MITIGATING ADVERSARIAL STYLOMETRY USING LEF-HT AND C2S-CGReLUNN FOR MULTI-AUTHOR WRITING STYLE ANALYSIS

RIYA SANJESH[1], Dr. PAMELA VINITHA ERIC[2]

[1]Research scholar, Department of CSE, [1]Presidency University, Bengaluru, India
[2]Professor, Department of ECE, [2]Presidency University, Bengaluru, India
E-mail: [1]riya.sanjesh@presidencyuniversity.in, [2]pamelavinitha.eric@presidencyuniversity.in

ABSTRACT

The analysis of multi-author writing style is necessary to identify the individual author from collaborative text. Most of the prevailing works in research literature to overcome the adversarial stylometry, have paid less or no attention to punctuation usage and geometric shapes of the handwritten text. Therefore, in this work, Cutmix-Shake-Shake Convolutional Gaussian-error Rectifier Linear Unit Neural Network (C2S-CGReLUNN) classifier-based multi-author identification is implemented. Initially, the handwritten images are collected and pre-processed. Later, the edges are detected using the Lanczos Kernel-based Canny Edge Detector (LK-CED). Next, the geometric patterns are analyzed using Logarithmic Exponential Fit Hough Transform (LEF-HT). Further, the stroke movement is detected using the Reflect Padding-based Fast Fourier Transform (RP2FT). Next, the stylometric features are extracted. Meanwhile, the text is recognized, and from this, the punctuation-based features are extracted, followed by the calculation of punctuation density. Also, the vocabulary richness is identified using the Entropy-based Spatial Indexing Bidirectional Encoder Representation from Transformers (ESI-BERT) technique. Finally, the extracted features, punctuation density of the author, and vocabulary richness identified output are given to the C2S-CGReLUNN for author classification. Thus, the multi-author writing style is analyzed effectively with a prediction accuracy of 99.07%.

Keywords: *Multi-Author Writing Style Analysis, Stylometric Features, Deep Learning, Natural Language Processing (NLP), Geometric Character Analysis, Stroke Movement Identification, and Edge Detection, Text Recognition, Convolutional Nueral Network*

1. INTRODUCTION

Multi-author writing style analysis involves the evaluation of the handwritten text between various authors [1] [2]. The prediction of authorship attribution is done regarding the word choice in the written text to the rhythm of the writing style [3] [4]. However, attributing the authorship becomes complex due to the improper Stylometric analysis [5] and ignoring textual attributes [6]. Although the traditional model used a Convolutional Neural Network (CNN) [7] [8] for the analysis, those techniques disregarded the geometric patterns of the text. The author's lexicon usage regarding the Stylometric features was neglected by prevailing works, which was crucial for NLP [9]. Also, the removal of noise from the handwritten text [10] and stroke movement detection were not considered. Thus, the multi-author classification using C2S-CGReLUNN is proposed in this paper.

1.1 Problem Statement

The research gaps identified in the literature related to adversarial stylometry analysis are:The edges of the handwritten text were not detected in [14], thus the identification of characters became difficult when one character was mistaken for another. Also, it led to higher error rates in author identification.None of the prevailing works concentrated on overcoming the adversarial stylometry of the author regarding punctuation density and the geometric characters like lines or curves for multi-author writing style analysis. Thus, falsification of the authorship regarding the writing style analysis becomes higher. The frequency components (unique rhythm style) were not analyzed in [11]. Thus, ignoring this nuance failed to capture the critical aspects of the author's unique style.

The semantic analysis was ignored in [12], which misinterpreted the meaning of the text with the same semantic field and different diversity. The contributions of the proposed work are, to detect the outline (edges) of the handwritten image, the LK-CED is utilized. This ensures that each character of the handwritten text is differentiated, thus helping in proper author identification.The author's writing style with adversarial stylometry is identified using geometric pattern analysis regarding LEF-HT and punctuation density. The analysis of geometric characters like lines and curves and the punctuation density usage evaluation detect the author even in adversarial stylometric conditions.

The stroke movement (unique rhythmic style) in the handwritten text of the authors is analyzed using RP2FT. Hence, the critical aspects of the unique writing style of the author are predicted more effectively. Words that belong to the same semantic field have different diversity. For example, the word nature might have different diversified words, such as evergreen, serene, fauna, and so on. Hence, this vocabulary richness present in the author's text is analyzed using ESI-BERT. The paper is structured as: Section 2 explains the existing works, Section 3 describes the proposed framework, Section 4 provides performance analysis, and lastly, Section 5 concludes the paper with future recommendations.

2. LITERATURE SURVEY

A contour texture-based approach [11] for offline writer identification is implemented on pre-processed digital handwritten documents. From the pre-processed image, the co-occurrence features of the exterior contours were identified based on Modified Local Binary Pattern (MLBP) and Ink-trace Width and Shape Letters (IWSL) measurements. From these measurements, similarities between the images were determined using the K-Nearest Neighbour rule. Although the author was classified effectively, false positive occurred for the authors with the same handwriting. Multiple author-specific writing style analyses using Generative Adversarial Network [12] (GAN) was used as a style encoder for image quality, and the Recurrent Neural Network (RNN) was used as a text generator for stroke sequence accuracy. The GAN and RNN were combined to produce varied handwriting styles. These printed-style images were further utilized to handle any text content. Thus, the text was recognized, the author was classified accurately based on the text content. However, the minute style could not be recognized by the model. Vertical Attention Network (VAN) was used [13] to analyse the images of handwritten text. Primarily, the Fully Conventional Network (FCN), which extracted the features from the handwritten paragraph image, was used as an encoder in VAN. Then, Optical Character Recognition (OCR) was used in the decoder side of the VAN to recognize the sequence of characters in the paragraph text image. Thus, the handwritten texts were recognized effectively. On the contrary, the model's performance was affected by the artifacts in the input image. A transformer model was integrated for handwritten text-line recognition [14]. The handwritten text images were collected, next, the Residual Network 50 (ResNet50) was utilized to extract high-level features from the image. Then, to bypass the information, temporal embedding was carried out. Further, in the text transcribe, the texts were encoded, and the multi-head self-attention layers were used to learn language-related dependencies. Thus, the robustness across different writing styles was adopted effectively by the model. Yet, the complex script could not be analyzed by this model. Authors of [15] presented a text extraction and recognition model. Initially, the unwanted region from the text image was filtered, here, the text was identified using edge detection and color-based segmentation. Next, the features were extracted from the segmented image. Then, from the extracted features, the characters were recognized using the CNN classifier. Thus, the text was recognized, the writer was classified accurately using CNN. On the contrary, the text with a cluttered background was difficult to recognize, thus affecting the prediction performance. Residual Swine Transformer Classifier (RSTC) [16] for word-level writer identification was proposed, where the handwritten images were taken and pre-processed, regarding resizing and normalization. Then, the local handwritten style information was analyzed using the Transformer Block of RSTC. Afterward, the global information of the handwritten images was encoded using the Identity Branch and the Global Block of the RSTC technique. Thus, by fusing the local and global information, the RSTC identified the writer effectively. However, the text was not recognized properly, thereby leading to improper writer identification. Handwritten Text Recognition

(HTR) in historical documents was proposed in [17]. Initially, the text documents were cleaned and pre-processed. Then, based on Transfer Learning (TL), the recognition model was fine-tuned. Hence, the recognition of text could be done precisely. Next, the data was augmented using geometric distortion. Finally, the Convolutional Recurrent Neural Network (CRNN) was utilized to recognize the text based on TL. Thus, the text in the document was recognized effectively. Yet, the stylometric analysis, which was essential for analyzing the richness of the text, was not performed effectively. Authors of [18] estimated style extraction and HTR for identifying the author. The Style Extractor (SEN) containing convolutional layers and the RNN was utilized to remove the semantic content from the images of handwritten texts and retain the personalized information of the author. Next, the SEN output was processed into a one-dimensional vector. From that, the style vectors were extracted. Finally, the author was determined based on the style vector effectively. However, the edges of the text were not analyzed, which gave ineffective style vectors. [19] integrated writer identification using deep learning. First, the handwritten samples were collected. Then, the quality of the image was improved by pre-processing. The Features from Accelerated Segment Test (FAST) keypoint detector and Harris corner detector were utilized to identify the point of interest in the handwritten text. Next, the patches that represent the local features were extracted. Finally, CNN was used to identify the writer precisely. On the contrary, the suboptimal detection of points of interest led to poor performance in writer detection. [20] investigated handwritten text-based writer identification using a graph-based solution. Initially, the input data was pre-processed for binarization, baseline correction, region separation, and stroke thinning. Then, the pre-processed data was transformed into a graph-based representation. From the generated graph, the essential features were extracted. Subsequently, the extracted features were inputted to the ensemble classifiers, where the writer identification was done. Thus, the experimental results revealed that the developed model had high adaptability and robustness. Nevertheless, this model had considerable training time.

3. PROPOSED MULTI-AUTHOR WRITING STYLE ANALYSIS FRAMEWORK

The framework for the classification of multi-author text forensics using C2S-CGReLUNN for the writing style of the author is depicted in Figure 1. The important steps involved in the proposed work are outline detection, geometric pattern analysis, stroke movement prediction, punctuation density calculation, vocabulary richness identification, and multi-author classification.

3.1 Input Image : The proposed framework regarding the multi-author handwritten analysis starts with the collection of handwritten images (D) of multi-authors from the "IAM Handwritten Forms Dataset". The dataset description is explained in section 4.1. The collected images are represented as given in equation (1).

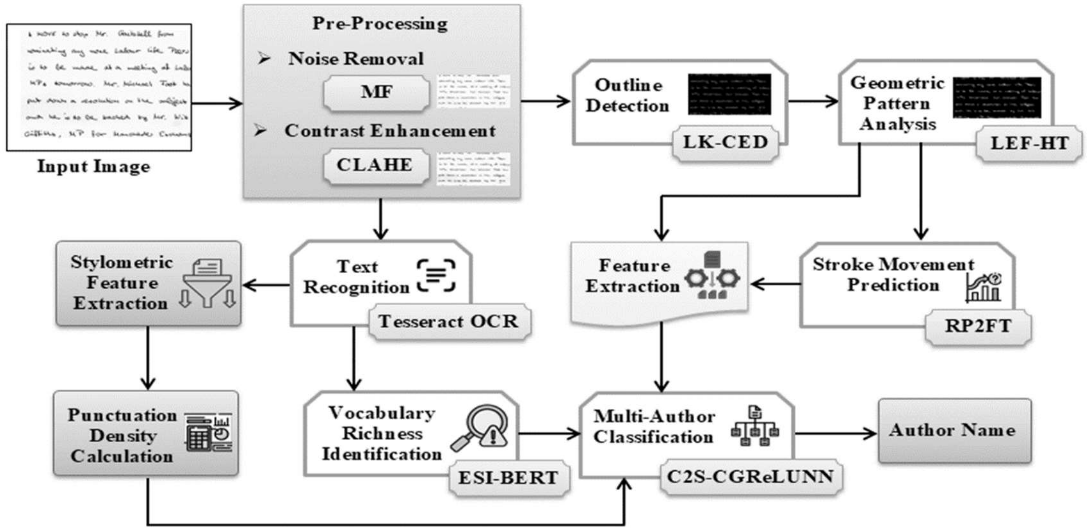


Figure1: Architecture of proposed work

$$D = \left[D_1, D_2, D_3, \ldots\ldots, D_h\right] \tag{1}$$

Where, (h) is the number of (D). Next, pre-processing is carried out to enhance the image quality and to make it suitable for further analysis.

3.2 Pre-Processing : In this preprocessing phase, the image (D) is denoised using Median Filter (MF), and contrast enhancement is done using Contrast Limited Adaptive Histogram Equalization (CLAHE) as follows, Initially, (D) is passed through the MF, which smoothens the image by replacing each pixel with the median of the neighbourhood pixels. Thus, the salt and pepper noise is spread over the image, thus degrading the image. It is then filtered as given below,

$$\alpha = D(i_1, i_2) * \left[{i_1 + i_2}/{2} \right] \tag{2}$$

Where, (α) is the noise-removed image and (i_1, i_2) are the pixel values of (D). Here, as given in equation (2), the median regarding the neighboring pixel values is evaluated and replaced with each pixel. After the removal of impulse noise from the image, the contrast of (α) is enhanced to improve the visibility of the features within the image. First, to enhance (α), it is divided into (c) regions so that over-enhancement and under-enhancement are avoided. Next, the histogram (λ), which shows the distribution of the pixel intensity values of the input image, is calculated regarding the number of grey scales and pixels $(k', \dddot{k})$ as,

$$\lambda = \left({k' - 1}/{\dddot{k}} \right) * \left[\alpha(c)\right] \tag{3}$$

The histogram (λ) that is evaluated as given in equation (3) displays the frequency of occurrence of each pixel. Finally, the contrast-enhanced handwritten image $(\hat{D})$, which is the pre-processed output, is attained by clipping the image as follows,

$$\hat{D} = \left\{ c * \left[D(i_1, i_2) \right] \right\} \times \lambda \tag{4}$$

Here, the image $(\hat{D})$, as given in equation (4), is attained by clipping the image regions and the respective histogram so that the contrast of the image is adjusted. Thus, the contrast-enhanced image $(\hat{D})$ is obtained. Edges are then detected from the image as discussed in the following section.

3.3 Outline Detection: After pre-processing, the outline, which is said to be the edges of the handwritten of $(\hat{D})$, is predicted using the LK-CED. Here, the Canny Edge Detector (CED) is used for outline identification in the proposed framework. The CED produces thin and continuous edges, eliminates the false edges, and adapts to different image characteristics. However, the complex edge structures in the input image with the handwritten text of the author are not detected due to the fixed kernel size. This is because the kernel of fixed size either misses the smaller edges of the handwritten text or fails to capture the nuance of the larger edges. So, to overcome this issue, the Lanczos Kernel (LK), which provides a sharpness-preserving effect, is used in CED to adjust the kernel size. The LK-CED is the modified version of CED done by the proposed work and is explained below.

Initially, the input $(\hat{D})$ is smoothened using the Gaussian filter, which removes the small variations in intensity and improves the overall quality of input. The smoothening is given in equation (5).

$$D'' = \left[\frac{1}{2 * \pi} \right] \times \frac{\exp^{\left(-\frac{\ddot{i}_1^2 + \ddot{i}_2^2}{2 \times \chi^2} \right)}}{\chi^2} \tag{5}$$

Where, (D'') is the smoothened image that is attained regarding the pixels $(\ddot{i}_1, \ddot{i}_2)$, $\pi = (22/7)$, and (χ) is the standard deviation of the pixels in the image $(\hat{D})$. Now, the LK $(\Im)$ that dynamically adjusts the kernel size with the Lanczos parameter (b) and enhances the sharpness of the image is equated as,

$$\Im(D'') = \begin{cases} \dfrac{\sin[\pi * D''] \cdot \sin\left[\dfrac{\pi * D''}{b} \right]}{[\pi * D'']^2} & \forall (D'' < b) \\ 0 & else \end{cases} \tag{6}$$

Thus, as per equation (6), when the smoothened image belongs to the Lanczos parameter, the kernel size is adjusted; or else, it is represented as zero. Then, to preserve the fine details of the image, the magnitude gradient (δ) that represents the strength of an edge at each pixel and the intensity gradient (ε) that helps in retaining the significant pixel intensity variations are evaluated as given in equations (7), and (8), respectively,

$$\delta = \Im(D'') * \sqrt{\left(\partial D''/\partial \ddot{i}_1\right)^2 + \left(\partial D''/\partial \ddot{i}_2\right)^2} \tag{7}$$

$$\varepsilon = arc\tan\left[\left(\partial D''/\partial \ddot{i}_2\right) / \left(\partial D''/\partial \ddot{i}_1\right)\right] \tag{8}$$

The magnitude gradient is calculated based on the evaluated kernel and the smoothened image with respective pixels. And, the intensity gradient is calculated based on the inverse trigonometric process of the tangent function. Now, the weak edges are thinned out by the non-maximum suppression, which removes the pixels that do not come under non-maxima. The outline-detected image (H) is given as,

$$H = \max\left[D''(\delta, \varepsilon)\right] \tag{9}$$

Hence, as given in equation (9), the maximum value of (δ) and (ε) of the input image gives the edge/outline detected image (H). The pseudo-code of LK-CED is given below Algorithm 1.

Further, the characterization of geometric shapes regarding the text of the author is identified.

3.4 Geometric Pattern Analysis

Now, from (H), which is the outline detected image, the geometric patterns, such as curves, lines, and corners, are analyzed using LEF-HT. Here, the Hough Transform, which captures complex geometric shapes, such as curves, lines, corners, and other complex parametric shapes efficiently, is used for the geometric pattern analysis. However, HT is less effective for non-linear shapes present in the input image. This affects the geometric pattern analysis. Therefore, to mitigate this issue, the Logarithmic Exponential Fit (LEF), which is a curve fitting function, is used to parameterize the non-linear shapes of the input image via HT. The method of LEF-HT is detailed below. The edge-detected image (H) is initially transformed using the LEF curve fitting function, which parameterizes the non-linear shapes in (H). This ensures that the edges of the non-linear shapes are analyzed by the LEF function, leading to proper edge detection.

The transformation of the input (H) is expressed as,

$$\widetilde{H} = \left\lfloor t^1 * \ln\left(\left\langle t^2 \times H\right\rangle + t^3\right)\right\rfloor + \left(\exp^H * t^3\right) \tag{10}$$

Where, $(\widetilde{H})$ is the transformed image output, (t^1, t^2) are the scaling vectors that represent the height and angle of the curve of the text, and (t^3) is constant. Hence, using equation (10), the non-linear shapes of the text in the input image are identified. Next, these scaling variables are accumulated in the array (B) to detect the curve patterns. This array creates the profile of the geometric shapes of the text, and this can be further used for analyzing the handwritten text. Finally, the peaks (corners, intersections, and endpoints) that represent the geometric patterns are determined using (B) as,

$$M = \left[B\left(t^1, t^2, t^3\right)\right] \otimes \widetilde{H} \tag{11}$$

Where, (M) is the geometric pattern analyzed image. As given in equation (11), the respective array's scaling factors are multiplied by the transformed image to provide the geometric shapes of the text. Next, the rhythmic style of the author's text is recognized to reflect the individual author's idiosyncrasies in stroke patterns.

3.5 Stroke Movement Prediction

Here, the stroke movement in (M) is evaluated using the RP2FT method. By analyzing the stroke movement, the differences between the writing styles of the authors are identified efficiently. The Fast Fourier Transform (FFT) that identifies the frequency components of strokes is used for the stroke movement prediction. Yet, the boundary effect in FFT causes edge discontinuity, leading to inaccurate stroke movement identification. Therefore, to mitigate this limitation, the Reflect Padding (RP), which mirrors the input and forms continuity during stroke analysis, is used in FFT. The procedure of RP2FT is shown below. To analyze the stroke movement present in the author's handwritten data, the boundary effect is first avoided by extending the input (M) using RP. The RP creates a smooth transition at the boundaries of the input and minimizes the discontinuity of the edges. It is given as,

$$M^* = M\left[\left(j+\hat{j}\right),\left(k+\hat{k}\right)\right] \tag{12}$$

Where, $\left(M^*\right)$is the padded image, $\left(j,k\right)$are the pixel values, and $\left(\hat{j},\hat{k}\right)$are the padding sizes. Equation (12) describes that the pixel values and the padding size are joined in the input to provide a padded image. Now, the frequency components, which distinguish the large-scale patterns like the overall direction of a word or letter and small-scale patterns like fluctuation in the pen's trajectory, are analyzed using the FFT as follows,

$$T = \sum M^* \times \exp^{-\left[\left(M^*\times\pi\right)/2*\left(\hat{j},\hat{k}\right)\right]} \tag{13}$$

Where, $\left(T\right)$is the transformed output with the frequency component. The summation of the padded image with the exponential function gives the frequency components present in the text image. Finally, to attain the stroke movement from $\left(T\right)$, the inversion is done as,

$$\dddot{T} = \frac{1}{2*\left(\hat{j},\hat{k}\right)} \times \sum T * \exp^{\left(M^*\times\pi\right)/2*\left(\hat{j},\hat{k}\right)} \tag{14}$$

Where, $\left(\dddot{T}\right)$is the stroke movement predicted output. Here, equation (14) gives the transposed value of $\left(T\right)$, and this shows the stroke movement of the handwritten text of the author. By this analysis, the dynamics of the writing style are predicted, which helps in the identification of the writer with respect to the characterization of the handwritten style.

3.6 Feature Extraction

Now, the features, such as edges, shapes, colors, textures, area, perimeter, circularity, local binary pattern, pixel intensity, stroke length, angle, and aspect ratio, are extracted from $\left(M\right)$. Also, from $\left(\dddot{T}\right)$, the features like frequency component, peak value, stroke interval, etc. are extracted. The features $\left(R\right)$ are represented as,

$$R = \left[R_1, R_2, R_3, \ldots., R_{m-1}, R_m\right] \tag{15}$$

Where, $\left(m\right)$is the number of features extracted. These features are further used during the multi-author classification.

3.7 Text Recognition

Meanwhile, from the pre-processed image $\left(\hat{D}\right)$, the text is recognized using the Tesseract Optical Character Recognition (Tesseract OCR), which converts the handwritten text within the image into machine-encoded text. The connected components are grouped, and based on this, the text lines and words are segmented. Finally, the patterns are matched and the characters are recognized automatically. The text-recognized output after the pattern matching is given as $\left(G\right)$. Then, from$\left(G\right)$, the stylometric features that show the lexical diversity and frequency of usage of common words are extracted.

3.8 Stylometric Feature Extraction

Now, from the text recognized output$\left(G\right)$, the Stylometric features$\left(K\right)$, such as tense, singular, plural, case, person, synonym, antonym, hyponym, hypernym, noun, pronoun, verb, adverb, participle, conjunction, preposition, interjection, letter and word spacing, comma, period, semicolon, question mark, exclamation mark, and so on are extracted.

$$K = \left\{K_1, K_2, K_3, K_4, \ldots., K_p\right\} \tag{16}$$

Where, $\left(p\right)$is the number of extracted Stylometric features. These features help in verifying the specific writer regarding the piece of writing. Next, by utilizing$\left(K\right)$, the punctuation density is evaluated.

3.9 Punctuation Density Calculation

Here, by utilizing punctuation features, such as commas, periods, semicolons, exclamation, and question marks from$\left(K\right)$, the punctuation density$\left(\eta\right)$is evaluated as,

$$\eta = \frac{Y}{d+Y} \tag{17}$$

Where, $\left(Y\right)$is the number of punctuation marks, and $\left(d\right)$is the number of words in the input text. The punctuation marks used by the author are added and then used for the identification of punctuation density. As the stylometry of each author varies, the punctuation density reflects the writing habit and style preference of the author. Further, the variety and diversity of words used by the author are evaluated.

3.10 Vocabulary Richness Identification

At the same time, from the recognized text (G), the ESI-BERT is used for vocabulary richness prediction. Here, the Bidirectional Encoder Representation from Transformers (BERT), which is an NLP technique that recognizes words with different meanings and usage, is used for embedding the text. However, the complexity issue in BERT slows down the instant writing style analysis. Therefore, to overcome the disadvantage of BERT, the input is indexed using the Entropy-based Spatial Indexing (ESI) function, which sorts the text in lexicographical order. The ESI function organizes the large textual data. In ESI, the entropy is used for identifying the diversity of the text, and indexing-based sorting is carried out to give an ordered sequence of words. Thus, the complexity of the BERT is avoided by indexing the input of BERT using the ESI function. The process of ESI-BERT is explained as,

First, the input (G) is indexed using the ESI function, which organizes the contextual data effectively. By this indexing, the embedding becomes faster. The indexed output (F) regarding the embedding matrix (γ) is equated as,

$$F = -\sum G[\gamma] * \{\log G[\gamma]\} \tag{18}$$

Here, as given in equation (18), the embedding matrix captures the semantic meaning of the words in the input text regarding the context of the text. Thus, by combining (γ) with the input, the meaning of the words is predicted. Now, to weigh the importance of different words used in the context of the author's text present in (F), the Self Attention Mechanism $(\Re)$ is calculated. It is expressed as,

$$\Re = \left(\frac{F}{n}\right) \times \left(\exp^{F} / \exp^{-F}\right) \tag{19}$$

Where, (n) is the total number of (F). Equation (19) shows the contextualized representation of the author's text. Finally, regarding $(\Re)$, weight (w), and bias (x), the vocabulary richness (Z), which shows the nuanced meaning based on the surrounding context, is derived as,

$$Z = \{\Re(G) * w\} + x \tag{20}$$

Thus, as per equation (20), which depicts the dot product between a single word and the key vector of all other words of the text, vocabulary richness is identified. This vocabulary richness (Z) is also given to the proposed classifier for multi-author identification. Finally, the multi-author writing style is classified.

3.11 Multi-Author Classification

In this phase, the C2S-CGReLUNN classifier is used for classifying the multi-author regarding their writing style. Here, the CNN, which is a deep learning model that automatically extracts important and complex features, is used for multi-author prediction. However, the CNN has over-fitting issues due to too many parameters and higher processing time. This affects the performance of the model regarding multi-author classification. Therefore, to overcome the over-fitting problem, the Cutmix-Shake-Shake (C2S) regularization is used in the input of the proposed classifier, and the Gaussian-error Rectifier Linear Unit (GReLU) activation function is used to improve the processing time in CNN. The architectural diagram of the proposed classifier is illustrated in Figure 2.

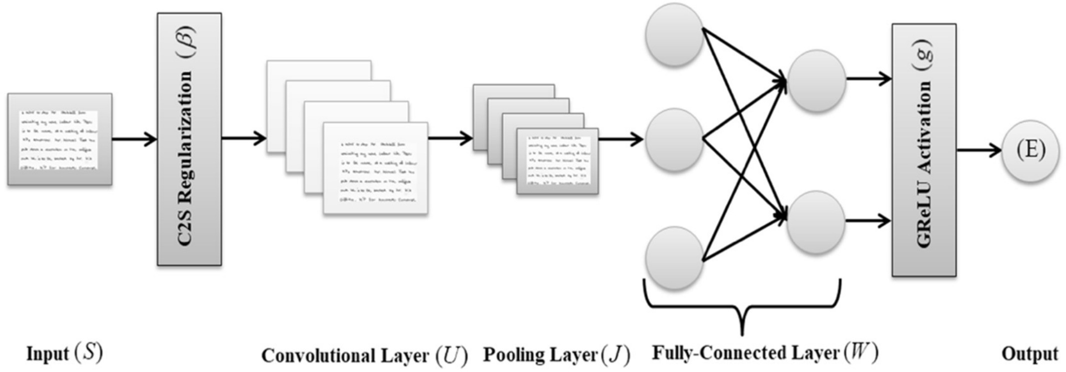

C2S Regularization (β)
GReLU Activation (g)
(E)
Input (S)
Convolutional Layer (U)
Pooling Layer (J)
Fully-Connected Layer (W)
Output

Figure 2: C2S-CGReLUNN classifier

The outputs, such as punctuation density (η), vocabulary richness (Z), extracted features (R) from geometric pattern analysis and stroke movement analysis, and stylometric features (K), are given as input (S) to the proposed classifier. Initially, the input (S) is regularized using the C2S technique, which penalizes the initial weights of the classifier and reduces the over-fitting issue in the classifier. It is expressed as,

$$\beta = [u(S)] \times \{(1-\kappa) * u(S)\} \quad (21)$$

Where, (β) is the regularized output, (u) is the weight value, and (κ) is the binary mask that indicates the cut region. Hence, the over-fitting is avoided by regularizing the input (S) regarding (u) and (κ).

A convolutional layer is the fundamental component of the proposed classifier. It has multiple neurons, and each neuron acts as a kernel, in which the input is processed efficiently. Thus, (β) is split into smaller blocks, and from each block, the features are extracted and mapped. The mapped features are activated using GReLU activation (g), which speeds up the process and reduces the processing time. The output of the convolutional layer (U) is given by,

$$U = \{[\beta \otimes u] + y\} * g \quad (22)$$

$$g = \langle \beta \times \psi(\beta) \rangle + \max(0, \beta) \quad (23)$$

Where, (y) is the bias value and (ψ) is the Gaussian cumulative distribution function. As given in equation (22), the regularized data is multiplied and added with the weight and bias value. Then, it is activated using (g) to give the convolutional output. Equation (23) gives the formula for the activation function, and here, the regularized output is multiplied with (ψ) and then added to the maximum value between the input. Now, in the pooling layer, the spatial dimensions of the feature map (U) are reduced and down-sampled regarding maximum pooling as,

$$J = \max \sum U * q \quad (24)$$

Where, (J) is the output of the pooling layer and (q) is the kernel size. Thus, from equation (24), the most important information is retained effectively by selecting the maximized summation of the input, thereby attaining the maximum value.

The fully connected layer, which is also known as the dense layer of the proposed classifier, connects every input node. Here, a linear transformation is applied to each neuron vector of the input and converted to a single vector. It is denoted as,

$$W = \left|\sum J \times u\right| + y \quad (25)$$

4. Materials and Methods

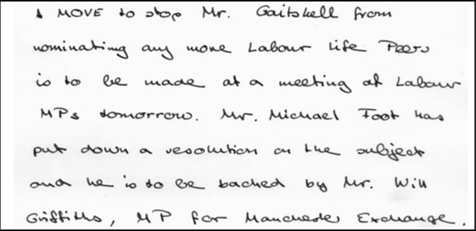
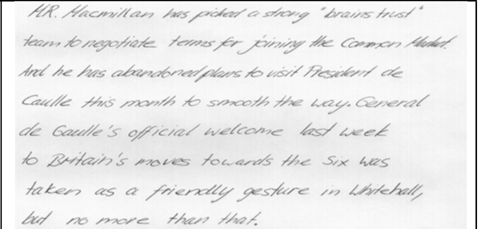
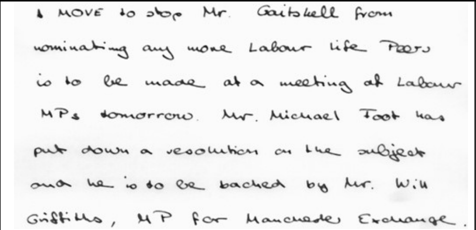
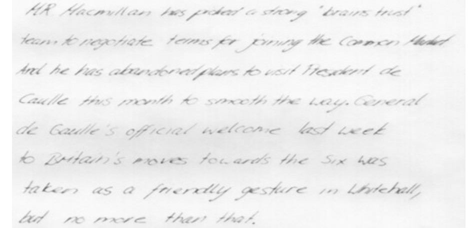
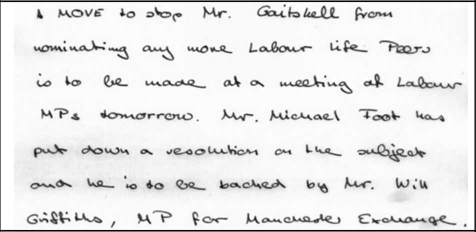
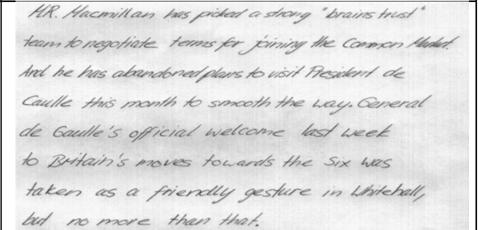
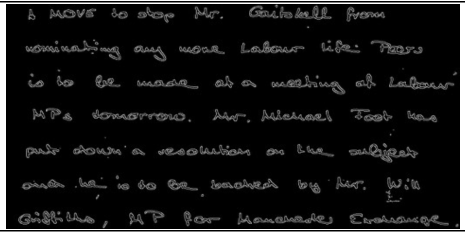
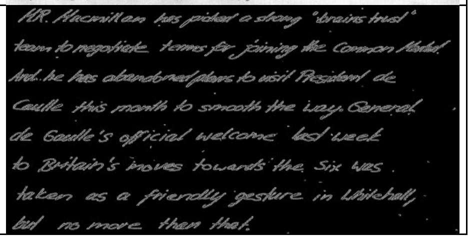
4.1 Dataset Description : The experiments were conducted using the **IAM Handwritten Forms Dataset**, which comprises English handwritten text samples from multiple authors. Each image in the dataset was scanned at 300 dpi resolution and stored in PNG format with 256 grayscale levels. A total of 1,539 handwritten images were used, of which 1,231 (80%) were allocated for training and 308 (20%) for testing. This dataset provides sufficient diversity in handwriting styles, punctuation patterns, and stroke dynamics to evaluate the proposed framework.The image results regarding input, noise removal, contrast enhancement, outline detection, and geometric pattern identification are given in Table 1..

4.2 Experimental Setup

All experiments were implemented using the Python programming environment (v3.10) with TensorFlow and PyTorch libraries. The simulations were executed on a workstation equipped with an Intel Core i7 processor, 16 GB RAM, and an NVIDIA RTX 3060 GPU. The same computational environment was maintained across all model comparisons to ensure fairness and reproducibility.

4.3 Pre-Processing The collected handwritten images were first denoised using a Median Filter (MF) to remove salt-and-pepper noise. Then, Contrast Limited Adaptive Histogram Equalization (CLAHE) was applied to enhance local contrast and highlight finer textual structures. The pre-processed images were resized and normalized between 0 and 1 to standardize the inputs for further analysis.

Table 1: Image Results

Input		
Noise Removal		
Contrast Enhancement		
Outline Detection		

4.4 Outline Detection Using LK-CED : To extract the structural contours of handwritten characters, a Lanczos-Kernel-based Canny Edge Detector (LK-CED) was used. Unlike conventional Canny detectors with a fixed kernel, the LK-CED dynamically adjusts the kernel size using the Lanczos parameter, preserving sharpness and detecting both fine and complex edge structures. Non-maximum suppression was applied to retain only the most significant edges, generating a continuous and thin outline for each character.

4.5 Geometric Pattern Analysis via LEF-HT

From the outline-detected images, geometric patterns such as curves, intersections, and lines were extracted using the Logarithmic Exponential Fit Hough Transform (LEF-HT). The LEF function parameterized nonlinear geometric shapes before applying the Hough Transform, improving shape detection accuracy. The geometric feature vectors obtained from this step represent each author's handwriting geometry and spatial composition.

4.6 Stroke Movement Detection Using RP2FT

The Reflect Padding-based Fast Fourier Transform (RP2FT) was used to analyze stroke rhythm and frequency components that reflect an author's characteristic writing dynamics. Reflect padding mitigated boundary discontinuities during transformation, ensuring smooth frequency analysis. The inverse FFT produced stroke-movement maps that captured temporal patterns such as pen-pressure variation and writing rhythm.

4.7 Text Recognition and Stylometric Feature Extraction : Simultaneously, the pre-processed handwritten images were converted to machine-encoded text using Tesseract OCR. From the recognized text, stylometric features—including lexical categories (nouns, verbs, adjectives), punctuation types, sentence length, and part-of-speech distributions—were extracted to model linguistic style. Punctuation-based metrics such as punctuation density were calculated as the ratio of punctuation marks to total words, reflecting each author's stylistic consistency.

4.8 Vocabulary Richness Identification Using ESI-BERT : Vocabulary richness and semantic diversity were evaluated using the Entropy-based Spatial Indexing Bidirectional Encoder Representation from Transformers (ESI-BERT) model. The ESI module indexed the textual data lexicographically to reduce computational complexity in BERT. Subsequently, contextual embeddings were generated through a self-attention mechanism to quantify lexical diversity and entropy in each author's text.

4.9 Multi-Author Classification via C2S-CGReLUNN : Finally, all extracted features—geometric patterns, stroke movements, punctuation density, stylometric features, and vocabulary richness—were concatenated and fed into the CutMix-Shake-Shake Convolutional Gaussian-error Rectifier Linear Unit Neural Network (C2S-CGReLUNN) classifier. The CutMix-Shake-Shake (C2S) regularization minimized over-fitting by penalizing redundant weights, while the GReLU activation accelerated convergence. The network consisted of convolutional, pooling, and fully connected layers that classified the handwriting samples by author identity. Model performance was assessed using accuracy, precision, recall, F-measure, False Positive Rate (FPR), and False Negative Rate (FNR).

5. RESULTS AND DISCUSSION

Here, the performance assessment of the proposed framework is done. The proposed work is employed in the PYTHON platform.

5.1 Performance Analysis Here, the performance of the proposed models, such as C2S-CGReLUNN, LEF-HT, RP2FT, ESI-BERT, and LK-CED is analyzed and compared with the existing techniques. By this comparison of the proposed models to the traditional methods, the better performance of the proposed work is validated.

Table 2 and Figure 2 give a comparison of the proposed classifier and the existing models regarding multi-author classification. The proposed classifier analyzed the input, such as punctuation density, vocabulary richness, stylometric features, features from geometric pattern analyzed output, and features from stroke movement analysis output from the handwritten text of the author. Also, in the proposed classifier, the over-fitting problem was avoided by regularizing the input using C2S regularization.

Table 2: Comparative Analysis of C2S-CGReLUNN

Techniques	**FPR**	**FNR**
Proposed C2S-CGReLUNN	0.0086	0.0123
CNN	0.0932	0.0978
DBN	0.1894	0.1567
DNN	0.2674	0.3013
ANN	0.3472	0.3902

Moreover, the process time was decreased by utilizing the GReLU activation function in the proposed model. Hence, the proposed C2S-CGReLUNN technique classified the multi-author with a False Positive Rate (FPR) of 0.0086, False Negative Rate (FNR) of 0.0123, accuracy of 99.07%, precision of 99.14%, recall of 99.01%, and F-Measure of 99.04% was obtained. However, the existing CNN, Deep Belief Network (DBN), Deep Neural Network (DNN), and Artificial Neural Network (ANN) attained an average FPR of 0.2243, FNR of 0.2365, accuracy of 96.81%, precision of 97.06%, recall of 96.34%, and F-Measure of 96.28%. These FPR and FNR attained by the existing techniques were higher than the proposed model, and the accuracy, precision, recall, and F-Measure of the existing models were lower than the proposed classifier. Thus, it was proved that the proposed model outperformed existing techniques regarding multi-author identification. Figure 3 reinforces the same understanding in a bar chart.

In the proposed method, the non-linear geometric patterns were identified using the LEF technique. This led to the effective identification of the linear and non-linear geometric pattern by the proposed LEF-HT. For comparing the proposed model with the existing techniques, metrics, such as the Jaccard Index (JI), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) were utilized. The JI quantified the similarity between the sets and focused on the overlapping text. Hence, a higher JI indicated that the proposed model analyzed the geometric pattern of the handwritten text more effectively. RMSE measured absolute error, whereas MAPE measured the relative error of the model. Also, the low values of the RMSE and MAPE showed the technique’s effective performance. Thus, as given in Table 3 and Figure 4, the proposed model achieved a JI of 0.9221, RMSE of 0.0321, and MAPE of 0.3211. Yet, the prevailing HT, Radon Transform (RT), Wavelet Transform (WT), and Affine Invariant Descriptor (AID) attained a JI of 0.8472, 0.7815, 0.7215, and 0.6673, RMSE of 0.0987, 0.1472, 0.2189, and 0.2873, and MAPE of 0.9824, 1.4237, 2.1732, and 2.8739, respectively. The JI of the proposed model was higher than the prevailing techniques, and the RMSE and MAPE values of the proposed LEF-HT were lower than the existing methods. This proved that the proposed method superiorly identified geometric patterns than the prevailing techniques.

The proposed RP2FT identified the stroke movement in the handwritten text with a Mean Squared Error (MSE) of 0.3683 and a Mean Absolute Error (MAE) of 0.1591. However, as given in Figure 5, the existing FFT, Discrete Wavelet Transform (DWT), Short-Time Fourier Transform (STFT), and Empirical Mode Decomposition (EMD) attained an MSE of 0.9972, 2.0913, 2.4673, and 2.8736 and MAE of 0.3927, 0.4864, 0.8952, and 1.0156, respectively. The usage of RP in the proposed model to overcome the boundary effect and discontinuity at the edges of the image led to effective stroke movement analysis. Thus, the MSE and the MAE attained by the proposed technique were lower than existing models. Therefore, the proposed method superiorly performed stroke movement analysis when compared to the traditional models.

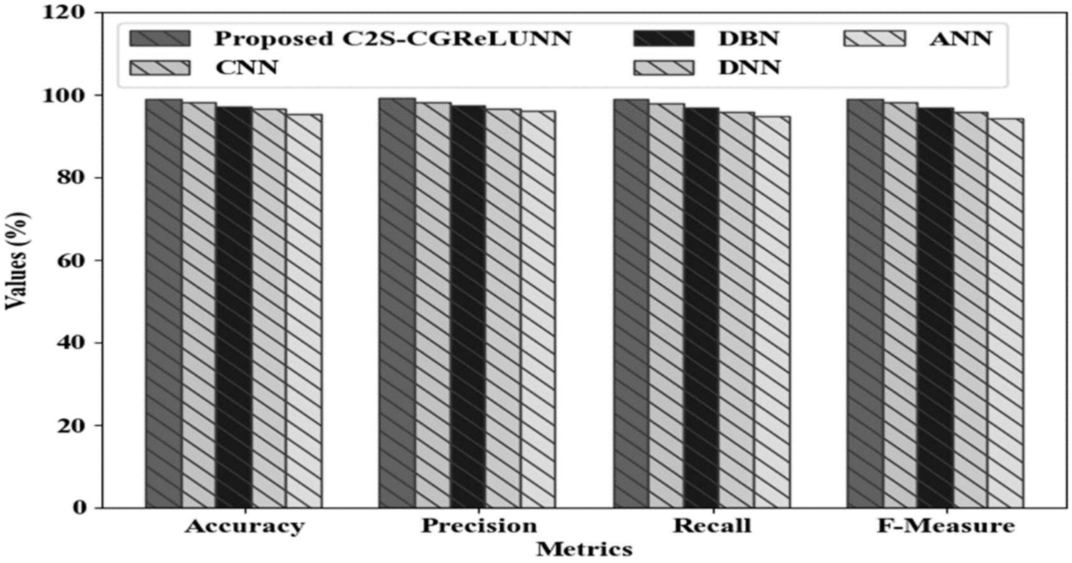


Figure 3: Graphical Comparison Regarding Multi-Author Classification

Table 3: Comparative Analysis of LEF-HT

Methods	Jaccard Index
Proposed LEF-HT	0.9221
HT	0.8472
RT	0.7815
WT	0.7215
AID	0.6673

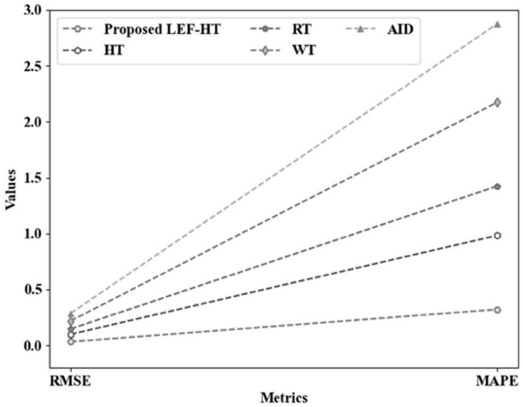


Figure 4: Graphical Comparison of LEF-HT

As given in Figure 6, the proposed ESI-BERT and the existing BERT, Embedding from Language Model (ELMo), Global Vectors for Word Representation (GloVe), and Word2Vec embedded the text with a precision of 98.21%, 97.63%, 97.49%, 96.79%, and 94.92%, and F-Measure of 97.98%, 97.53%, 97.26%, 96.52%, and 94.87%, respectively. The indexing of the input using ESI in the proposed model sorted the text in the lexicographical order, thus reducing the complexity of the model. This improved the vocabulary richness identification in the proposed technique. Thus, the proposed ESI-BERT attained higher precision and F-measure than the prevailing models. This showed that the proposed method outperformed existing models regarding vocabulary richness identification.

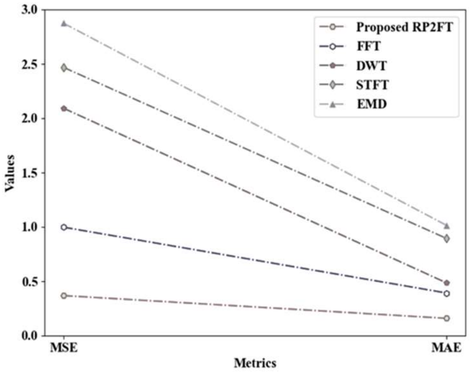


Figure 5: Comparative Analysis of RP2FT

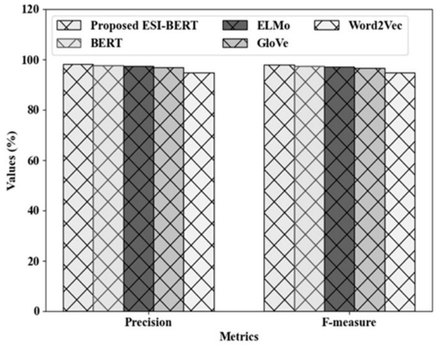


Figure 6: Graphical Comparison regarding ESI-BERT

Table 4: Comparative Analysis of LK-CED

Methods	**SSIM**
Proposed LK-CED	0.9176
CED	0.8863
SO	0.8435
RCO	0.7821
MSA	0.7156

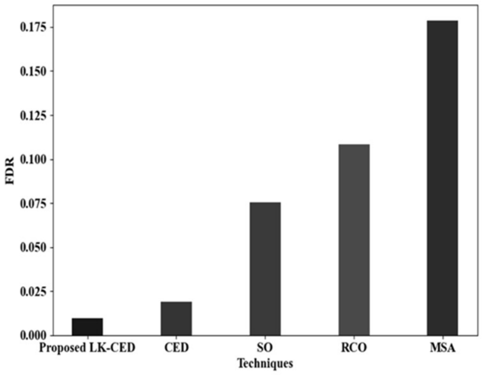


Figure 7: Graphical Comparison of LK-CED

The proposed LK-CED dynamically adjusted the kernel size using the LK function. Thus, the complex edge structures of the handwritten text were captured effectively by the proposed LK-CED technique. Hence, the proposed model attained a Structural Similarity Index Measure (SSIM) of 0.9176 and a False Discovery Rate (FDR) of 0.0097. As shown in Table 4 and Figure 7, the existing CED, Sobel Operator (SO), Roberts Cross Operator (RCO), and Mean Shift Algorithm (MSA) achieved an SSIM of 0.8863, 0.8435, 0.7821, and 0.7156 and FDR of 0.0189, 0.0753, 0.1082, and 0.1786, respectively. Thus, the proposed model attained higher SSIM and lower FDR than existing techniques. Hence, the proposed LK-CED detected the outlines of the handwritten text image more effectively than the traditional model.

Table 5: Related Works Comparison

Study	**Method**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**
Proposed Work	C2S-CGReLUNN	99.07	99.14	99.01
[21]	GAN	98	95	97
[22]	CNN	98.5	-	-
[23]	CNN	98	99	98
[24]	GloVe + CNN	98.67	98	98
[25]	DNN	92.69	-	-

Table 5 describes the comparison of the proposed work and the existing classifiers regarding multi-author writing style classification. In the proposed work, the punctuation density of the handwritten text of the author was evaluated. Also, the geometric patterns, stroke movement, stylometric features, and vocabulary richness were analyzed in the proposed framework. This led to the identification of the multi-author with an accuracy of 99.07%, precision of 99.14%, and recall of 99.01%. However, in existing [21], the Generative Adversarial Network (GAN) processed the image without outline detection, thus attaining a recall of 97%, which was lower than the proposed model. Also, the prevailing work [22] performed the analysis without analyzing the Stylometric feature, which led to an accuracy of 98.5%. This accuracy was less than the proposed technique. In existing [23] and [24], the context was not predicted, thus giving a precision of 99% and 98%, respectively. In prevailing [25], the over-fitting issue presented in the Deep Neural Network (DNN) was not solved, which attained 92.69% accuracy. Thus, the proposed classifier attained better values of accuracy, precision, and recall when contrasted with existing works. This showed that the proposed work effectively analyzed the multi-author identification than existing works.

6. CONCLUSION

The proposed work effectively analyzed the multi-author writing style. Here, the outline of the pre-processed handwritten image was detected using LK-CED with an FDR of 0.0097. The geometric pattern was then identified using LEF-HT with a JI of 0.9221. Next, the RP2FT was used to evaluate the stroke movement, which attained an MAE of 0.1591. Meanwhile, the vocabulary richness from the recognized text was predicted using ESI-BERT with a precision of 98.21%. Finally, regarding the stylometric features and other inputs, the multi-author classification was done using the C2S-CGReLUNN deep learning model with an accuracy of 99.07%. Hence, the proposed model identified the author regarding writing style more effectively. In future, mixed-handwritten documents will also be considered to further enhance the proposed multi-author writing style analysis.

REFERENCES:

[1] F. Alwajih, E. Badr, and S. Abdou, “Writer adaptation for E2E Arabic online handwriting recognition via adversarial multi task learning,” Egyptian Informatics Journal, vol. 23, no. 3, pp. 373–382, Mar. 2022.

[2] N. Triki, “Exemplification and reformulation in expert linguists’ writings: Elaborative metadiscourse between disciplinarity and individuality,” *Journal of English for Academic Purposes*, vol. 71, p. 101407, Jul. 2024.

[3] A. Kumar and P. B. Pati, “Offline HWR Accuracy Enhancement with Image Enhancement and Deep Learning Techniques,” *Procedia Computer Science*, vol. 218, pp. 35–44, Jan. 2023.

[4] V. Ruiz-Parrado, R. Heradio, E. Aranda-Escolastico, Á. Sánchez, and J. F. Vélez, “A bibliometric analysis of off-line handwritten document analysis literature (1990–2020),” *Pattern Recognition*, vol. 125, p. 108513, Dec. 2021.

[5] J. Huertas-Tato, A. Martín, and D. Camacho, “Understanding writing style in social media with a supervised contrastively pre-trained transformer,” *Knowledge-Based Systems*, vol. 296, p. 111867, Apr. 2024.

[6] G. Zhao, W. Wang, X. Wang, X. Bao, H. Li, and M. Liu, “Incremental recognition of multi-style Tibetan character based on transfer learning,” *IEEE Access*, vol. 12, pp. 44190–44206, Jan. 2024.

[7] B. R. Kavitha and C. Srimathi, “Benchmarking on offline Handwritten Tamil Character Recognition using convolutional neural networks,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 4, pp. 1183–1190, Jun. 2019.

[8] Z. Chen, H.-X. Yu, A. Wu, and W.-S. Zheng, “Letter-Level Online Writer Identification,” *International Journal of Computer Vision*, vol. 129, no. 5, pp. 1394–1409, Feb. 2021.

[9] F. Rollo, G. Bonisoli, and L. Po, “A comparative analysis of word embeddings techniques for Italian news categorization,” *IEEE Access*, vol. 12, pp. 25536–25552, Jan. 2024.

[10] S.-B. Lim, Y. Lee, and Y. Song, “Definition and automatic extraction performance analysis of stroke elements in the English Alphabet,” *IEEE Access*, vol. 12, pp. 18931–18938, Jan. 2024

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

[12] M. A. Tusher, S. C. Kongara, S. D. Pande, S. Kim, and S. Bharany, “A harmonic approach to handwriting style synthesis using deep learning,” *Computers, Materials & Continua/Computers, Materials & Continua (Print)*, vol. 79, no. 3, pp. 4063–4080, Jan. 2024.

[13] D. Coquenet, C. Chatelain, and T. Paquet, “End-to-End handwritten paragraph text recognition using a vertical attention

network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 508–524, Jan. 2022.

[14] L. Kang, P. Riba, M. Rusiñol, A. Fornés, and M. Villegas, "Pay attention to what you read: Non-recurrent handwritten text-Line recognition," *Pattern Recognition*, vol. 129, p. 108766, May 2022, doi: 10.1016/j.patcog.2022.

[15] R. Islam, Md. R. Islam, and K. H. Talukder, "An efficient ROI detection algorithm for Bangla text extraction and recognition from natural scene images," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, pp. 6150–6164, Mar. 2022.

[16] P. Zhang, "RSTC: a new residual SWIN transformer for offline Word-Level writer identification," *IEEE Access*, vol. 10, pp. 57452–57460, Jan. 2022.

[17] J. C. Aradillas, J. J. Murillo-Fuentes, and P. M. Olmos, "Boosting offline handwritten text recognition in historical documents with few labeled lines," *IEEE Access*, vol. 9, pp. 76674–76688, Jan. 2021.

[18] Z.-R. Wang and J. Du, "Fast writer adaptation with style extractor network for handwritten text recognition," *Neural Networks*, vol. 147, pp. 42–52, Dec. 2021.

[19] A. Semma, Y. Hannad, I. Siddiqi, C. Djeddi, and M. E. Y. E. Kettani, "Writer Identification using Deep Learning with FAST Keypoints and Harris corner detector," *Expert Systems With Applications*, vol. 184, p. 115473, Jul. 2021.

[20] A. U. Rahman and Z. Halim, "A graph-based solution for writer identification from handwritten text," *Knowledge and Information Systems*, vol. 64, no. 6, pp. 1501–1523, Apr. 2022.

[21] W. Ou, S. H. H. Ding, Y. Tian, and L. Song, "SCS-GAN: Learning Functionality-Agnostic Stylometric Representations for Source Code Authorship Verification," *IEEE Transactions on Software Engineering*, vol. 49, no. 4, pp. 1426–1442, May 2022.

[22] A. S. Abdullah, S. Geetha, A. B. A. Aziz, and U. Mishra, "Design of automated model for inspecting and evaluating handwritten answer scripts: A pedagogical approach with NLP and deep learning," *Alexandria Engineering Journal*, vol. 108, pp. 764–788, Sep. 2024, doi: 10.1016/j.aej.2024.08.067.

H. Khalid and M. Aslam, "Automating the Evaluation of Urdu Handwriting for Novice Writers with Localized Feedback.," *IEEE Access*, vol. 12, pp. 70357–70376, Jan. 2024,

[23] R. Hossain, M. M. Hoque, M. A. A. Dewan, N. Siddique, Md. N. Islam, and I. H. Sarker, "Authorship classification in a resource constraint language using convolutional neural networks," *IEEE Access*, vol. 9, pp. 100319–100338, Jan. 2021.

[24] H. A. A. Hamad and M. Shehab, "Improving the segmentation of Arabic handwriting using ligature detection technique," *Computers, Materials & Continua/Computers, Materials & Continua (Print)*, vol. 79, no. 2, pp. 2015–2034, Jan. 2024.