Ingénierie des Systèmes d'Information
Vol. 31, No. 3, March, 2026, pp. 965-975
Journal homepage: http://iieta.org/journals/isi

A Hybrid CNN-LSTM Model for Handwritten Text Recognition with Comparative Analysis of Similarity Algorithms for Automated Answer Scoring

Jayashree Bagade[1], Sagar Rajebhosale[2], Manisha Dhage[3], Chetas Hedaoo[4], Kavita Sultanpure[1*], Shaunak Godbole[5]

[1] Information Technology Department, Vishwakarma Institute of Technology, Pune 411037, India
[2] Computer Engineering Department, Keystone School of Engineering, Pune 412308, India
[3] AI and DS Department, Marathwada Mitra Mandal's College of Engineering, Pune 411052, India
[4] Institute of Mechanism Theory, Machine Dynamics and Robotics (IGMR), RWTH Aachen University, Aachen 52062, Germany
[5] Master of Cybersecurity, Department of Information Technology, Monash University, Melbourne 3168, Australia

Corresponding Author Email: kavita.sultanpure1@vit.edu

https://doi.org/10.18280/isi.310326

Received: 14 November 2025
Revised: 25 January 2026
Accepted: 19 March 2026
Available online: 31 March 2026

Keywords:
handwritten text recognition, CNN-LSTM, automated answer scoring, BERT embeddings, similarity algorithms, Connectionist Temporal Classification

ABSTRACT

This paper presents a hybrid deep learning model for handwritten text recognition (HTR) that combines Convolutional Neural Networks (CNN) with multi-layer Long Short-Term Memory (LSTM) networks and Connectionist Temporal Classification (CTC) for end-to-end training. The proposed architecture employs CNNs for hierarchical feature extraction from handwritten text images, followed by reshaped layers and interconnected LSTM networks to capture both spatial and temporal dependencies in handwritten sequences. The model was trained and evaluated on the IAM Handwriting Database, comprising 1,539 pages from 657 writers with 115,320 labeled words. We further investigate the application of this HTR system in automated answer scoring by comparing five similarity algorithms: BERT embeddings with cosine similarity, Jaccard similarity index, TF-IDF with cosine similarity, Universal Sentence Encoder (USE), and NLP-based approaches. The similarity algorithms were tested on 18 handwritten answer responses across three test papers, with six graded answers (Grade A to E) per paper. Experimental results demonstrate that BERT-based similarity achieved the highest accuracy of 94.44% in matching predicted scores with pre-established grades, while TF-IDF and basic cosine similarity approaches achieved only 16.67% accuracy. The Jaccard similarity index achieved moderate performance at 33.33%. Training analysis revealed characteristic learning patterns with the character error rate (CER) decreasing initially but showing signs of overfitting after approximately 40 epochs on the validation set, attributed to variations in writing styles within the IAM dataset. This research contributes a comprehensive HTR pipeline suitable for automated assessment applications and provides empirical evidence for selecting appropriate similarity metrics in educational technology contexts.

1. INTRODUCTION

In the rapidly evolving landscape of information technology, the digitization of documents has become integral to managing and analysing vast amounts of textual data. Optical Character Recognition (OCR) technology has emerged as a pivotal tool in this process, enabling the converting of printed or the handwritten data into machine-readable data. This research paper delves into the realm of Document Identification and Similarity Index, exploring the synergies between OCR technology and the quest for efficient document management.

Traditional methods of document management often faced challenges in handling large document repositories, making it cumbersome to identify, organize, and retrieve relevant information. With the advent of OCR, the digitization of documents not only facilitates accessibility but also opens avenues for advanced analysis and classification. OCR algorithms have evolved significantly, capable of recognizing diverse fonts, languages, and even handwritten text, making them versatile in addressing the needs of various domains.

English language OCR systems have found effective application across various commercial uses. The seminal study on handwritten OCR for the English language, boasts over 2900 citations. The research aimed to give an extensive overview of the latest technologies in computerized handwriting processing.

2. LITERATURE SURVEY

Arica and Yarman-Vural [1] offered an extensive examination of character recognition in their work which has

garnered over 500 citations. Their conclusion underscores the inherent nature of characters, asserting that imposing rigid mathematical rules on character patterns is nearly impossible. The complexity of patterns goes beyond the representation capabilities of structural and statistical models alone. Arica and Yarman-Vural [1] propose that various combinations of statistical and structural data information in character patterns can be effectively achieved through the application of Convolutional Neural Networks (CNNs) or harmonic Markov models (HMMs).

Connell and Jain [2] demonstrated a template-based online character recognition system that can represent different handwriting styles for a given character. Using a decision tree, they efficiently classified characters and achieved an accuracy of 86°.

Each language possesses distinct writing styles and distinctive features that set it apart from others. In the realm of efficiently recognizing handwritten and machine-printed English text, researchers have extensively utilized a wide range of feature extraction and classification techniques.

These methodologies comprise diverse strategies, encompassing HOG, bidirectional LSTM, directional features, multi-layer perceptron (MLP), hidden Markov model (HMM), Artificial Neural Network (ANN), and Support Vector Machine (SVM), among others.

In a recent study [3], researchers employed a FCNN on the datasets of RIMES and IAM, yielding promising results. The investigation demonstrated a character error rate (CER) of 4.7% and a word error rate (WER) of 8.22%, with corresponding rates of 2.46% and 5.68%. Furthermore, Jayasundara introduced an innovative approach called capsule networks (CapsNet) for recognition of handwritten characters, particularly effective in scenarios with very limited datasets. The study argues that these methods require minimal training samples per class, potentially as few as 200. The proposed technique is suggested to gain results comparable to latest advanced systems, utilizing only a tenth of the data; and after it is applied to smaller datasets, it achieved an accuracy of 0.9046.

The diversity in writing styles among individuals poses a challenge for classifiers to deliver consistent performance within the same class. In recent times, OCR research has shifted its emphasis toward deep learning methodologies with minimal attention given to manually crafted features. While the adoption of deep learning has heightened computational complexity, particularly in the training phase, it has concurrently enhanced the accuracy of classification.

Tesseract, a widely favored and commonly employed OCR engine available online [4], follows a sequential four-step process. Instead of prioritizing accuracy, its primary objective is to accommodate a broad range of languages and fonts. It has been observed that, on average, Tesseract tends to deliver higher accuracy as compared to the Transym OCR which is different openly available OCR engine. However, it is worth noting that Tesseract is not consistently faster than Transym OCR.

Programs and applications leverage OCR engines to turn the digital images into an editable format. Among the various OCR products, Adobe Acrobat Pro stands out as one of the most extensively utilized, thanks to its numerous OCR-related functionalities. Another software, Abby Fine Reader, offers similar features in this regard.

The widespread adoption of CNN in OCR is attributed in part to the accessibility of extensive datasets. Typically, researchers opt for a deep learning methodology when dealing with languages that offer a sufficiently large dataset for meaningful model learning. Despite the enhanced classification accuracy achieved by frameworks based on deep learning, there is a trade-off involving heightened computational complexity, as mentioned earlier. Recent studies have emerged wherein classical feature extraction approaches, coupled with feature selection algorithms, have been employed, resulting in state-of-the-art outcomes, for example [5-7].

A notable trend among researchers is the increasing use of CNNs to recognize the handwritten and machine text.

CNN-based architectures are preferred because they are suitable for tasks involving image input. It is widely used in visual recognition tasks in various fields [1, 2].

A novel “seven-stage approach [8]” was put out to extract text from natural photos. The filtering procedure is first applied as pre-processing to enhance the image. The separation of lateral surfaces is completed by separating the necessary content from the backdrop using the Thresholding technique. Next, the portion that is not needed is removed after the MSER has been identified. Next, the stroke width variance algorithm calculates the stroke width. Lastly, a CNN (convolutional neural network) is used to extract the features needed to identify the characters. These features are then sent to the OCR in order to extract the text.

A further in this study [9], author proposed an OCR framework based on a neural network that works on a word-level basis. This framework is based on the BLSTM. Compared to a standard OCR framework, the neural network results in an improvement of more than 20%. Furthermore, the method which doesn’t require segmentation, which is one of the most common causes of error. Furthermore, the neural network resulted in a decrease of more than 9% in CER when compared to the other available OCR framework.

Zhang et al. [10] proposed method using a two-step, iterative CRF calculation with a belief propagation obstruction to separate text containing and non-text containing parts and then using OCR on the content portion to get the best result. If there are more than one text line, we use two relational graphs to separate lines and use OCR confidence level as a benchmark to find parts containing the text.

In research, numerous additional techniques exist beyond the ones mentioned. These approaches aim to enhance the overall OCR system by either improving accuracy or addressing common sources of errors.

The OCR approach [11] is ideal for historical texts lacking font information, employing a three-step process. Initially, the text is binarized and enhanced. The second step involves segmenting the text into its individual components, utilizing the KNN to group similar symbols. Lastly, in the third step, the same classification method is applied to each image in the document, and recognition is informed by the preceding step. In summary, this method achieves a commendable throughput rate of 95.44%.

The research [12] gives a comprehensive overview of the processes and advancements in OCR. It systematically explores the various steps involved in text recognition, encompassing key stages such as preprocessing, recognition

and the feature extraction. The study delves into recent research trends in OCR, shedding light on innovative approaches, algorithms, and applications that have contributed to the evolving landscape of text recognition.

The study by Nguyen et al. [13] offers a comprehensive exploration of techniques employed after the OCR phase to enhance the accuracy and usability of digitized documents. The survey delves into various methodologies, including error correction, layout analysis, and document structure improvement, providing a critical overview of advancements in post-OCR processing. By synthesizing insights from a range of studies, the survey contributes valuable perspectives on the challenges and innovations within this domain, thereby enriching the literature on OCR technology and its applications in document processing.

Biró et al. [14] introduces an innovative approach to OCR by combining CRNN and Scene Text Visual Recognition (SVTR) models. The focus is on creating a synthesized OCR system capable of recognizing text in multiple languages. The research emphasizes real-time collaborative tools, suggesting the potential application of the proposed OCR approach in dynamic and collaborative digital environments. The synthesis of CRNN and SVTR models aims to increase not only the accuracy but also the efficiency of text recognition, showcasing a promising direction in the evolution of OCR technology.

Souibgui et al. [15] introduces an innovative approach to recognition of text and document enhancement. The proposed Text-DIAE is a self-supervised autoencoder specifically designed to address degraded text within documents. Unlike conventional OCR systems, Text-DIAE excels in recognizing and enhancing text subjected to various degradation conditions, including noise, blurriness, and other forms of visual interference. Through self-supervised learning, the model effectively learns to encode and decode degraded text, imparting robustness to variations in document quality. The study focuses on enhancing the The study focuses on enhancing the text reconition performance and enhancement of document tasks, presenting promising outcomes in effectively addressing real-world challenges associated with document degradation.

The research paper [16] investigates the utilization of a Convolutional Recurrent Neural Network (CRNN) in the context of scene text recognition. The study emphasizes harnessing the combined capabilities of convolutional and recurrent layers within a neural network architecture to tackle the challenges associated with recognizing text in natural scenes. Through the application of machine learning techniques, the authors seek to improve the accuracy and resilience of scene text recognition, providing valuable insights for the broader fields of OCR and document analysis. The research highlights the CRNN model's efficacy in handling intricate visual contexts and diverse text formats, presenting potential advancements in the creation of efficient and adaptable text recognition systems for real-world applications.

The research [17] introduces a comprehensive approach to extracting information from unstructured data. This hybrid solution synergizes the capabilities of OCR and NLP to improve the accuracy and efficiency of extracting meaningful information from a variety of textual sources. By incorporating OCR, the system converts printed or handwritten text into machine-readable data, and NLP techniques are subsequently applied to further process and interpret the content, enabling a more nuanced understanding of information within unstructured data. This innovative amalgamation addresses the limitations associated with individual technologies, providing a holistic solution for extracting valuable insights from unstructured textual information.

The research paper [18] introduces OCR-D, an open-source OCR framework designed specifically for historical printed documents. This framework provides an end-to-end solution, encompassing the entire OCR workflow, from preprocessing to post-processing. By focusing on historical documents, OCR-D addresses the unique challenges posed by diverse fonts, layouts, and degradation in aged prints. The paper likely discusses the architecture, features, and performance of OCR-D, making it a valuable addition to the literature survey for its contribution to advancing OCR technologies in the context of historical document digitization.

The paper [19] presents an innovative method for improving OCR accuracy in the context of medical reports. The research focuses on leveraging deep learning techniques as a post-correction mechanism to rectify errors introduced during the OCR process. By addressing the unique challenges posed by medical documents, such as complex terminology and varied formatting, the proposed approach aims to enhance the overall reliability of extracting text from medical reports. This contribution to OCR post-correction methods holds particular significance in the healthcare domain, where accuracy and precision in document processing are crucial for effective clinical decision-making and information retrieval.

The paper [20] introduces an adaptive framework for the recognition of handwritten numerical digits using OCR methods. The research emphasizes the use of artificial intelligence to enhance the accuracy of digit recognition in handwritten documents. The framework is designed to be adaptive, allowing for continuous learning and improvement in recognition capabilities over time. This study contributes to the broader field of OCR by specifically addressing challenges associated with handwritten numerical digits, offering insights into the development of intelligent systems for enhanced document analysis and comprehension.

The research paper [21], explores the consequences of OCR errors on in performance of information retrieval systems. The study investigates how inaccuracies introduced during the OCR process affect the effectiveness of retrieving relevant information from digitized documents. By evaluating the impact of OCR errors on search precision, recall, and overall retrieval performance, the research provides valuable insights into the challenges associated with OCR technology and its implications for information retrieval systems. Understanding the extent of these errors is crucial for researchers and practitioners seeking to optimize OCR applications in the context of document management and analysis.

The study [22], focuses on the development of an OCR system tailored for historical documents, addressing the challenge of limited training data. The research explores innovative techniques to enhance OCR performance in scenarios where historical documents pose unique

challenges, such as diverse fonts, degraded text, and limited available training samples. By proposing efficient strategies to overcome data scarcity issues, the study contributes valuable insights to the broader field of OCR, particularly in the field of historical documents analysis.

The paper [23] presents a study that explores the application of CNNs in the field of handwritten document recognition. The research focuses on leveraging CNNs, a type of deep learning architecture well-suited for image processing tasks, to automatically recognize and interpret handwritten documents. The objective is to increase not only the accuracy but also efficiency of handwritten document recognition systems, offering a promising approach to address the challenges associated with diverse handwriting styles and variations in document layouts. The study likely investigates the effectiveness of CNNs in capturing intricate patterns and features within handwritten documents, ultimately contributing valuable insights to the broader literature on document recognition and characterizing the role of deep learning in intelligent document processing.

The paper [24] investigates the errors generated by OCR systems and proposes a deep statistical analysis approach to enhance post-OCR processing. The study employs advanced statistical techniques to analyze and categorize OCR errors, aiming to understand their patterns and characteristics. By leveraging deep learning methods, the research seeks to develop effective strategies for mitigating OCR errors during post-processing stages. The findings contribute valuable insights into the improvement of OCR accuracy and highlight the importance of addressing errors in the post-OCR phase for enhanced document processing and information retrieval.

The research [25] examines the incorporation of partial least squares (PLS) as a technique for reducing features in OCR applied to biometric identification. The study explores the impact of PLS on enhancing the efficiency and accuracy of OCR systems, specifically within the realms of artificial intelligence and biometric applications. This inventive approach seeks to optimize the extraction of features, with the overarching goal of enhancing the overall performance of OCR technology in scenarios related to biometric identification.

Marti and Bunke [26] provided an in-depth examination of OCR techniques with a specific focus on CNNs. The paper systematically reviews and analyzes recent advancements in OCR, emphasizing the application of CNNs in this context. The literature survey explores various CNN architectures employed for OCR tasks, assessing their strengths, limitations, and comparative performance. Additionally, the review highlights key challenges in OCR using CNNs, such as handling diverse fonts, languages, and document layouts. The synthesis of this comprehensive survey contributes valuable insights to the evolving landscape of OCR technologies, informing researchers and practitioners about the latest technological and potential directions for future advancements in the field.

A seminal contribution in this domain is the CRNN proposed by Shi et al. [27], which integrates CNN for feature extraction with recurrent layers for sequence modeling and Connectionist Temporal Classification (CTC) for transcription without explicit segmentation. This architecture has become a foundational model for many modern HTR systems. Similarly, Graves [28] introduced the use of CTC, enabling sequence-to-sequence learning without the need for aligned labels. This approach significantly improved the efficiency of training end-to-end recognition systems and is widely adopted in HTR pipelines.

Recent studies have also explored attention-based encoder-decoder architectures, which overcome limitations of CTC by dynamically focusing on relevant parts of the input sequence. However, CNN-LSTM-CTC architectures remain computationally efficient and robust for large-scale handwritten datasets such as IAM.

In addition to recognition, the integration of OCR/HTR systems into automated assessment frameworks has gained attention. Research in educational technology highlights the potential of combining text recognition with Natural Language Processing (NLP) for evaluating subjective answers. For instance, semantic similarity models based on transformer architectures such as Jacob Devlin's BERT have shown remarkable performance in capturing contextual meaning, outperforming traditional approaches like TF-IDF and Jaccard similarity.

Studies on automated short-answer grading demonstrate that context-aware embeddings significantly improve scoring accuracy compared to lexical matching techniques. However, most existing systems assume digitally available text, limiting their applicability in handwritten examination settings.

This gap motivates the need for integrated frameworks that combine HTR with semantic evaluation. The present work contributes to this area by coupling a CNN-LSTM-CTC-based HTR model with multiple similarity algorithms, enabling automated scoring directly from handwritten responses. Unlike prior studies that treat recognition and evaluation as separate tasks, this approach provides a unified pipeline for real-world educational applications.

3. METHODOLOGY

The flow of the suggested system is shown in detail in Figure 1. A camera or scanner is used to capture an image of the response sheet. Using OCR technology, scanned photos can be transformed into text that can be read by machines. Preprocess the recovered image using a variety of methods, such as slant and skew correction, noise reduction, and enhancement. Line, word, and character segmentation are applied to the pre-processed image. Character classes are identified by the classifier using extracted features, and a similarity score is computed.

Traditional Handwritten Text Recognition (HTR) systems face limitations in handling variations in handwriting styles, noise, and complex spatial dependencies. This research addresses these challenges through the integration of a CNN-based feature extraction model followed by a novel multi-LSTM network architecture, providing a comprehensive solution to HTR. Figure 2 depicts the architecture of the system.

For sequence recognition tasks where input and output alignment is unknown, such as handwriting recognition, speech recognition, or OCR, CNN with multi-layer Long Short-Term Memory (LSTM) networks and CTC pipeline are frequently utilised. CNN extracts features, LSTEM models sequence relationships, and CTC manages alignment and

training when you wish to transfer an input sequence (such an image or audio) to an output sequence (like text) without explicitly knowing which part of the input corresponds to which letter. CNN creates feature maps by extracting spatial elements like as shapes, edges, and strokes. To capture context and temporal dependencies, a sequence of feature vectors is fed into a stacked LSTM. many LSTM layers piled on top of one another. The LSTM generates a probability distribution over characters for every time step. Training without explicit input-label alignment is possible using CTC.

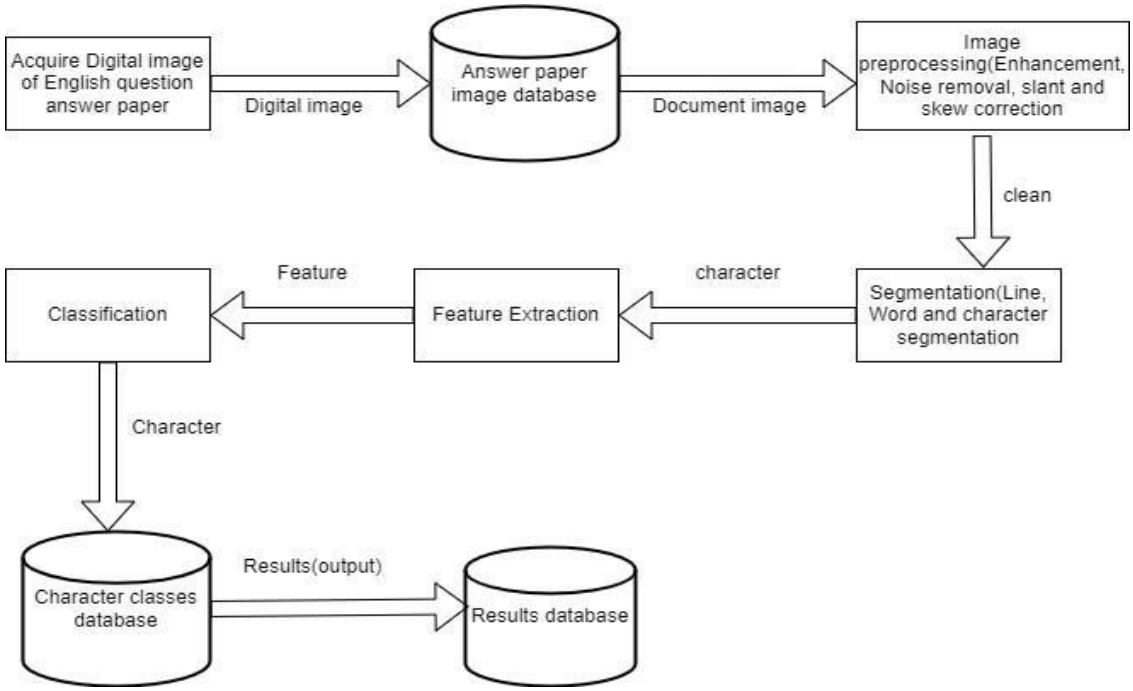

Figure 1. Flow diagram of proposed system

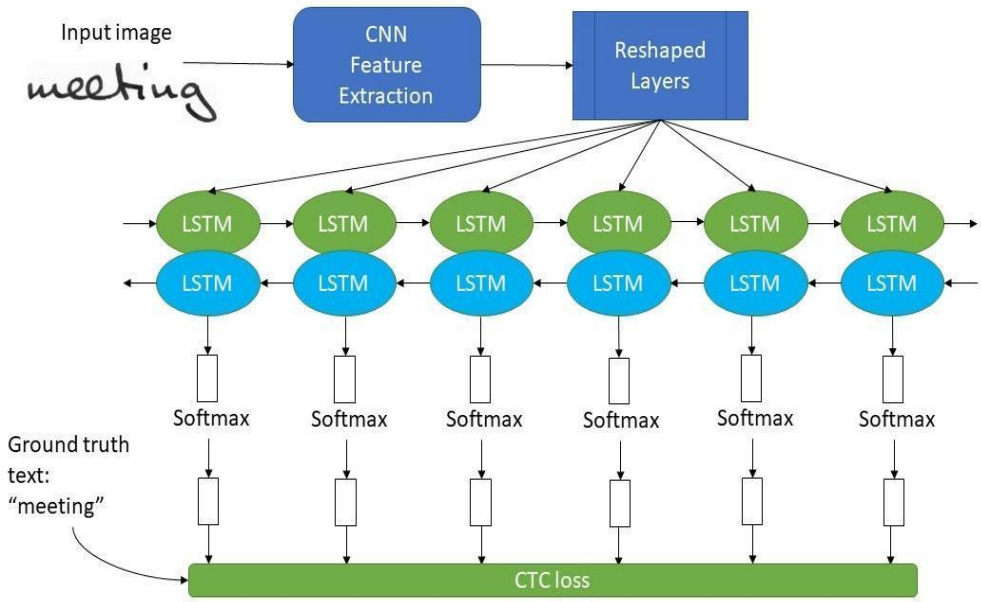

Figure 2. Architecture of system

3.1 Convolutional Neural Networks

The initial stage of the proposed system employs CNNs to extract hierarchical features from the input images. CNNs have proven effective in capturing spatial hierarchies, enabling the model to discern complex patterns and variations in handwriting styles.

3.2 Reshaped layers

The feature vectors extracted by the CNNs are then processed through Reshaped layers to enhance the model's ability to capture relevant spatial information. This preprocessing step optimally prepares the data for further analysis by ensuring that the subsequent LSTM networks

receive well-structured input.

3.3 Multi-LSTM Networks

To capture both spatial and temporal dependencies present in handwritten text, a set of interconnected LSTM networks is employed. Each LSTM network is designed to capture long-range dependencies within the sequence of features, fostering a holistic understanding of the handwritten content.

3.4 SoftMax activation function

The outputs of each LSTM network are passed through Softmax activation functions. This activation function transforms the raw output into a probability distribution over a predefined set of characters, enabling the model to predict the likelihood of each character in the sequence.

3.5 Connectionist Temporal Classification block

The Softmax outputs are then fed into the CTC loss block, which facilitates the alignment of the predicted sequence with the ground truth. The CTC loss enables end-to-end training of the entire system, providing a mechanism for the model to learn the alignment and transcription simultaneously.

4. DATASET USED

IAM Handwriting Database [29-33] serves as a valuable asset for training, evaluating handwritten text recognizers, and conducting experiments pertaining to writer identification and verification. This database consists of unconstrained handwritten English text in diverse forms and is scanned at a resolution of 300dpi, resulting in PNG images containing 256 gray levels. These images include complete forms, text lines, and extracted words. The entire dataset, encompassing forms, text lines, words, and sentences, is accessible for download in PNG format, with each file accompanied by XML meta-information. The text entries in the IAM database are constructed using sentences derived from the LOB Corpus.

IAM Handwriting Database 3.0 incorporates contributions from 657 writers, totalling 1,539 pages of scanned text. It includes 5,685 isolated labelled sentences, 13,353 isolated labelled text lines, and 115,320 isolated labelled words. Notably, the words have been extracted using an automatic segmentation scheme, which was subsequently validated manually.

The segmentation scheme, detailed in prior work by the institute, is explained in the relevant literature. Each image file is accompanied by corresponding label files in XML format, providing segmentation data and a range of predicted parameters derived from preprocessing, as outlined in the relevant literature.

5. ARCHITECTURE

For the model the batch size of 32 was used with learning rate of $5x10^{-4}$ was used with epoch cycles of 1000 iterations and the number of parallel processes or threads used to load and preprocess data the during training phase was 20.

In our CNN architecture, for 1st Convolutional layer 32 filters were used and kernel size of 3x3 was used, Activation function used is Relu, kernel initialization is he normal, having a pool_size of 2 × 2.

For 2nd Convolutional Layer 64 filters were used and kernel size of 3x3 was used, Activation function used is Relu, kernel initialization is he normal, having a pool_size of 2 × 2. Here two max pool with pool size and strides 2 is used. Hence, downsampled feature maps are 4× smaller and are considered as new inputs for that dense Layer was used with Relu as an activation function.

Dropout Layer is also used for randomly setting input units to 0 with a frequency of rate of 0.2 at each step during training time.

In RNN, for 1st layer, LSTM is used with 128 units as input, return Sequence set to true and a dropout rate of 0.25. For 2nd layer, LSTM is used with 64 units as input, return Sequence set to true and a dropout rate of 0.25.

Later, Dense layer is again used, the length of the vocabulary used in a character-to-number mapping (char_to_num). The + 2 indicates the addition of two extra units, for special tokens like start and end symbols or padding, with softmax as activation function.

And finally, CTC Loss layer is used for calculating CTC loss at each step & Optimizer used is Adam with default arguments.

6. SIMILARITY INDEX

The various techniques used for identifying similarity indexes include:

6.1 Bidirectional Encoder Representations from Transformers' similarity index

Bidirectional Encoder Representations from Transformers (BERT) does not inherently have a similarity score calculation; however, it can be used for semantic similarity tasks through fine-tuning or by extracting embeddings from BERT and then applying a similarity metric. Firstly, the input sentences are tokenized using a tokenizer. These tokenized sentences are then used to get contextual embeddings. These embeddings can be pooled or aggregated to obtain fixed-size vectors representing the entire sentences. During fine-tuning, the model learns to map input sentence pairs to a similarity score. During the fine-tuning process, the commonly employed loss function is often mean squared error (MSE) or another loss based on similarity. After obtaining the embeddings, a similarity metric is applied to assess the similarity between the embeddings of two sentences. Cosine similarity is used, which computes the cosine between two vectors.

$$Cosine\ Similarity\ (A,B) = \frac{A.B}{\|A\|\|B\|} \quad (1)$$

where,

$A \cdot B$ is the dot product of the vectors A and B.

$\|A\|$ $\|B\|$are the magnitudes of vectors A and B, respectively.

Another metric that is sometimes used is Euclidean distance or Manhattan distance, depending on the application.

6.2 Jaccard's similarity index

Jaccard similarity serves as a metric for gauging the

similarity between two sets. It is quantified as the ratio of the size of the intersection of the sets to the size of their union. The formula for Jaccard Similarity (J) is expressed as follows:

$$Jaccard's\ (A,B) = \frac{|A \cap B|}{|A \cup B|} \quad (2)$$

| A∩B | : The size (number of elements) of the intersection of sets A and B.

| A∪B | : The size of the union of sets A and B.

The Jaccard Similarity ranges from 0 to 1, where 0 indicates no similarity (no common elements) and 1 indicates complete similarity (all elements are common).

Jaccard Similarity is often used in NLP and text analysis for tasks such as document similarity, clustering, and recommendation systems. It's particularly useful when dealing with sets of items, such as words in documents or user-item interactions.

6.3 TF-IDF Similarity

TF-IDF similarity is an indicator of the similarity between two documents, considering the frequency of terms they have in common. Widely applied in information recovery and text mining, TF-IDF similarity is determined through the following steps:

6.3.1 Term Frequency

For each term in a document, calculate the term frequency, which is the number of times the term appears in the document.

The formula for TF is often given by:

$$TF(t,d) = \frac{Number\ of\ occurences\ of\ term\ t\ in\ document\ d}{Total\ number\ of\ terms\ in\ document\ d} \quad (3)$$

6.3.2 Inverse Document Frequency

For each term, calculate the inverse document frequency, which measures how rare or common a term is across all documents.

The formula for IDF is often given by:

$$IDF(t,D) = loglog\left(\frac{Total\ number\ of\ documents\ in\ corpus\ |D|}{Number\ of\ documents\ containing\ term\ t + 1}\right) \quad (4)$$

The "+1" in the denominator is to avoid division by zero.

6.3.3 TF-IDF score

Multiply the TF and IDF scores for each term to get the TF-IDF score for that term in a particular document.

The formula for TF-IDF is often given by:

$$TF - IDF(t,d,D) = TF(t,d) * IDF(t,D) \quad (5)$$

6.3.4 Cosine similarity

Once TF-IDF scores are calculated for each term in both documents, the documents can be represented as vectors in a multi-dimensional space. The similarity between the two documents is then calculated using the cosine similarity between their TF-IDF vectors.

$$Cosine\ Similarity\ (d1,d2) = \frac{d1.d2}{\|d1\|.\|d2\|} \quad (6)$$

Here, d1 · d2 is the dot product of the TF-IDF vectors of the two documents, and |d1| and |d2| are the Euclidean norms (or magnitudes) of the vectors. The TF-IDF similarity involves calculating TF-IDF scores for each term in both documents and then computing the cosine similarity between the resulting TF-IDF vectors.

6.4 Universal Sentence Encoder similarity index

The Universal Sentence Encoder (USE) similarity is calculated using the USE model developed by Google. The USE is a pre-trained deep learning model that transforms sentences into high-dimensional vectors in a semantic space, capturing their semantic meaning. The similarity between two sentences can be measured using various similarity metrics, such as cosine similarity.

Given two sentences, each sentence is passed through the USE model to obtain high-dimensional vector embeddings for each sentence.

The embeddings capture the semantic information of the sentences in a continuous vector space. The similarity between the embeddings of the two sentences is often measured using cosine similarity. Cosine similarity gauges the cosine of the angle between two vectors and varies between -1 (indicating complete dissimilarity) and 1 (representing identical or complete similarity). The resultant cosine similarity score offers an assessment of the similarity or dissimilarity of two sentences in terms of their semantic content. This score spans from -1 to 1, with a higher value indicating a higher degree of similarity.

7. RESULTS

An automated grading system should assign a score based on how well prepared and written responses match. Nonetheless, five distinct similarity metrics are used. The trained network predicts the similarity score for textual responses using multiple similarity measures. Five similarity indicators are analysed and used to calculate the similarity score between the pre-graded response and the test paper response. The Hybrid OCR model was tested using three different exam papers with six different rated answers ranging from Grade 1 to Grade 6. The tables below display the prediction rates for each paper response along with a comparison with the ratings from the preset grading system.

Table 1 shows the similarity score utilising five similarity measures between six different graded answers and the written response to the test 1 question. Similar results for tests 2 and 3 are shown in Tables 2 and 3.

Tables 4-6 display the accuracy of the predicted response to previously graded responses. With response 1 being the least correct and response 6 being the best, the actual accuracy of the graded papers increases from response 1 to response 6. When we compare the results of each similarity index algorithm, we find that Bert's Algorithm has the highest accuracy (94.44%), whereas NLP, TF-IDF, and Cosine similarity have the lowest accuracy (16.67%). At 33.33%, Jaccard's similarity index accuracy is in the middle. Therefore,

the best algorithm for the suggested hybrid model is Bert's algorithm.

Table 1. Predicted scores for test 1

	Response 1	Response 2	Response 3	Response 4	Response 5	Response 6
NLP	0.927179	0.9372031	0.982409	0.979848	0.963663	1.0
Bert's	0.730277	0.841064	0.823705	0.8621376	0.901318	0.918714
Jaccard's	0	0	0.992063	0.992063	0.994652	0.996551
TF-IDF	0.262446	0.2475293	0.243709	0.355498	0.350407	0.605439
Cosine	0.3506283	0.4577176	0.426335	0.446714	0.332143	0.721433

Table 2. Predicted scores for test 2

	Response 1	Response 2	Response 3	Response 4	Response 5	Response 6
NLP	0.983604	0.977434	0.981429	0.987528	0.989230	1.0
Bert's	0.803445	0.812047	0.849074	0.879383	0.899944	0.927431
Jaccard's	0.96875	0	0	0.990566	0	0.996825
TF-IDF	0.465288	0.329288	0.350162	0.418527	0.341950	0.541592
Cosine	0.512151	0.475031	0.496146	0.508048	0.476731	0.494735

Table 3. Predicted scores for test 3

	Response 1	Response 2	Response 3	Response 4	Response 5	Response 6
NLP	0.970894	0.985005	0.991156	0.988901	0.993377	1.0
Bert's	0.814346	0.842840	0.849382	0.875386	0.909021	0.931793
Jaccard's	0	0.980392	0	0.989898	0.994764	0.996062
TF-IDF	0.392852	0.364887	0.367230	0.341547	0.424030	0.500351
Cosine	0.540748	0.583819	0.595461	0.554688	0.509232	0.598844

Table 4. Accuracy of the predicted test1

	NLP	Bert's	Jaccard's	TF-IDF	Cosine	Expected Range
Response 1	0.0	1.0	1.0	0.0	1.0	0-0.825
Response 2	0.0	1.0	0.0	1.0	0.0	0.825-0.85
Response 3	0.0	1.0	0.0	0..0	0.0	0.85-0.875
Response 4	0.0	1.0	0.0	0.0	0.0	0.875-0.9
Response 5	0.0	1.0	0.0	0.0	0.0	0.90-0.925
Response 6	1.0	1.0	1.0	0.0	0.0	0.925-1.0

Table 5. Accuracy of the predicted test 2

	NLP	Bert's	Jaccard's	TF-IDF	Cosine	Expected Range
Response 1	0.0	1.0	1.0	0.0	1.0	0-0.825
Response 2	0.0	1.0	0.0	1.0	0.0	0.825-0.85
Response 3	0.0	1.0	0.0	0..0	0.0	0.85-0.875
Response 4	0.0	1.0	0.0	0.0	0.0	0.875-0.9
Response 5	0.0	1.0	0.0	0.0	0.0	0.90-0.925
Response 6	1.0	1.0	1.0	0.0	0.0	0.925-1.0

Table 6. Accuracy of the predicted test 3

	NLP	Bert's	Jaccard's	TF-IDF	Cosine	Expected Range
Response 1	0.0	1.0	1.0	0.0	1.0	0-0.825
Response 2	0.0	1.0	0.0	1.0	0.0	0.825-0.85
Response 3	0.0	1.0	0.0	0..0	0.0	0.85-0.875
Response 4	0.0	1.0	0.0	0.0	0.0	0.875-0.9
Response 5	0.0	1.0	0.0	0.0	0.0	0.90-0.925
Response 6	1.0	1.0	1.0	0.0	0.0	0.925-1.0

Figure 2 is a graph illustrating the error rate of a machine learning model across multiple epochs. An epoch signifies a complete iteration through the entire training dataset. The error rate serves as a metric indicating the model's performance on the training data.

The blue line in the graph shows the error rate on the training set. The red line shows error rate on a validation set. Validation set is a holdout set of data that is not used to train models. It is used to evaluate how well the model is generalizing to unseen data.

Figure 3 shows that the error rate on the training set decreases as the number of epochs increases. This means that the model is learning the training data better as it is trained for more epochs. However, the error rate on the validation set also decreases initially, but then starts to increase again after a certain number of epochs. This is a sign of overfitting which

occurs when the model learns the training data too well, and is no longer able to generalize to unseen data. The other possible reasons for the spike in the error rate may be owing to the fact that handwritten text in the IAM dataset contains variations in writing style.

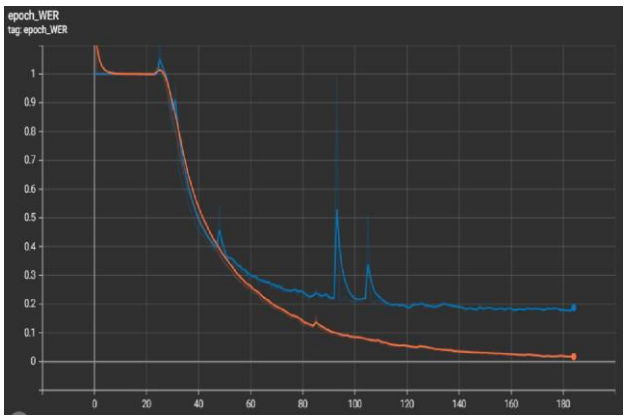

Figure 3. Error rate

The ideal number of epochs to train a model is the point where the error rate on the validation set is minimized. This is typically the point before the error rate on the validation set starts to increase again.

Presented here in Figure 4 is a graphical representation of the CER exhibited by a machine learning model trained on the IAM dataset. The CER serves as a metric evaluating the model's performance on the dataset, calculated by dividing the number of errors made by the model by the total number of characters in the dataset.

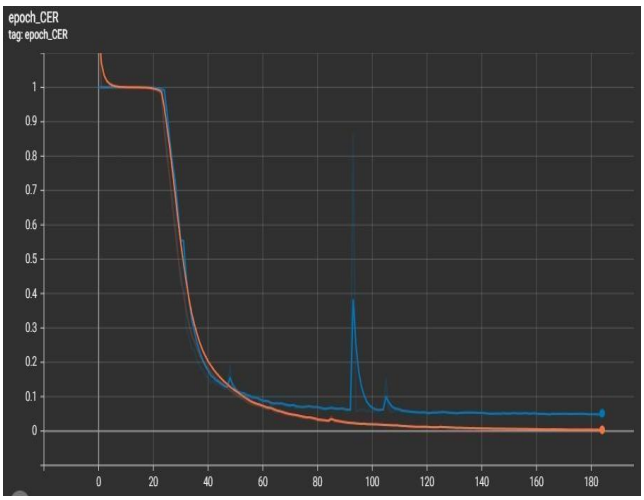

Figure 4. Character error rate (CER)

The x-axis of the graph shows the number of epochs trained. An epoch is one pass through the entire training dataset. The y-axis of the graph shows the CER.

The blue line in the graph shows the CER on the training set. The red line shows the CER on a validation set. The validation set is a holdout set of data that is not used to train the model. It is used to evaluate how well the model is generalizing to unseen data.

The graph shows that the CER on the training set decreases as the number of epochs increases. This means that the model is learning the training data better as it is trained for more epochs. However, the CER on the validation set also decreases initially, but then starts to increase again after a certain number of epochs. This is a sign of overfitting.

Overfitting occurs if model learns the training data too well, and is no longer able to generalize to unseen data. The spike in the CER on the validation set at around 40 epochs is a sign of overfitting. This suggests that the model is starting to learn the specific idiosyncrasies of the training data, rather than the general patterns that are common to all handwritten text.

Another possible reason for the spike is variations in writing style and noisy data. The IAM dataset contains a wide variety of writing styles, from very neat and tidy handwriting to very messy and difficult to read handwriting. This can make it difficult for the model to learn to generalize to all writing styles. Additionally, the IAM dataset contains some noisy data, such as images that are blurred or have smudges. This can also make it difficult for the model to learn to generalize to real-world data.

The possible solutions to address this overfitting are increasing the size of the training data, using a more representative training dataset, simplifying the model, using regularization techniques, such as early stopping or dropout. Also, a data augmentation technique to create a more diverse training dataset can be used to solve the problem of variations in writing style detection.

8. CONCLUSION

This research presented and evaluated a hybrid CNN-LSTM model for HTR with specific application to automated answer scoring. The proposed architecture successfully integrates convolutional layers for spatial feature extraction with multi-LSTM networks for temporal sequence modeling, unified through CTC for end-to-end training. Our experimental evaluation on the IAM Handwriting Database demonstrated that the model effectively learns to recognize handwritten English text, though training analysis revealed overfitting behavior after approximately 40 epochs. The CER analysis indicated that while the model performs well on training data, variations in writing styles and noisy samples in the validation set present ongoing challenges for generalization. These findings suggest that future work should focus on data augmentation techniques, regularization methods such as dropout and early stopping, and potentially expanding the training dataset to improve robustness across diverse handwriting styles. The comparative analysis of five similarity algorithms for automated answer scoring yielded significant practical insights.

The maximum accuracy was attained using BERT-based embeddings with cosine similarity, according to the results. These findings show that for semantic similarity assessment in educational situations, contextualised word embeddings from transformer-based models such as BERT are significantly more efficient than frequency-based or set-based methods. This work's primary contributions are (1) empirical evidence demonstrating the superiority of BERT embeddings for automated answer scoring, (2) a thorough examination of overfitting patterns in HTR models trained on the IAM database, and (3) a comprehensive HTR pipeline combining CNN and LSTM architectures suitable for educational applications.

Nevertheless, it is important to acknowledge the limitations of this study. With just eighteen replies from three papers, the automated scoring evaluation was carried out on a somewhat

modest scale. Larger, more varied datasets covering a variety of topics and question kinds should be used to verify the generalisability of the similarity algorithm rankings. Furthermore, the overfitting shown during training indicates that optimisation of the current model architecture and training procedure might be beneficial.

In conclusion, this research demonstrates that the combination of modern deep learning architectures for HTR with transformer-based similarity metrics offers a viable pathway for automated assessment systems. The significant performance advantage of BERT over traditional similarity measures provides clear guidance for practitioners developing educational technology applications, while the detailed analysis of model training dynamics contributes valuable insights for advancing HTR research.

The following are some future research directions: (1) applying sophisticated regularisation techniques and data augmentation to address overfitting; (2) extending the automated scoring evaluation to larger datasets with a variety of question formats; (3) looking into ensemble approaches that combine multiple similarity metrics; (4) investigating domain-specific BERT model fine-tuning strategies in educational assessment; and (5) implementation of this system for multilingual answers.

REFERENCES

[1] Arica, N., Yarman-Vural, F.T. (2001). An overview of character recognition focused on off-line handwriting. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 31(2): 216-233. https://doi.org/10.1109/5326.941845

[2] Connell, S.D., Jain, A.K. (2001). Template-based online character recognition. Pattern Recognition, 34(1): 1-14. https://doi.org/10.1016/S0031-3203(99)00197-1

[3] Ptucha, R., Such, F.P., Pillai, S., Brockler, F., Singh, V., Hutkowski, P. (2019). Intelligent character recognition using fully convolutional neural networks. Pattern Recognition, 88: 604-613. https://doi.org/10.1016/j.patcog.2018.12.017

[4] Jayasundara, V., Jayasekara, S., Jayasekara, H., Rajasegaran, J., Seneviratne, S., Rodrigo, R. (2019). Textcaps: Handwritten character recognition with very small datasets. In 2019 IEEE winter conference on applications of computer vision (WACV), Waikoloa, HI, USA, pp. 254-262. https://doi.org/10.1109/WACV.2019.00033

[5] Sahlol, A.T., Abd Elaziz, M., Al-Qaness, M.A., Kim, S. (2020). Handwritten Arabic optical character recognition approach based on hybrid whale optimization algorithm with neighborhood rough set. IEEE Access, 8: 23011-23021. https://doi.org/10.1109/ACCESS.2020.2970438

[6] Kumar, M., Jindal, M.K., Sharma, R.K., Jindal, S.R. (2018). Offline handwritten numeral recognition using combination of different feature extraction techniques. National Academy Science Letters, 41(1): 29-33. ttps://doi.org/10.1007/s40009-017-0606-x

[7] Cilia, N.D., De Stefano, C., Fontanella, F., di Freca, A.S. (2019). A ranking-based feature selection approach for handwritten character recognition. Pattern Recognition Letters, 121: 77-86. https://doi.org/10.1016/j.patrec.2018.04.007

[8] Chaithanya, C.P., Manohar, N., Issac, A.B. (2019). Automatic text detection and classification in natural images. International Journal of Recent Technology and Engineering.

[9] Sankaran, N., Jawahar, C.V. (2012). Recognition of printed Devanagari text using BLSTM Neural Network. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), Tsukuba, Japan, pp. 322-325.

[10] Zhang, H., Liu, C., Yang, C., Ding, X., Wang, K. (2011). An improved scene text extraction method using conditional random field and optical character recognition. In 2011 International Conference on Document Analysis and Recognition, Beijing, China, pp. 708-712. https://doi.org/10.1109/ICDAR.2011.148

[11] Vamvakas, G., Gatos, B., Stamatopoulos, N., Perantonis, S.J. (2008). A complete optical character recognition methodology for historical documents. In 2008 The Eighth IAPR International Workshop on Document Analysis Systems, Nara, Japan, pp. 525-532. https://doi.org/10.1109/DAS.2008.73

[12] Karthick, K., Ravindrakumar, K.B., Francis, R., Ilankannan, S. (2019). Steps involved in text recognition and recent research in OCR; a study. International Journal of Recent Technology and Engineering, 8(1): 2277-3878. https://www.academia.edu/download/67425903/Steps_Involved_in_Text_Recognition_and_Recent_Research_in_OCR_A_Study.pdf.

[13] Nguyen, T.T.H., Jatowt, A., Coustaty, M., Doucet, A. (2021). Survey of post-OCR processing approaches. ACM Computing Surveys (CSUR), 54(6): 1-37. https://doi.org/10.1145/3453476

[14] Biró, A., Cuesta-Vargas, A.I., Martín-Martín, J., Szilágyi, L., Szilágyi, S.M. (2023). Synthetized multilanguage OCR using CRNN and SVTR models for realtime collaborative tools. Applied Sciences, 13(7): 4419. https://doi.org/10.3390/app13074419

[15] Souibgui, M.A., Biswas, S., Mafla, A., Biten, A.F., et al. (2023). Text-DIAE: A self-supervised degradation invariant autoencoder for text recognition and document enhancement. In Proceedings of the AAAI Conference on Artificial Intelligence, 37(2): 2330-2338. https://doi.org/10.1609/aaai.v37i2.25328

[16] Liu, Y., Wang, Y., Shi, H. (2023). A convolutional recurrent neural-network-based machine learning for scene text recognition application. Symmetry, 15(4): 849. https://doi.org/10.3390/sym15040849

[17] Dash, B. (2021). A hybrid solution for extracting information from unstructured data using optical character recognition (OCR) with natural language processing (NLP). Research Gate.

[18] Neudecker, C., Baierer, K., Federbusch, M., Boenig, M., Würzner, K.M., Hartmann, V., Herrmann, E. (2019). OCR-D: An end-to-end open source OCR framework for historical printed documents. In Proceedings of the 3rd International Conference on Digital Access to Textual Cultural Heritage, Brussels, Belgium, pp. 53-58. https://doi.org/10.1145/3322905.3322917

[19] Jain, P.H., Kumar, V., Samuel, J., Singh, S., Mannepalli, A., Anderson, R. (2023). Artificially intelligent readers: An adaptive framework for original handwritten numerical digits recognition with OCR Methods. Information, 14(6): 305. https://doi.org/10.3390/info14060305

[20] Martínek, J., Lenc, L., Král, P. (2020). Building an efficient OCR system for historical documents with little training data. Neural Computing and Applications, 32(23): 17209-17227. https://doi.org/10.1007/s00521-020-04910-x

[21] Abbas, S., Alhwaiti, Y., Fatima, A., Khan, M. A., et al. (2022). Convolutional neural network based intelligent handwritten document recognition. Computers, Materials & Continua, 70(3): 4563-4581. https://doi.org/10.32604/cmc.2022.021102

[22] Jatowt, A., Coustaty, M., Nguyen, N.V., Doucet, A. (2019). Deep statistical analysis of OCR errors for effective post-OCR processing. In 2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL), Champaign, IL, USA, pp. 29-38. https://doi.org/10.1109/JCDL.2019.00015

[23] Akhtar, Z., Lee, J.W., Attique Khan, M., Sharif, M., Ali Khan, S., Riaz, N. (2023). Optical character recognition (OCR) using partial least square (PLS) based feature reduction: An application to artificial intelligence for biometric identification. Journal of Enterprise Information Management, 36(3): 767-789. https://doi.org/10.1108/JEIM-02-2020-0076

[24] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., Gao, J. (2021). Deep learning--based text classification: A comprehensive review. ACM Computing Surveys (CSUR), 54(3): 1-40. https://doi.org/10.1145/3439726

[25] Memon, J., Sami, M., Khan, R.A., Uddin, M. (2020). Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR). IEEE Access, 8: 142642-142668. https://doi.org/10.1109/ACCESS.2020.3012542

[26] Marti, U.V., Bunke, H. (1999). A full English sentence database for off-line handwriting recognition. In Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318), Bangalore, India, pp. 705-708. https://doi.org/10.1109/ICDAR.1999.791885

[27] Shi, B., Bai, X., Yao, C. (2016). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(11): 2298-2304. https://doi.org/10.1109/TPAMI.2016.2646371

[28] Graves, A. (2012). Connectionist temporal classification. In Supervised sequence labelling with recurrent neural networks, pp. 61-93. https://doi.org/10.1007/978-3-642-24797-2_7

[29] Natarajan, P., Saleem, S., Prasad, R., MacRostie, E., Subramanian, K. (2006). Multi-lingual offline handwriting recognition using hidden Markov models: A script-independent approach. In Summit on Arabic and Chinese Handwriting Recognition, pp. 231-250. https://doi.org/10.1007/978-3-540-78199-8_14

[30] Senior, A.W., Robinson, A.J. (1998). An off-line cursive handwriting recognition system. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(3): 309-321. https://doi.org/10.1109/34.667887

[31] Plamondon, R., Srihari, S.N. (2000). Online and off-line handwriting recognition: A comprehensive survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1): 63-84. https://doi.org/10.1109/34.824821

[32] Stig, J., Leech, G.N., Goodluck, H. (1978). Manual of information to accompany the Lancaster-Oslo: Bergen corpus of British English, for use with digital computers.

[33] Prasad, D., Gadpal, A., Kapadni, K., Visave, M., Sultanpure, K. (2020). CascadeTabNet: An approach for end to end table detection and structure recognition from image-based documents. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, Seattle, WA, USA, pp. 572-573. https://doi.org/10.1109/CVPRW50498.2020.00294