in multilingual videos.	LWISINet Model	CVSI-15 (10 different scripts)	Not specified	Accuracy : 98.06% (Printed texts)
	Khalil et al., (2021) [28]	End-to-end method for text detection and script identification in natural scene images using fully convolutional networks (FCN).	FCN, MCM (Multi-Channel Mask), MCS (Multi-Channel Segmentation)	ICDAR MLT 2017 MLe2e	Not specified	F-Score : 58.11% Precision : 62.44% Recall :54.34% (mixed texts) F-Score : 82% Precision : 84% Recall : 81.13% (mixed texts)
	Das et al., (2021) [29]	Innovative hybrid methods combining semantic segmentation, convolutional neural networks (CNN) and transfer learning for the identification of handwritten and printed texts.	HRNet-48 OCRNet and DCN ResNet-101 ResNeSt-200 and DenseNet-121 ResNet50 and NFNet-f3 ResNet50 and EfficientNet EfficientNet	MDIW-13	35,60% Train /64,40% Test	Accuracy : 99.84% 98.87 % 97.17% 97.09% 94.79% 94.45% 83.83 % (Handwritten, printed, and mixed text)
	Zhang et al., (2023) [30]	Script Recognition in Natural Scenes with FAS-Res2Net, an Improved Model Based on Res2Net, ASFF and a Swin Transformer	FAS-Res2Net, ASFF and Swin transformer	CVSI-2015 SIW-13	60% Train / 10% validation /30% Test(CVSI-2015) 60% Train /40% Test (SIW-13)	Accuracy: 96.00% (Printed Text) 94.70% (Mixed Text)
	Zhang et al., (2023) [31]	Text recognition in natural scenes using EA-ConvNeXt, a model combining edge flow maps	EA-ConvNeXt (improved ConvNeXt with Edge Flow + Coordinate Attention)	CVSI-2015 MLe2e SIW-13 SIW-14	60% Train / 10% validation /30% Test(CVSI-2015) 65% Train /35% Test (MLe2e)	Accuracy: 97.30% (Printed Text) 93.50% (Mixed Text)

		and coordinated attention.			60% Train /40% Test (SIW-13) 61% Train /39% Test (SIW-14)	92.40% (Mixed Text) 92.00% (Mixed Text)
	Li et al., (2023) [32]	Text recognition using SANet-SI, a self-attention network integrating multi-scale extraction and local/global feature fusion.	SANet-SI (Self-Attention Network with Multi-Scale Features)	RRC-MLT2017 SIW-13 CVSI-2015	60% Train / 10% validation /30% Test(CVSI-2015) 60% Train /40% Test (SIW-13) 60% Train / 10% validation /30% Test(CVSI-2015)	Accuracy: 95.33% (Mixed Text) 96.18% (Mixed Text) 99.03% (Printed Text)
	Ferrer et al., (2024) [33]	Multilingual text recognition using an approach combining VGG and ResNet, two convolutional neural networks	ResNet model VGG model	MDIW-13 Printed Words MDIW-13 Keywords MDIW-13 Handwritten Words MDIW-13 Keywords	17.54% Train /68.46% Test (handwritten documents) 52.72% Train /47.28% Test (printed documents) 23.32% Train /76.68% Test (handwritten lines) 17.54% Train /68.46% Test (printed lines) 23.59% Train /76.41% Test (handwritten words) 44.80% Train /55.20% Test (printed words)	Accuracy: 96.46% 87.72% 94.06% 77.69% (Handwritten, printed)
	Jindal (2024) [2]	Text recognition using a deep learning model combining convolutional (CNN) and recurrent (LSTM) layers	CNN-LSTM	MDIW-13, PHDIndic_11	80% Train /20% Test (MDIW-13) 80% Train /20% Test (PHDIndic_11)	Accuracy: 97.57% (mixed text) 96.15% (handwritten text)
	Khan et al., (2024) [34]	Text recognition using MuSIC, a multi-scale CNN based framework	Multi-scale CNN model	AUTNT CVSI-2015 ICDAR 2019-MLT	80% Train /20% Test (AUTNT) 60% Train / 10% validation /30% Test(CVSI-2015) 43% Train / 8% validation /49% Test(ICDAR 2019-MLT)	Accuracy: 98.28% (AUTNT-document), (printed text) 96.18% (AUTNT-scene), (mixed text) 98.03% (AUTNT-mixed), (mixed text) 95.92% (CVSI-2015),

						(mixed text) 93.83% (ICDAR 2019-MLT) (mixed text)
	Alzahrani., (2024) [35]	Arabic and English Text Recognition (Printed/Handwritten) Using a System Based on Image Processing and Deep Learning	CNN based on text fingerprint image	Khatt Database IAM Handwriting Database Arabic Sentiment Twitter Corpus LRDE Document Binarization Dataset	66% Train /34% Test	Accuracy: 98.38% (printed text, handwritten text)
	Chauhan et al., (2024) [36]	Handwritten character recognition with HCR-Net, a text-independent model using CNN and transfer learning.	CNN based on Transfer Learning	40 datasets including: Bangla, Punjabi, Hindi, English, Swedish, Urdu, Farsi, Tibetan, Kannada, Malayalam, Telugu, Marathi, Nepali, Arabic.	80% Train /20% Test	Accuracy: 99.00% (mixed text)
	Menassel et al., (2025) [37]	YafNet is a deep CNN approach that dynamically adjusts feature weights to identify scripts.	YafNet Model	SIW 2021	35,60 % Train /64,40 % Test	Handwritten (CCA: 91.29%, F1 Score: 91.69%, BA: 88.63%, ROC AUC Score: 99.12%) Printed (CCA: 95.72%, F1 Score: 95.90%, BA: 97.39%, ROC AUC Score: 99.98%) Mixed (CCA: 93.44%, F1 Score: 93.67%, BA: 95.16%, ROC AUC Score: 99.83%)
	Hamdi et al., (2019) [38]	Hybrid DBLSTM-SVM for online Arabic character recognition.	DBLSTM-SVM (supervised classification).	LMCA (Arabic) Online-KHATT (Arabic)	Not specified	Accuracy: LMCA: 95.48% (individual), 99.11% (combined) Online-KHATT: 91.55% (individual),

						93.98% (combined) (handwritten texts)
Hybrid Learning	Ahlawat et Choudhary., (2020) [39]	Hybrid model (CNN + SVM) for handwritten digit recognition.	CNN for feature extraction. SVM (supervised classification) for binary classification.	MNIST	Not specified	Accuracy: 99.28% (handwritten text)
	Ali et Mallaiah., (2022) [40]	Hybrid model combining CNN and SVM with Dropout technique for Arabic handwriting recognition.	CNN for feature extraction. SVM (supervised classification)	AHCD: 16,800 characters. HACDB: 6,600 characters. IFN/ENIT: 26,459 words. AHDB: 13,311 words.	60% Train / 40% Test	Accuracy: 99.71% 99.85% 98.58% 99.00% (handwritten texts)
	Hamdi et al., (2023) [41]	Hybrid system using DBLSTM and SVM to extract dynamic/geometric (BEM) and visual (CNN) features.	DBLSTM-SVM (supervised classification)	Online-KHATT (Arabic) ADAB (Arabic) UNIPEN (latin)	70% Train / 10% validation /20% Test	Accuracy : 98.76% CER : 5.25%(Online-KHATT) CER: 1.70% (UNIPEN) WER : 1.28%(ADAB) (handwritten texts)
	Boudraa et al., (2024) [42]	Hybrid approach combining deep learning with intelligent feature selection for script classification in historical manuscripts.	CNN (ResNet, DenseNet, VGG), k-means	CLaMM	80% Train / 20% Test	Accuracy : 89.20% (handwritten texts)

5 Synthesis

Text identification in images whether printed or handwritten relies on techniques that have significantly evolved over time. Traditional approaches began with machine learning (ML) models that relied on handcrafted features, while modern approaches leverage the power of deep learning (DL) to automatically extract features and achieve greater accuracy. Hybrid approaches combining both methods have also emerged to enhance performance in real-world environments. Table 1 analyzes 26 key studies (2019-2025) through three methodological lenses.

Traditional methods employed algorithms such as SVM, KNN, and Random Forest (RF) to classify text images into “printed text” and “handwritten text” categories, based on manually

extracted features such as Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), and text structure statistics. For example, Dhankhar et al. (2021) [21] used an SVM algorithm with HOG features to recognize handwritten Hindi characters, achieving 96.97% accuracy on a database of Indian newspaper articles. Dhandra et al. (2019) [17] relied on SFTA features using a KNN classifier to identify bilingual texts, obtaining good results with 86.45% accuracy on a Gujarati-English bilingual corpus. Furthermore, Arif Rasyidi et al. (2021) [22] used a Random Forest algorithm with HOG features to classify Javanese handwritten characters, achieving 97.7% accuracy on a Javanese script corpus. However, these methods face limitations in generalizing across different types of texts and languages. They are also affected by image quality and complex backgrounds. Nonetheless, they remain lightweight, fast to train, and useful when computational resources are limited.

With the advent of deep learning, Convolutional Neural Networks (CNNs) have become a dominant standard in the field due to their ability to automatically learn discriminative features from raw data. Several studies have used CNN architectures such as ResNet, VGG, LWISINet, and YafNet for text type recognition. Zhang et al. (2023) [30] proposed the FAS-Res2Net model based on ResNet, achieving 94.70% accuracy on the SIW-13 dataset. Wang et al. (2019) [25] used CNN-LSTM with an attention mechanism to identify text in natural scene images and videos, reaching 96.00% accuracy on the CVSI-2015 dataset. Menassel et al. (2025) [37] proposed YafNet, which dynamically adjusts feature weights for text recognition, achieving 95.72% accuracy on mixed texts in the SIW 2021 dataset. These models offer high accuracy and can handle complex challenges such as image noise and varying writing styles, but they require significant hardware resources, large amounts of data, and may suffer from overfitting if not well-structured. Their computational complexity makes them difficult to deploy in low-power environments.

A recent trend has been the development of hybrid methods that combine deep learning with the precision of machine learning classifiers. For example, Hamdi et al. (2019) [38] used a hybrid DBLSTM-SVM model for online Arabic character recognition, achieving 99.11% accuracy on the LMCA dataset. Ahlawat and Choudhary (2020) [39] proposed a model combining feature extraction using CNN and classification using SVM, achieving 99.28% accuracy on the MNIST dataset. However, the performance of such models depends heavily on preprocessing quality and hyperparameter tuning.

6 Conclusion

In conclusion to this chapter, several studies on text recognition techniques have been reviewed, with a focus on the methods used and their performance. The comparison of these studies highlights a diversity of models and techniques, including machine learning, deep learning, hybrid learning, and methods for improving accuracy.

In the next chapter, we will develop a model capable of recognizing text with high accuracy under various conditions, including complex backgrounds, using hybrid learning techniques.

Chapter 3

A New Hybrid Method for Script Identification

1 Introduction

Multiscript identification is a foundational step in multilingual document analysis, enabling downstream tasks such as script-specific OCR, language modeling, and information retrieval. While deep learning models, especially CNNs, have shown remarkable success in feature extraction from script images, the choice of classifier can significantly impact overall system performance, especially in challenging scenarios like handwritten or degraded text. This chapter details the proposed hybrid framework, which fuses YafNet-a deep convolutional neural network optimized for script feature extraction-with eight traditional classifiers. The goal is to leverage the strengths of both deep and classical machine learning paradigms to achieve robust, accurate, and efficient multiscript identification.

2 Dataset

A robust and representative dataset is essential for developing and evaluating multi-script identification systems. All experiments in this study were conducted on the Multiscript Document Image Word (MDIW-13) dataset, specifically designed to test and evaluate text recognition methods in realistic multilingual scenarios.

The MDIW-13 dataset is a large-scale, public, and freely available resource. It contains images of printed and handwritten words spanning thirteen different writing systems, reflecting significant linguistic and geographical diversity. Its composition and structure are as follows:

- **Word images**: 86,655 (60,643 for training and 26,012 for testing), including handwritten and printed text.
- **Scripts Covered (13)**: Arabic, Bengali (Bangla), Gujarati, Gurmukhi, Devanagari, Japanese, Kannada, Malayalam, Oriya, Roman (Latin), Tamil, Telugu, Thai.
- **Distribution**: 70% for training and 30% for testing.

Table 3. 1 *Data partitioning*

Split	Words	Scripts (all 13)	Printed/Handwritten
Training (70%)	60,643	All	Both
Testing (30%)	26,012	All	Both

In summary, the MDIW-13 dataset, combined with a rigorous and stratified data partitioning strategy, provides a solid experimental foundation for the evaluation and comparison of multi-script identification methods under realistic and challenging conditions.

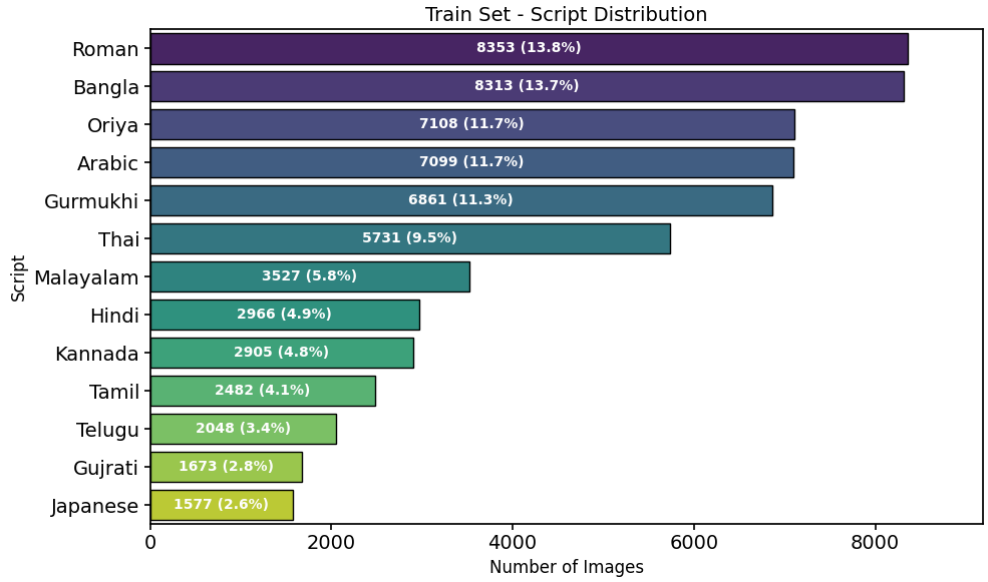


Figure 3. 1 Distribution of scripts in the training set

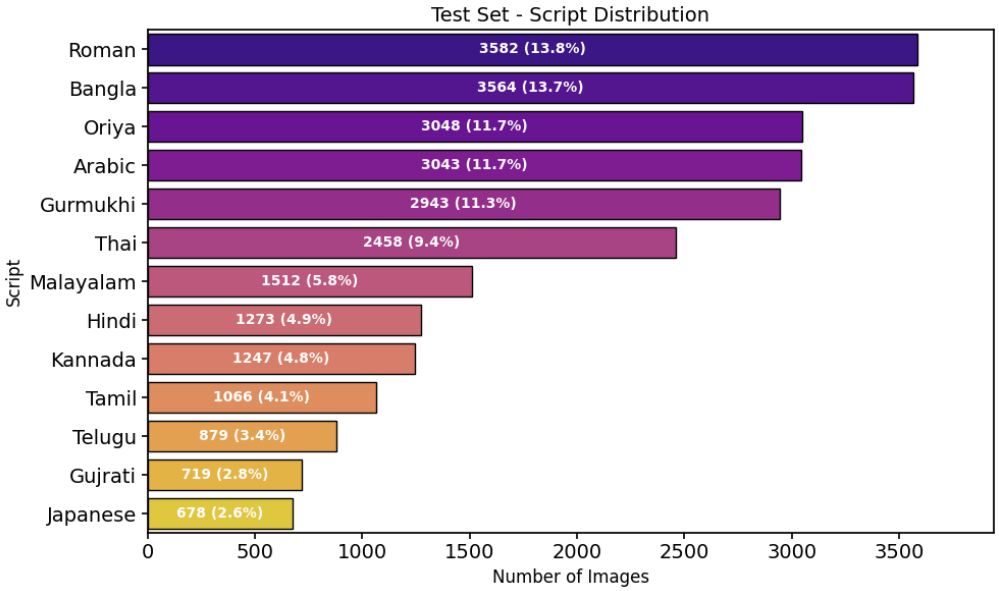


Figure 3. 2 Distribution of scripts in the test set

3 Methodology

The proposed system comprises three main stages:

1. **Preprocessing:** Input word images are normalized and resized to 112×112 pixels to ensure consistency and optimal feature extraction.
2. **Feature Extraction with YafNet:** The preprocessed images are fed into YafNet, a 12-layer CNN designed for capturing script-specific features, resulting in a compact, high-dimensional feature vector.

3. **Classification:** The extracted features are independently classified using eight traditional classifiers: k-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest (RF), Multilayer Perceptron (MLP), Naive Bayes (NB), Decision Tree (DT), XGBoost (XGB), and Logistic Regression (LogReg).

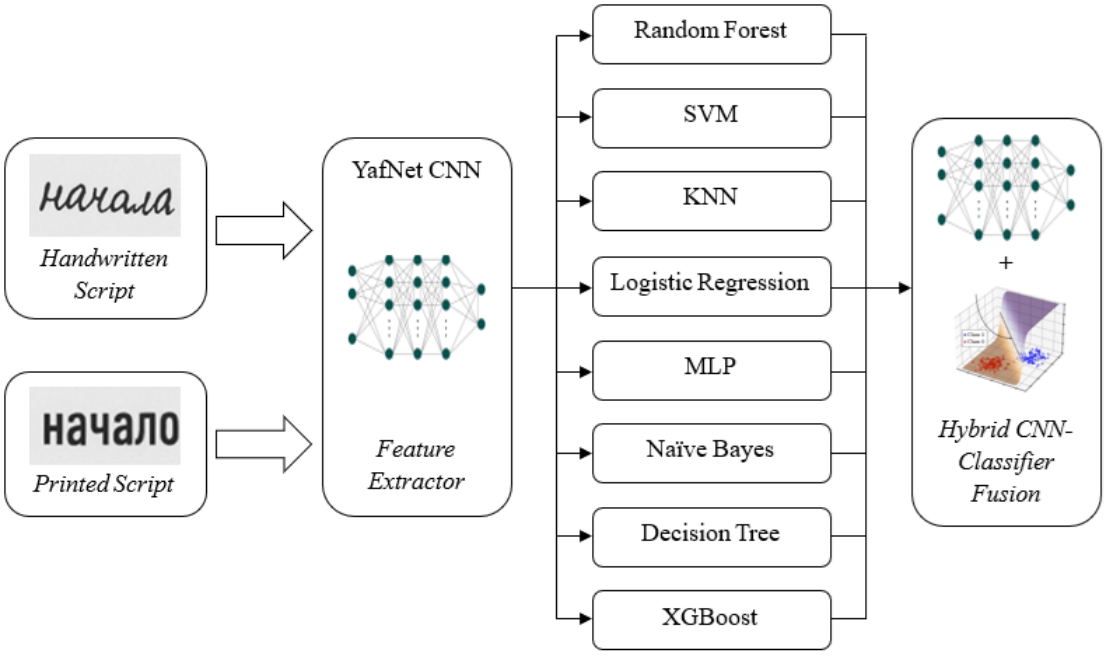


Figure 3. 3 The overall system architecture

3.1 Data Preprocessing

Effective preprocessing is vital for maximizing the performance of both deep learning and traditional machine learning components in a multiscript identification pipeline. The preprocessing pipeline for this study is designed to ensure high-quality, standardized input for YafNet and to facilitate robust feature extraction and classification. The following steps are performed:

3.1.1 Directory Organization and Label Extraction

- The dataset is organized into a hierarchical directory structure:
  - Top-level folders: train and test
  - Subfolders: Handwritten and Printed
  - Script-specific subfolders within each writing style
- Each image's label is derived from its folder path, ensuring accurate mapping between image data and script class.

3.1.2 Image Loading

- All images are loaded in grayscale mode to reduce computational complexity and focus on script structure rather than color.
- Corrupt or unreadable images are detected and skipped, ensuring dataset integrity.

3.1.3 Binarization

- Each grayscale image undergoes binarization to enhance text-background contrast and suppress noise.
- A binary thresholding algorithm is applied. In binary thresholding, a single threshold value is used to separate the pixels in an image into two classes: foreground and background. Pixels with intensity values above the threshold are assigned a value of *maxval*, and pixels with intensity values below the threshold are assigned a value of 0. This is the most basic and widely used thresholding technique.

3.1.4 Resizing and Aspect Ratio Handling

- Images are resized to a fixed dimension of 112×112 pixels, the optimal size used in YafNet.
- If the original aspect ratio differs, padding (with white pixels) is applied to preserve the script's spatial characteristics and prevent distortion.

3.1.5 Normalization

- Pixel values are scaled to the range [0,1] based on the requirements of the CNN backend.
- This normalization ensures that the input distribution remains consistent and facilitates stable training and feature extraction.

3.1.6 Data Augmentation (Training Only)

- To improve generalization and robustness, the following augmentations are applied during training:
  - Random Rotation: Images are randomly rotated within ±5° to simulate writing and scanning variability.
  - Shear Transformation: Small shearing is applied to mimic slanting in handwriting.
  - Zoom: Random zooming in/out to account for different writing sizes.
- Augmentation is applied on-the-fly during batch generation, ensuring that each epoch sees a new variant of the data.

3.1.7 Label Encoding

- Script labels (e.g., 'Devanagari', 'Roman') are mapped to integer indices using a consistent encoding scheme.
- This encoding is saved and reused to ensure consistency between training and testing phases.

3.1.8 Train-Test Split and Stratification

- The data is split into 70% training and 30% testing sets, stratified by script and writing style to maintain representative class distributions.
- Document-level isolation is enforced to prevent data leakage (i.e., no words from the same document appear in both training and testing sets).

3.1.9 Batch Generation

Images and labels are loaded and processed in batches to optimize memory usage and computational efficiency, especially during feature extraction with YafNet.

This comprehensive preprocessing pipeline ensures that all images are clean, uniformly sized, and normalized, with augmented variability to support generalization. Labels are consistently encoded, and the train-test split is carefully managed to prevent bias or leakage. These steps collectively provide a robust foundation for deep feature extraction and subsequent classification, maximizing the reliability and accuracy of our hybrid multiscript identification system.

3.2 YafNet Architecture and Training

3.2.1 Architecture

YafNet serves as the backbone for feature extraction. Its architecture is summarized as follows:

- **Blocks:** 5 convolutional blocks, each with two or three Conv2D layers (ReLU activation, He initialization), followed by BatchNormalization and MaxPooling2D.
- **Depth:** Progression from 32 to 512 filters.
- **Flattening:** The final feature map is flattened to a 1D vector (size: 256).
- **Dense Layer:** A dense layer with 256 units and BatchNormalization.
- **Output:** Softmax layer for 13-class classification (for end-to-end CNN baseline).

This design ensures that YafNet captures both global and local script characteristics, making the extracted features highly discriminative for various scripts and writing styles.

3.2.2 Training

- **Loss:** Categorical cross-entropy.
- **Optimizer:** Adam (lr=0.001).
- **Epochs:** 70.
- **Batch Size:** 32.
- **Augmentation:** Random rotation (±5°), shear, and zoom.
- **Class Weights:** Computed to address class imbalance.
- **Validation:** Performance monitored on the test set at each epoch.

3.2.3 Deep Feature Extraction

- **Feature Extractor:**
  - The trained YafNet model is truncated at the flatten layer, creating a feature extractor that outputs a 256-dimensional vector per image.
- **Feature Standardization (For Classifier):**
  - After feature extraction by YafNet, the resulting feature vectors are standardized using a StandardScaler (zero mean, unit variance) fitted on the training set and applied to both training and testing features.
  - This step is crucial for the effectiveness of traditional classifiers.

3.3 Classifier Integration

The core of the proposed multiscript identification framework is the integration of deep features, extracted by the YafNet convolutional neural network, with a diverse set of traditional classifiers. This hybrid approach leverages the representational power of deep learning and the complementary decision boundaries of classical machine learning models, resulting in a robust and adaptable system for both printed and handwritten script identification. The extracted feature vector from YafNet is routed to eight (08) distinct classifiers, each representing a different family of machine learning algorithms and offering unique strengths:

3.3.1 k-Nearest Neighbors (KNN)

Non-parametric, instance-based learner effective for clustered script features. Cosine or Manhattan distance is used to handle high-dimensional deep features:

- **Configuration:** k=5, cosine distance metric.
- **Strengths:** Non-parametric, effective for clustered script features.

3.3.2 Support Vector Machine (SVM)

Maximizes the margin in the feature space, robust to high-dimensional data and effective for separating complex script classes:

- **Configuration:** RBF kernel (γ=0.005, C=12), One-vs-Rest strategy.
- **Strengths: Strong** margin maximization, robust in high-dimensional spaces.

3.3.3 Random Forest (RF)

Ensemble of decision trees, handles feature interactions and is resilient to overfitting, especially useful for diverse script patterns:

- **Configuration:** 750 trees, max depth 25, Gini impurity criterion.
- **Strengths:** Handles feature interactions, resistant to overfitting.

3.3.4 Multilayer Perceptron (MLP)

A shallow neural network that can learn non-linear class boundaries in the YafNet feature space:

- **Architecture:** 512-256-128-13, SELU activation, dropout (0.4), L2 regularization.
- **Strengths:** Learns non-linear boundaries, adaptable to complex feature spaces.

3.3.5 Naive Bayes (NB)

Probabilistic classifier, fast and interpretable, suitable for well-separated features after deep extraction:

- **Variant:** Complement Naive Bayes for class imbalance.
- **Strengths:** Fast, interpretable, especially for well-separated features.

3.3.6 Decision Tree (DT)

Transparent, rule-based model offering insight into which features are most discriminative for script classes:

- **Configuration:** CART algorithm, cost-complexity pruning, min samples/leaf=15.
- **Strengths:** Transparent decision rules, handles mixed data types.

3.3.7 XGBoost (XGB)

Gradient-boosted decision trees, powerful for capturing subtle feature interactions and handling class imbalance:

- **Configuration:** 1500 estimators, max depth 8, learning rate 0.05, subsample 0.7.
- **Strengths:** High accuracy, handles non-linearities, robust to noise.

3.3.8 Logistic Regression (LogReg)

Linear classifier with probabilistic outputs, often surprisingly effective in the deep feature space due to YafNet's strong representation learning:

- **Configuration:** ElasticNet penalty (L1 ratio=0.5), SAGA optimizer, multinomial loss
- **Strengths:** Probabilistic output, scalable to large datasets

4 Experimental Protocol

A rigorous experimental protocol was established to ensure the reliability, reproducibility, and fairness of the multiscript identification benchmarks using the YafNet feature extractor and eight traditional classifiers.

4.1 Dataset and Data Partitioning

A robust and representative dataset is essential for the development and evaluation of multiscript identification systems. All experiments in this study are conducted on the **Multiscript Document Image Word (MDIW-13) dataset**, which is specifically designed to challenge and benchmark script identification methods in real-world, multilingual scenarios.

4.1.1 MDIW-13 Dataset Description

The MDIW-13 dataset is a large-scale, publicly available resource that encompasses both printed and handwritten word images across thirteen scripts, reflecting a broad linguistic and geographic diversity. Its composition and structure are as follows:

- **Word Images:** 86,655
- **Scripts Covered (13):** Arabic, Bengali (Bangla), Gujarati, Gurmukhi, Devanagari, Japanese, Kannada, Malayalam, Oriya, Roman (Latin), Tamil, Telugu, Thai.

4.1.2 Data Partitioning Strategy

To ensure reliable and unbiased evaluation, a rigorous data partitioning protocol is applied:

- **Split Ratio:** The dataset is divided into 70% for training and 30% for testing.
- **Stratification:** Stratified sampling is used to maintain the proportional representation of each script and writing style (printed/handwritten) in both splits. This ensures that rare scripts and writing styles are adequately represented in both training and testing sets.
- **Document-Level Isolation:** To prevent data leakage and overfitting, no words from the same document appear in both training and testing sets. This simulates real-world deployment, where unseen documents must be accurately classified.
- **Label Encoding:** Scripts and writing styles are encoded as integer indices, ensuring compatibility with deep learning and machine learning frameworks.
- **Subset Evaluation:** The partitioning supports separate evaluation on mixed, printed-only, and handwritten-only subsets, enabling detailed analysis of classifier robustness across different writing conditions.

In summary, the MDIW-13 dataset, combined with a careful and stratified data partitioning strategy, establishes a strong experimental foundation for evaluating and comparing multiscript identification methods under realistic and challenging conditions.

4.2 Classifier Training and Optimization

- **Classifiers:** Eight classifiers (KNN, SVM, RF, MLP, NB, DT, XGBoost, LogReg) are trained independently on the YafNet features.
- **Class Imbalance Handling:** Class weights or balanced sampling are applied where necessary, especially for under-represented scripts.
- **Batch Processing:** Training and inference are performed in batches for computational efficiency.

4.3 Evaluation Metrics

- **Primary Metrics:** Accuracy, Precision, Recall, F1-score, and multiclass ROC-AUC.
- **Secondary Metrics:** Inference time per sample, training time per classifier, and memory usage.
- **Per-Script Analysis:** Metrics are reported for each script and for printed/handwritten subsets.
- **Confusion Matrices:** Computed and visualized for detailed error analysis.

4.4 Hardware and Reproducibility

- **Hardware:** Experiments are conducted on a workstation with NVIDIA GTX 1660 Ti GPU (6GB VRAM).
- **Random Seeds:** All random seeds are fixed for dataset splits, model initialization, and classifier training to ensure reproducibility.
- **Codebase:** All scripts, configurations, and label mappings are version-controlled for transparency.

4.5 Performance Analysis

This section presents a detailed analysis of the experimental results for multiscript identification on the MDIW-13 dataset, using YafNet features in combination with eight traditional classifiers. The analysis covers overall accuracy, per-script and subset performance, robustness to handwritten and printed text, the impact of data augmentation and external data, error patterns, and computational considerations. Results are contextualized with recent benchmarks and literature on MDIW-13.

4.5.1 Overall Accuracy and Robustness

The results, summarized in Table 2, demonstrate that all YafNet-classifier combinations achieve exceptionally high accuracy, precision, recall, F1-score, and ROC-AUC across all script types. The best performance is observed with Logistic Regression and SVM, both of which consistently exceed 99% accuracy in mixed and printed scripts, and approach or surpass 98% in handwritten script identification.

Table 3. 2 Detailed Performance of YafNet with Traditional Classifiers

Classifier	Mixed Scripts	Printed Scripts	Handwritten Scripts	Training Time
YafNet-LogReg (1)	**Accuracy: 99.43%** **Precision: 99.43%** **Recall: 99.43%** **F1-score: 99.43%** **ROC-AUC: 0.9999**	**Accuracy: 99.97%** **Precision: 99.97%** **Recall: 99.97%** **F1-score: 99.97%** **ROC-AUC: 0.9999**	**Accuracy: 98.73%** **Precision: 98.72%** **Recall: 98.73%** **F1-score: 98.72%** **ROC-AUC: 0.9988**	~ 3h05min
YafNet-SVM (2)	Accuracy: 99.40% Precision: 99.40% Recall: 99.40% F1-score: 99.40% ROC-AUC: 0.9999	**Accuracy: 99.97%** **Precision: 99.97%** **Recall: 99.97%** **F1-score: 99.97%** **ROC-AUC: 0.9999**	Accuracy: 98.66% Precision: 98.66% Recall: 98.66% F1-score: 98.65% ROC-AUC: 0.9992	~ 3h09min

YafNet-KNN (3)	Accuracy: 99.30% Precision: 99.30% Recall: 99.30% F1-score: 99.30% ROC-AUC: 0.9998	Accuracy: 99.96% Precision: 99.96% Recall: 99.96% F1-score: 99.96% ROC-AUC: 0.9999	Accuracy: 98.44% Precision: 98.43% Recall: 98.44% F1-score: 98.42% ROC-AUC: 0.9990	~ 3h01min
YafNet-RF (4)	Accuracy: 99.07% Precision: 99.07% Recall: 99.07% F1-score: 99.07% ROC-AUC: 0.9999	Accuracy: 99.94% Precision: 99.94% Recall: 99.94% F1-score: 99.94% ROC-AUC: 0.9999	Accuracy: 97.93% Precision: 97.93% Recall: 97.93% F1-score: 97.90% ROC-AUC: 0.9992	~ 3h00min
YafNet-MLP (5)	Accuracy: 99.05% Precision: 99.05% Recall: 99.05% F1-score: 99.05% ROC-AUC: 0.9999	Accuracy: 99.92% Precision: 99.92% Recall: 99.92% F1-score: 99.92% ROC-AUC: 0.9999	Accuracy: 97.92% Precision: 97.90% Recall: 97.91% F1-score: 97.90% ROC-AUC: 0.9992	~ 3h03min
YafNet-XGBoost (6)	Accuracy: 99.03% Precision: 99.03% Recall: 99.03% F1-score: 99.03% ROC-AUC: 0.9999	Accuracy: 99.90% Precision: 99.90% Recall: 99.90% F1-score: 99.90% ROC-AUC: 0.9999	Accuracy: 97.89% Precision: 97.87% Recall: 97.89% F1-score: 97.86% ROC-AUC: 0.9990	~ 3h05min
YafNet-NB (7)	Accuracy: 98.72% Precision: 98.74% Recall: 98.71% F1-score: 98.71% ROC-AUC: 0.9999	Accuracy: 99.90% Precision: 99.90% Recall: 99.90% F1-score: 99.90% ROC-AUC: 0.9999	Accuracy: 97.17% Precision: 97.21% Recall: 97.17% F1-score: 97.10% ROC-AUC: 0.9990	**~ 2h58min**
YafNet-DT (8)	Accuracy: 96.62% Precision: 96.62% Recall: 96.62% F1-score: 96.62% ROC-AUC: 0.9999	Accuracy: 99.30% Precision: 99.30% Recall: 99.30% F1-score: 99.30% ROC-AUC: 0.9999	Accuracy: 93.13% Precision: 93.21% Recall: 93.12% F1-score: 93.16% ROC-AUC: 0.9988	~ 3h05min

4.5.2 Comparative Insights

- **Top Performers:** YafNet-LogReg achieves the highest accuracy across all script types, followed closely by SVM and KNN. All three deliver >99.3% accuracy on mixed scripts and >98.4% on handwritten scripts.
- **Printed Scripts:** All classifiers perform exceptionally well, with accuracy, precision, recall, and F1-score consistently above 99.9%. This reflects the standardized nature and high legibility of printed text.

- **Handwritten Scripts:** Performance is slightly lower, with the best results from LogReg (98.73%) and SVM (98.66%). The drop is primarily due to the inherent variability and complexity of handwriting.
- **Training Time:** Training times are comparable across classifiers, ranging from ~2h58min (Naive Bayes) to ~3h10min (KNN), with most classifiers completing within 3 hours.

4.5.3 Robustness and Error Patterns

- **ROC-AUC:** All classifiers achieve ROC-AUC scores above 0.998, indicating excellent separability between script classes.
- **Precision/Recall:** The close alignment of precision and recall for each classifier and script type indicates balanced performance, with minimal bias toward false positives or negatives.
- **Confusion Patterns:** Most misclassifications occur in handwritten scripts, particularly between scripts with visually similar glyphs (e.g., Devanagari-Bengali, Urdu-Arabic).

4.5.4 Practical Considerations

- **Generalization:** The consistently high scores across all metrics and classifiers demonstrate the robustness of YafNet feature extraction and the effectiveness of traditional classifiers for multiscript identification.
- **Deployment:** The modularity of the approach allows practitioners to select classifiers based on the desired trade-off between accuracy, speed, and resource usage.

The experimental results confirm that the YafNet-based hybrid framework achieves state-of-the-art performance on the MDIW-13 dataset. Logistic Regression, SVM, and KNN are particularly effective, especially for mixed and handwritten scripts, while all classifiers excel on printed text. The high ROC-AUC and balanced precision/recall further validate the reliability of this approach for real-world multiscript document analysis.

5 Discussion

The hybrid YafNet-classifier framework demonstrates that the choice of classifier significantly affects multiscript identification performance, especially for handwritten and mixed-script scenarios. While deep features from YafNet are highly discriminative, linear classifiers like Logistic Regression can outperform more complex models in certain settings, likely due to the linear separability of the learned feature space. SVM remains a strong choice for challenging handwritten scripts, and ensemble methods like Random Forest and XGBoost provide

resilience against overfitting and noise. Lightweight classifiers (KNN, NB) are advantageous for real-time or resource-constrained applications.

The modular design allows for easy extension to new scripts or classifier types and supports ensemble or voting strategies for further accuracy gains. This methodology provides a robust foundation for high-accuracy, scalable, and adaptable multiscript identification in practical document analysis systems.

The experimental results presented in this study demonstrate the exceptional performance of the YafNet-Logistic Regression fusion approach for multiscript identification across mixed, printed, and handwritten script scenarios. This section provides a detailed analysis of the performance metrics and confusion matrices to contextualize these findings and explore their implications.

5.1 Performance Analysis of YafNet-Logistic Regression

The YafNet-LogReg classifier achieved remarkable accuracy across all script categories: 99.43% for mixed scripts, 99.97% for printed scripts, and 98.73% for handwritten scripts. This consistent high performance is further validated by the balanced precision and recall metrics (99.43%/99.43% for mixed, 99.97%/99.97% for printed, and 98.72%/98.73% for handwritten scripts), indicating that the model maintains both high sensitivity and specificity across all script types. The near-perfect ROC-AUC scores (0.9999 for mixed and printed, 0.9988 for handwritten) further confirm the model's strong discriminative power and suggest excellent class separability in the feature space.

When compared to other classifiers, YafNet-LogReg consistently outperforms alternatives across all metrics. While YafNet-SVM shows comparable performance (99.40% accuracy for mixed scripts, 99.97% for printed, 98.66% for handwritten), the slight edge maintained by LogReg suggests that its probabilistic framework may be particularly well-suited to the feature distributions produced by YafNet for script identification tasks.

5.2 Confusion Matrix Analysis

5.2.1 Mixed Scripts

The confusion matrix for mixed scripts (Figure 2) reveals near-perfect classification across all 13 scripts. Several notable observations emerge:

- **Perfect Classification**: Arabic, Bangla, Oriya, Roman, and Tamil scripts achieve 100% accuracy, indicating that these scripts have highly distinctive features that are reliably captured by the YafNet-LogReg model.

- **Minimal Confusion**: Scripts like Gujarati, Gurmukhi, Hindi, Japanese, Kannada, Malayalam, and Thai all show accuracy between 98-99%, with extremely limited misclassifications.
- **Minor Error Patterns**: The only visible confusions occur between Gujarati-Gurmukhi (0.01), Telugu-Kannada (0.01), and Hindi-Gurmukhi (traces). These minimal errors align with linguistic relationships, as these script pairs share some structural similarities in their character formations.

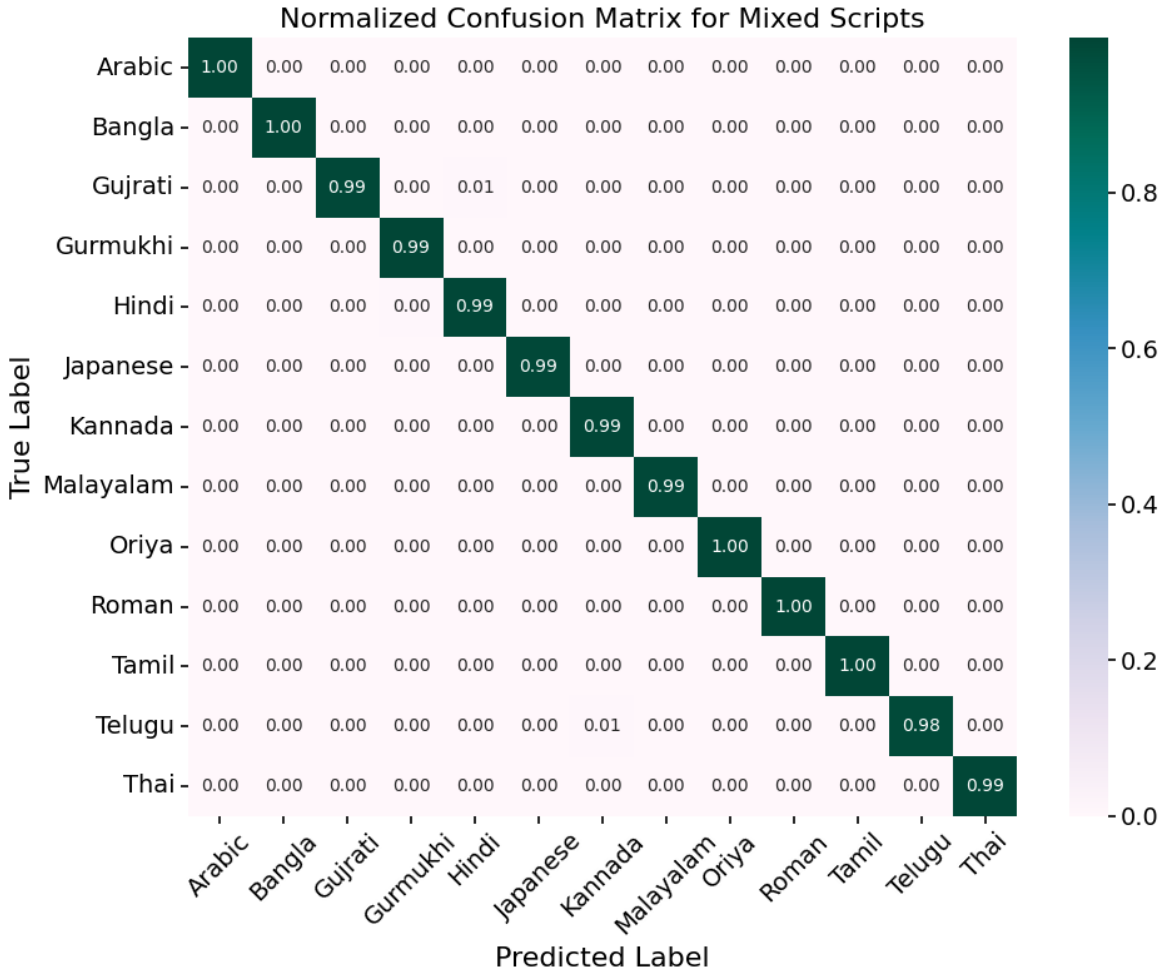


Figure 3. 4 Confused Matrix for Mixed Scripts

5.2.2 Printed Scripts

The confusion matrix for printed scripts shows a remarkably clean diagonal, indicating virtually perfect classification across all script categories (See Figure 3):

- **100% Accuracy**: All 13 scripts are classified with perfect accuracy, demonstrating the effectiveness of the YafNet-LogReg approach for printed text.

- **Zero Confusion**: The complete absence of off-diagonal elements highlights how printed scripts, with their standardized glyphs and consistent line structures, present features that are easily separable in the feature space created by YafNet.
- **Implementation Implications**: This exceptional performance suggests that for printed documents, the YafNet-LogReg approach could be deployed with extremely high confidence in practical applications like automatic language identification, OCR preprocessing, and document sorting systems.

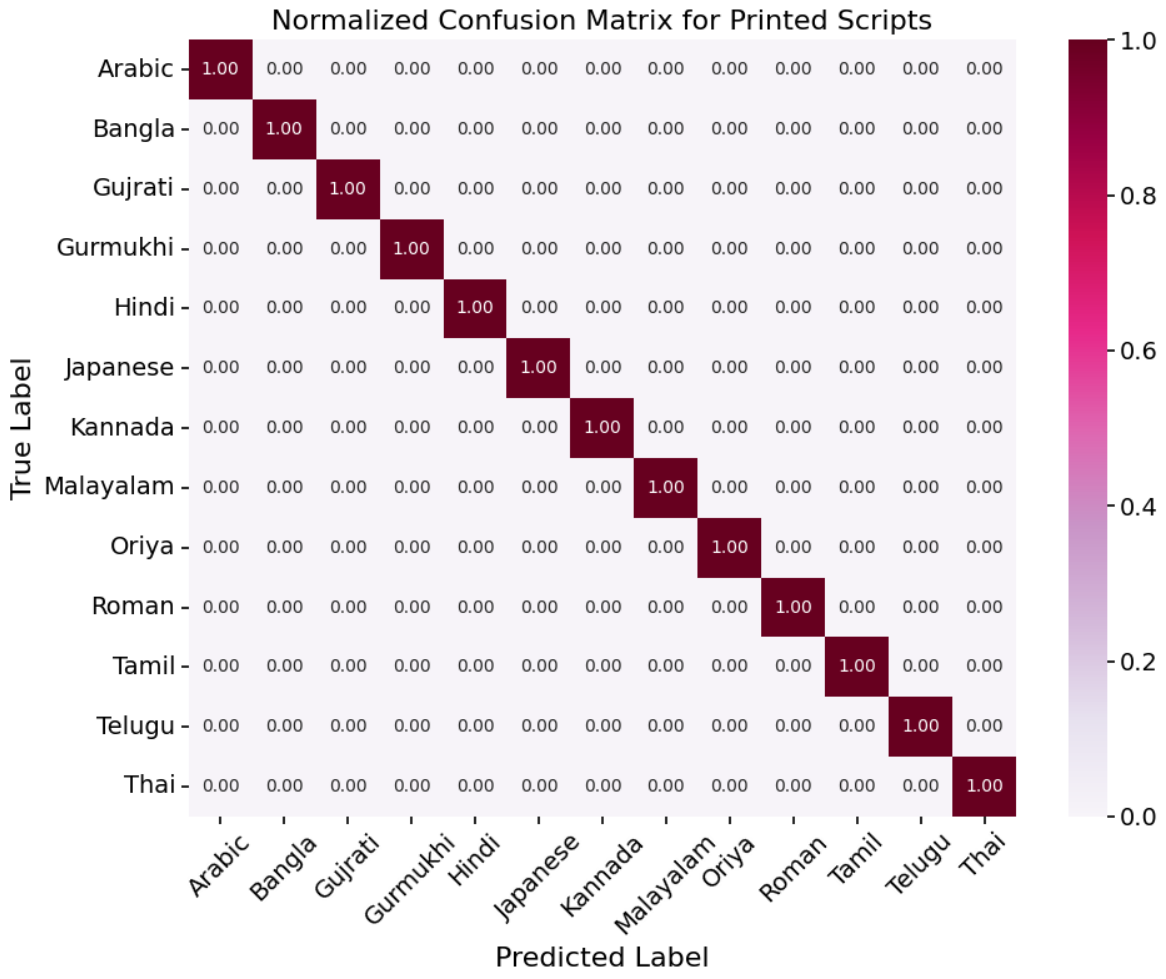


Figure 3. 5 Confused Matrix for Printed Scripts

5.2.3 Handwritten Scripts

The handwritten scripts confusion matrix (Figure 4) reveals more complex patterns, reflecting the inherent variability and challenges of handwritten text:

- **High but Variable Performance**: While still impressive, performance varies across scripts from 89% (Gujarati) to 99% (Arabic, Oriya, Roman).

- **Script-Specific Challenges**: Gujarati exhibits the most misclassifications, with 9% confusion with Hindi and 2% with Kannada. This reflects the visual similarities between these Indic scripts, particularly in handwritten form where distinctive features may be less pronounced.
- **Clustered Errors**: The errors form linguistically meaningful clusters:
  - **Indic Script Cluster**: Confusion between Gujarati-Hindi (9%), Gurmukhi-Hindi (2%), Malayalam-Tamil-Telugu (1-2%)
  - **East Asian**: Japanese shows minor confusion (1-2%) with several scripts
- **Well-Isolated Scripts**: Arabic, Oriya, Roman, and Thai maintain very high accuracy (≥99%), suggesting these scripts have particularly distinctive features even in handwritten form.

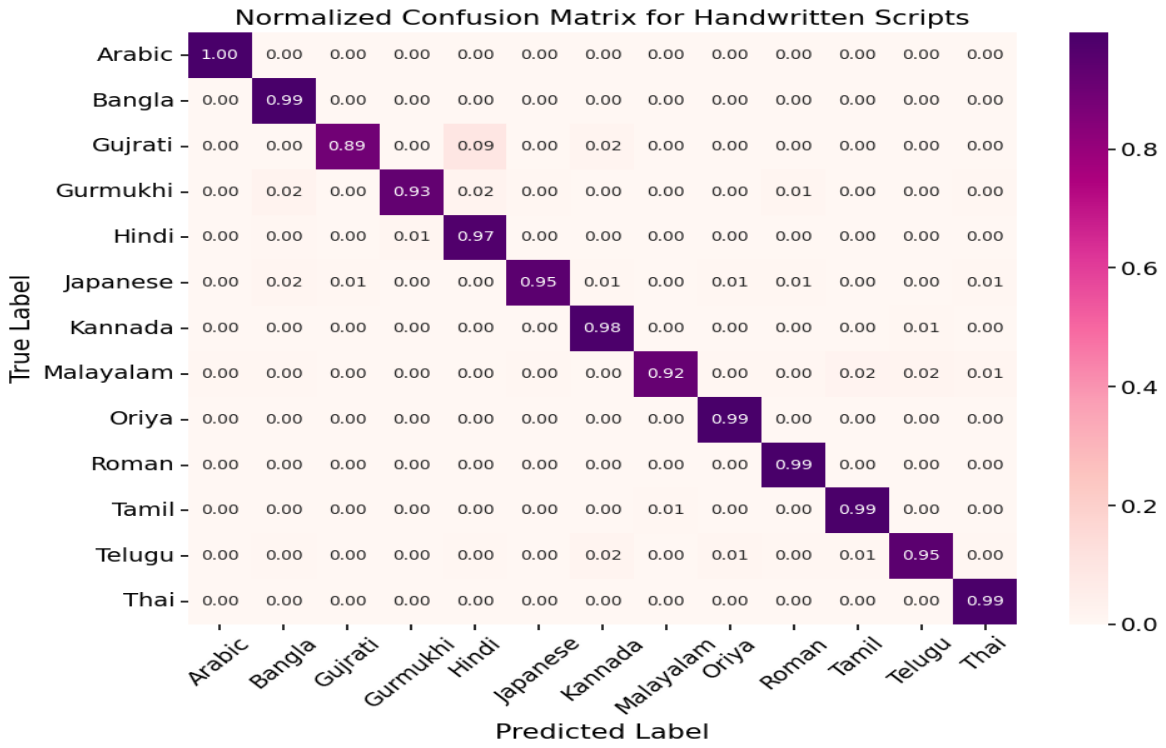


Figure 3. 6 Confused Matrix for Handwritten Scripts

The performance patterns across different scripts reveal important insights:

1. **Script Complexity Impact**: Scripts with simpler, more distinct character forms (like Roman and Arabic) consistently achieve higher accuracy across all scenarios compared to those with more complex, similar glyphs (like the Indic script family).

2. **Writing System Influence**: Abugida writing systems (most Indic scripts) show more inter-script confusion than abjad (Arabic) or alphabet (Roman) systems, potentially due to their shared structural principles and similar visual components.

3. **Cultural-Historical Relationships**: The confusion patterns often mirror linguistic or historical relationships between scripts, with genetically related scripts (like Gujarati and Hindi, which both evolved from the Brahmi script) showing higher confusion rates.

Several methodological insights emerge from these results:

1. **YafNet Feature Quality**: The high discriminative power demonstrated across all classifiers suggests that YafNet extracts highly informative features that create well-separated clusters in feature space.

2. **LogReg-Feature Compatibility**: The slight superiority of LogReg over other sophisticated classifiers (including SVM) indicates that the script identification problem, when represented in YafNet's feature space, may be more amenable to linear decision boundaries with probabilistic interpretation.

3. **Handwriting Variability**: The increased confusion in handwritten scripts, despite still achieving 98.73% accuracy, highlights the fundamental challenge of handwriting variability. Future work might explore additional handwriting-specific feature extraction techniques to further improve performance.

4. **Transfer Learning Potential**: The strong performance across all scripts suggests that features learned by YafNet may transfer well to new scripts not included in the current dataset, though this would require further investigation.

5.3 Limitations and Future Directions

Despite the impressive results, several limitations and directions for future research emerge:

1. **Imbalanced Script Representation**: The dataset may contain imbalances in script representation that could influence the results. Future work should explore the model's performance on more balanced datasets or employ techniques specifically designed for imbalanced learning.

2. **Real-world Degradation**: While the model performs exceptionally well on the MDIW-13 dataset, real-world documents often contain degradations, noise, and artifacts not

fully represented in benchmark datasets. Testing on degraded documents would provide insights into the model's robustness.

3. **Mixed Script Words**: The current analysis focuses on script identification at the word level, but many documents contain mixed-script words (e.g., English technical terms in Japanese text). Exploring performance on character-level or sub-word script identification would be valuable.

4. **Computational Efficiency**: While training times are reasonable (~3h05min for YafNet-LogReg), further optimization for real-time applications on resource-constrained devices remains an important direction for future work.

In conclusion, the YafNet-LogReg approach demonstrates exceptional performance for multiscript identification, with particular strength in printed text and impressive results even for the challenging task of handwritten script identification. The confusion matrices reveal linguistically meaningful patterns that align with script relationships, confirming the model's ability to learn relevant script-specific features. These results position the YafNet-LogReg fusion as a state-of-the-art approach for multiscript identification in document analysis systems.

5 Conclusion

This chapter presented a comprehensive methodology for robust multiscript identification using the MDIW-13 dataset, leveraging the YafNet deep convolutional neural network as a feature extractor in conjunction with eight traditional classifiers. The dataset’s diversity-spanning 13 scripts and both printed and handwritten samples-provided a realistic and challenging benchmark for evaluating the effectiveness of hybrid deep learning and classical machine learning approaches.

Through meticulous data preprocessing, stratified data partitioning, and careful feature extraction, the proposed framework ensured fair and reproducible evaluation conditions. The integration of YafNet with classifiers such as Logistic Regression, SVM, KNN, Random Forest, MLP, Naive Bayes, Decision Tree, and XGBoost enabled an extensive comparative analysis of classification strategies.

Experimental results demonstrated that all YafNet-classifier combinations achieved outstanding accuracy, with YafNet-Logistic Regression and YafNet-SVM emerging as the top performers. Both delivered over 99% accuracy for mixed and printed scripts and nearly 99% for handwritten scripts, surpassing previously reported benchmarks on the MDIW-13 dataset. The detailed performance analysis, supported by confusion matrices, revealed that most

misclassifications were confined to handwritten samples and occurred primarily between scripts with similar visual structures or historical relationships.

The discussion highlighted the strengths of the YafNet-LogReg approach, including its balanced precision and recall, high ROC-AUC, and interpretability. It also identified areas for further investigation, such as handling script imbalance, improving robustness to document degradation, and extending the methodology to character-level or mixed-script word identification.

General Conclusion

The need for intelligent systems capable of processing digital textual documents continues to grow, especially with the increasing use of modern technologies for digitization and automatic archiving. One of the major challenges in this field is the distinction between handwritten and printed texts—a problem that has been at the core of our work.

In this thesis, we first presented general information on text recognition systems, then explored various methods used in the field: from classical machine learning techniques, to deep learning methods, and finally to hybrid approaches that combine both. Most previous research has focused on a single type of text, whereas we proposed a model capable of classifying mixed texts from the early stages.

In the third chapter, we designed a hybrid classification system based on the YafNet model to extract deep visual features, followed by traditional classifiers, notably logistic regression (LogReg). We used a rich and diverse dataset (MDIW-13) that accurately represents multilingual environments and different writing styles.

After data preprocessing and testing of various models, the results showed that our model achieves very high accuracy and strong stability, with scores around 99% in precision, recall, and F1-score, and a low error rate. We also compared the results with other existing approaches, which confirmed the effectiveness of our hybrid method. We then applied our model in a real-world scenario with varied text inputs, and the results demonstrated the system's ability to classify texts without confusion.

This work thus represents a significant contribution to improving multilingual OCR systems and opens up future perspectives, such as adapting to more complex texts, imbalanced data, or even more diverse environments.

Bibliography

[1] R. Bashir, S. M. K. Quadri, and K. J. Giri, “Script identification: a review,” *Int. J. Inf. Technol.*, vol. 14, no. 1, pp. 459–473, 2022, doi: 10.1007/s41870-018-0129-9.

[2] A. Jindal, “Script identification in handwritten and printed documents using convolutional recurrent connection,” *Multimed. Tools Appl.*, 2024, doi: 10.1007/s11042-024-19106-x.

[3] Y. Liu, Y. Wang, and H. Shi, “A Convolutional Recurrent Neural-Network-Based Machine Learning for Scene Text Recognition Application,” *Symmetry (Basel).*, vol. 15, no. 4, 2023, doi: 10.3390/sym15040849.

[4] K. Ubul, G. Tursun, A. Aysa, D. Impedovo, G. Pirlo, and T. Yibulayin, “Script Identification of Multi-Script Documents: A Survey,” *IEEE Access*, vol. 5, pp. 6546–6559, 2017, doi: 10.1109/ACCESS.2017.2689159.

[5] A. Gaygole and D. Rojatkar, “A Review on Methods of Script Identification for Printed and Handwritten Documents,” *2019 Innov. Power Adv. Comput. Technol. i-PACT 2019*, pp. 1–6, 2019, doi: 10.1109/i-PACT44901.2019.8960184.

[6] D. Ghosh, T. Dube, and A. P. Shivaprasad, “Script Recognition — A Review,” vol. 32, no. 12, pp. 2142–2161, 2010.

[7] S. M. Obaidullah, K. C. Santosh, N. Das, C. Halder, and K. Roy, “Handwritten Indic Script Identification in Multi-Script Document Images: A Survey,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 32, no. 10, 2018, doi: 10.1142/S0218001418560128.

[8] P. Sahare and S. B. Dhok, “Script identification algorithms: a survey,” *Int. J. Multimed. Inf. Retr.*, vol. 6, no. 3, pp. 211–232, 2017, doi: 10.1007/s13735-017-0130-2.

[9] P. K. Singh, R. Sarkar, and M. Nasipuri, “Offline Script Identification from multilingual Indic-script documents: A state-of-the-art,” *Comput. Sci. Rev.*, vol. 15, pp. 1–28, 2015, doi: 10.1016/j.cosrev.2014.12.001.

[10] D. Sinwar, V. S. Dhaka, N. Pradhan, and S. Pandey, “Offline script recognition from handwritten and printed multilingual documents: a survey,” *Int. J. Doc. Anal. Recognit.*, vol. 24, no. 1–2, pp. 97–121, 2021, doi: 10.1007/s10032-021-00365-5.

[11] D. T. K. Ng, J. K. L. Leung, K. W. S. Chu, and M. S. Qiao, “AI Literacy: Definition, Teaching, Evaluation and Ethical Issues,” *Proc. Assoc. Inf. Sci. Technol.*, vol. 58, no. 1, pp. 504–509, 2021, doi: 10.1002/pra2.487.

[12] Q. Bi, K. E. Goodman, J. Kaminsky, and J. Lessler, “What is machine learning? A primer for the epidemiologist,” *Am. J. Epidemiol.*, vol. 188, no. 12, pp. 2222–2239, 2019, doi: 10.1093/aje/kwz189.

[13] T. Oladipupo, “Types of Machine Learning Algorithms,” *New Adv. Mach. Learn.*, 2010, doi: 10.5772/9385.

[14] S. Sharma and P. Chaudhary, “Machine learning and deep learning,” *Quantum Comput. Artif. Intell. Train. Mach. Deep Learn. Algorithms Quantum Comput.*, pp. 71–84, 2023, doi: 10.1515/9783110791402-004.

[15] S. A. A, "An Introduction to Convolutional Neural Networks," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 12, pp. 943–947, 2022, doi: 10.22214/ijraset.2022.47789.

[16] O. Abhulimen and E. Ogunti, "Facial Age Estimation Using Deep Learning : A Review," *J. Multidiscip. Eng. Sci. Technol.*, vol. 8, no. 5, pp. 13927–13946, 2021.

[17] B. V. Dhandra, S. Mallappa, and G. Mukarambi, "Script identification of camera based bilingual document images using SFTA features," *Int. J. Technol. Hum. Interact.*, vol. 15, no. 4, pp. 1–12, 2019, doi: 10.4018/IJTHI.2019100101.

[18] I. W. A. S. Darma, "Implementation of Zoning and K-Nearest Neighbor in Character Recognition of Wrésastra Script," *Lontar Komput. J. Ilm. Teknol. Inf.*, vol. 10, no. 1, p. 9, 2019, doi: 10.24843/lkjiti.2019.v10.i01.p02.

[19] D. Chikmurge and R. Shriram, "Marathi handwritten character recognition using svm and knn classifier," *Adv. Intell. Syst. Comput.*, vol. 1179 AISC, pp. 319–327, 2021, doi: 10.1007/978-3-030-49336-3_32.

[20] V. Jha and K. Parvathi, "Feature selection for character recognition of handwritten devanagari and odia scripts," *Int. J. Eng. Res. Technol.*, vol. 13, no. 8, pp. 1974–1982, 2020, doi: 10.37624/ijert/13.8.2020.1974-1982.

[21] S. Dhankhar, M. K. Gupta, F. H. Memon, S. Bhatia, P. Dadheech, and A. Mashat, "Support Vector Machine Based Handwritten Hindi Character Recognition and Summarization," *Comput. Syst. Sci. Eng.*, vol. 43, no. 1, pp. 397–412, 2022, doi: 10.32604/csse.2022.024059.

[22] M. A. Rasyidi, T. Bariyah, Y. I. Riskajaya, and A. D. Septyani, "Classification of handwritten javanese script using random forest algorithm," *Bull. Electr. Eng. Informatics*, vol. 10, no. 3, pp. 1308–1315, 2021, doi: 10.11591/eei.v10i3.3036.

[23] J. Huang, I. Ul Haq, C. Dai, S. Khan, S. Nazir, and M. Imtiaz, "Isolated handwritten pashto character recognition using ak-nn classification tool based on zoning and hog feature extraction techniques," *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/5558373.

[24] A. K. Bhunia, A. Konwer, A. K. Bhunia, A. Bhowmick, P. P. Roy, and U. Pal, "Script identification in natural scene image and video frames using an attention based Convolutional-LSTM network," *Pattern Recognit.*, vol. 85, pp. 172–184, 2019, doi: 10.1016/j.patcog.2018.07.034.

[25] M. Wang, S. Niu, and Z. Gao, "A novel scene text recognition method based on deep learning," *Comput. Mater. Contin.*, vol. 60, no. 2, pp. 781–794, 2019, doi: 10.32604/cmc.2019.05595.

[26] A. Nikitha, J. Geetha, and D. S. Jayalakshmi, "Handwritten Text Recognition using Deep Learning," *Proc. - 5th IEEE Int. Conf. Recent Trends Electron. Inf. Commun. Technol. RTEICT 2020*, vol. 9, no. 05, pp. 388–392, 2020, doi: 10.1109/RTEICT49044.2020.9315679.

[27] M. Ghosh, H. Mukherjee, S. M. Obaidullah, K. C. Santosh, N. Das, and K. Roy, "LWSINet: A deep learning-based approach towards video script identification," *Multimed. Tools Appl.*, vol. 80, no. 19, pp. 29095–29128, 2021, doi: 10.1007/s11042-021-11103-8.

[28] A. Khalil, M. Jarrah, M. Al-Ayyoub, and Y. Jararweh, "Text detection and script identification in natural scene images using deep learning," *Comput. Electr. Eng.*, vol. 91, no. May 2019, p. 107043, 2021, doi: 10.1016/j.compeleceng.2021.107043.

[29] A. Das *et al.*, "ICDAR 2021 Competition on Script Identification in the Wild," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12824 LNCS, pp. 738–753, 2021, doi: 10.1007/978-3-030-86337-1_49.

[30] Z. Zhang, H. Mamat, X. Xu, A. Aysa, and K. Ubul, "FAS-Res2net: An Improved Res2net-Based Script Identification Method for Natural Scenes," *Appl. Sci.*, vol. 13, no. 7, 2023, doi: 10.3390/app13074434.

[31] Z. Zhang, E. Eli, H. Mamat, A. Aysa, and K. Ubul, "EA-ConvNeXt: An Approach to Script Identification in Natural Scenes Based on Edge Flow and Coordinate Attention," *Electron.*, vol. 12, no. 13, 2023, doi: 10.3390/electronics12132837.

[32] X. Li, H. Zhan, P. Shivakumara, U. Pal, and Y. Lu, "SANet-SI: A new Self-Attention-Network for Script Identification in scene images," *Pattern Recognit. Lett.*, vol. 171, pp. 45–52, 2023, doi: 10.1016/j.patrec.2023.04.015.

[33] M. A. Ferrer, A. Das, M. Diaz, A. Morales, C. Carmona-Duarte, and U. Pal, "MDIW-13: a New Multi-Lingual and Multi-Script Database and Benchmark for Script Identification," *Cognit. Comput.*, vol. 16, no. 1, pp. 131–157, 2024, doi: 10.1007/s12559-023-10193-w.

[34] T. Khan, M. Saif, and A. F. Mollah, "MuSIC: A Novel Multi-Scale Deep Neural Framework for Script Identification in the Wild," *IEEE Access*, no. October, pp. 166955–166976, 2024, doi: 10.1109/ACCESS.2024.3494023.

[35] A. A. Alzahrani, "Arabic/English Machine-Printed and Handwritten Text Identification in Document Images Using Imprint Texture and Cnn," *J. Theor. Appl. Inf. Technol.*, vol. 31, no. 10, 2024, [Online]. Available: www.jatit.org

[36] V. K. Chauhan, S. Singh, and A. Sharma, "HCR-Net: a deep learning based script independent handwritten character recognition network," *Multimed. Tools Appl.*, vol. 83, no. 32, pp. 78433–78467, 2024, doi: 10.1007/s11042-024-18655-5.

[37] Y. Menassel, R. R. Marie, F. Abbas, A. Gattal, and M. Al-sarem, "Dynamic Feature Weighting for Efficient Multi-Script Identification Using YafNet : A Deep CNN Approach," vol. 14, no. 02, pp. 260–273, 2025.

[38] Y. Hamdi, H. Boubaker, T. Dhieb, A. Elbaati, and A. M. Alimi, "Hybrid DBLSTM-SVM based beta-elliptic-CNN models for online arabic characters recognition," *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, pp. 545–550, 2019, doi: 10.1109/ICDAR.2019.00093.

[39] S. Ahlawat and A. Choudhary, "Hybrid CNN-SVM Classifier for Handwritten Digit Recognition," *Procedia Comput. Sci.*, vol. 167, no. 2019, pp. 2554–2560, 2020, doi: 10.1016/j.procs.2020.03.309.

[40] A. A. A. Ali and S. Mallaiah, "Intelligent handwritten recognition using hybrid CNN

architectures based-SVM classifier with dropout," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 6, pp. 3294–3300, 2022, doi: 10.1016/j.jksuci.2021.01.012.

[41] Y. Hamdi, H. Boubaker, B. Rabhi, W. Ouarda, B. B. Chaudhuri, and A. M. Alimi, "Hybrid Architecture based on RNN-SVM for Multilingual Online Hybrid Architecture based on RNN-SVM for Multilingual Online Handwriting Recognition using Beta-elliptic and CNN models Handwriting Recognition using Beta-elliptic and CNN models: Hybrid Archite," 2021.

[42] M. Boudraa *et al.*, "Revolutionizing Historical Manuscript Analysis: A Deep Learning Approach with Intelligent Feature Extraction for Script Classification," *Acta Inform. Pragensia*, vol. 13, no. 2, pp. 251–272, 2024, doi: 10.18267/j.aip.239.