International Journal of Mathematical, Engineering and Management Sciences
Vol. 10, No. 4, 1032-1054, 2025
https://doi.org/10.33889/IJMEMS.2025.10.4.050

Ram Arti Publishers

Enhanced Offline Writer Recognition System Employing Blended Multi-Input CNN and Bi-LSTM Model on Diverse Handwritten Texts

Naresh Purohit
Department of Computer Science & Engineering,
Government Engineering College Bikaner, Bikaner, Rajasthan, India.
Corresponding author: iteng.naresh@gmail.com

Subhash Panwar
Department of Computer Science & Engineering,
Government Engineering College Bikaner, Bikaner, Rajasthan, India.
E-mail: panwar.subhash@gmail.com

(Received on September 13, 2024; Revised on December 24, 2024; Accepted on February 25, 2025)

Abstract
Since author's writing styles are often ambiguous, writer recognition is an appealing research problem for handwritten manuscript investigation. Pattern identification allows for recognizing the author of a handwritten work. Due to the variety of text visuals, especially handwriting images, author recognition is challenging. Convolution Neural Network (CNN) excels in many fields. This paper presents a complete framework that emphasizes feature engineering and uses a simple, efficient deep neural network. Writer recognition begins with data collection. Programming a technique to generate several database texts enhances the data. To capture and categorize feature information, a multi-path CNN with a Bidirectional Long Short Term Memory (Bi-LSTM) module is developed. By integrating CNN and Bi-LSTMs, the proposed model combines spatial and temporal information, offering a comprehensive representation of handwriting. This synergy makes it particularly effective in handling the variability and complexity of offline writer recognition. Languages are occasionally blended when writing in multilingual regions. The system performed well on two publicly accessible handwritten English language benchmark databases, CVL and IAM, in addition to an in-house bilingual database comprising English and Hindi scripts. This study is unconventional since it addressed offline writer recognition using monolingual and bilingual handwritten text corpora. Analytically, our findings are encouraging compared to previous studies. The model achieves accuracies of 98.78% on the IAM dataset, 98.55% on the CVL dataset, and 99% on an in-house bilingual dataset. These results demonstrate a significant improvement over other state-of-the-art methodologies in offline writer recognition.

Keywords- Bidirectional long short term memory, Convolution neural network, Feature extraction, Handwritten document analysis, Image classification.

Nomenclature

ANN	Artificial Neural Network
Bi-LSTM	Bidirectional Long Short Term Memory
CNN	Convolutional Neural Network
ESVM	Exemplar Support Vector Machine
FAST	Features from Accelerated Segment Test
GMM	Gaussian Mixture Models
GR-RNN	Global-context Residual Recurrent Neural Network
KAS	K-Adjacent Segments
k-NN	k-Nearest Neighbour
LBP	Local Binary Pattern
LoG	Laplacian of Gaussian
LSTM	Long Short Term Memory
PDF	Probability Distribution Functions
RBF	Radial Basis Function
ReLU	Rectified Linear Unit

RNN	Recurrent Neural Network
SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Features
SVM	Support Vector Machine
VLAD	Vector of Locally Aggregated Descriptor

1. Introduction

In biometrics, a person is automatically identified or verified by using biological or behavioural characteristics. Within the domain of cognitive informatics, writer recognition and authentication have become popular topics of research recently, with a variety of implications in the domains of security, historical document analysis, and forensic investigation. Depending on the method of acquisition, one can utilize handwriting either online or offline. Online handwriting utilizes input devices such as cell phones and other gadgets, while offline handwriting relies solely on a pencil, pen, and page. Online handwriting exhibits individual diversity due to multiple parameters such as pen placement, motion, orientation, angle, and pressure (Kumar & Sundaram, 2024). The writer identification system operates in a sequential manner. The initial step entails the collection and pre-processing of data to generate appropriate information set that is then input into the model. A feature extraction method is employed to gather handwriting variations or patterns from this data. Afterwards, a classifier is applied to classify the information with the objective of determining the relevant author. Various investigators have developed writer recognition methodologies for various languages, gathering information from written sources in a particular dialect (Nabi et al., 2023).

Deep learning models are frequently employed in the fields of pattern recognition and image processing. CNN is an artificial neural network that is predominantly used for identifying and verifying subjects of study. ANN is an algorithm that is significantly influenced by biological nerve systems and replicates the functionality of the human brain. CNN exhibits resemblances to traditional ANN in that it consists of neurons that undergo self-optimization via the learning process. Every neuron gets an input and performs a computation.

CNN has demonstrated exceptional performance in numerous machine learning tasks and computer vision (Elbarawy & Ghonaim, 2021). In order to demonstrate writer recognition framework, a deep multi-input CNN is developed with Bi-LSTM module (Sunny et al., 2020). This approach is very effective and text-independent for identifying author attributes in handwritten images. The suggested pre-processing approaches lead to square-shaped image patches that are handwritten. The skeleton of each image patch is also computed. The patches are passed into a deep CNN using two distinct pathways, which each assign a feature representation to every patch. The Bi-LSTM module is used together with feature maps extracted by CNN for the purpose of learning sequences and generating sequences. The later instance is classified using Softmax and SVM algorithms. The main findings of the study are outlined below.

- The hybrid approach is formed by merging local CNN features with the Bi-LSTM sequential module.
- A bilingual handwritten text dataset is developed in-house.
- The effectiveness of the proposed approach has been proven using an in-house bilingual database as well as two publicly accessible databases, namely IAM (Marti & Bunke, 2002) and CVL (Kleber et al., 2013).

The remaining part of the paper is segmented in the following fashion: A few recent approaches that are available in the discipline of offline writer recognition are introduced in Section 2. Details of the datasets are available in Section 3. Detailed descriptions of the proposed fused CNN Bi-LSTM architecture are provided in Section 4. The performance examination of the suggested system is presented in Section 5. The paper is ultimately concluded in Section 6.

2. Related Works

Several mainstream offline writer recognition strategies are examined in this section. Before the emergence of deep learning-based neural networks, traditional strategies that involved manually designing features were used. The texture and shape features of visual data are typically taken into account by handcrafted feature-based techniques.

Six various PDF utilised in Bulacu & Schomaker (2007) to determine the identities of authors in the manuscripts they studied. The PDF were derived from handwritten visuals that exhibit texture and diversity in writing styles. The extreme precision was achieved by integrating two texture attributes (Direction Co-Occurrence PDFs and Run-Length PDFs) and one allograph feature (Grapheme Emission PDF). Jain & Doermann (2011) introduced a new strategy to improve efficiency by using the properties of KAS. In addition, they utilized Euclidean distance to categorize authors and implemented exemplar clustering on the KAS features. On the IAM dataset, they attained a precision rate of 89.6%. In 2014, the same researcher improved the accuracy from 89.6% to 94.1% by integrating KAS characteristics with SURF and contour gradient descriptors (Jain & Doermann, 2014). The LoG filter employed (Wu et al., 2014) along with SIFT descriptors to extract word areas from manuscripts. They conducted their experiments on the HIT-MW Chinese database and achieved a commendable success rate of 95.4%.

Deep learning methods generally provide higher recognition performance compared to feature-driven approaches that are individually constructed. The reason for this can be attributed to the presence of extensive volumes of data, including several features. CNN is a potent deep learning technique that is capable of precisely identifying and classifying visual attributes in data. In recent years, a combination of CNN deep learning models and VLAD encoding, SVM, and other processes have been successful in achieving high accuracy rates in writer recognition by extracting characteristics. To get feature maps from the IAM database, Xing & Qiao (2016) used a CNN framework like LeNet-5 that has five convolution layers and three max pooling layers. Christlein et al. (2017) proposed an additional method of grouping SIFT key points that were retrieved from image data. These grouped key points are used as the image target in their proposed model. An illustration of a deep learning model that can handle unlabelled data is proposed by Chen et al. (2019). They developed a model called WLSR (Weighted Label Smoothing Regularization) that uses a weighted uniform label distribution approach. ResNet-50 (Li & Lima, 2021), a kind of CNN, was employed to achieve an accuracy of 99.2% on the CVL dataset.

A procedure is suggested by Kumar et al. (2022) for distinguishing hollow Hindi characters that remain unchanged regardless of scale, rotation, and deformation. The feature extraction approaches SURF, SIFT, and ORB (Oriented FAST and Rotated BRIEF) are employed using both vertical and horizontal projection. The algorithms used for classifiers include k-NN, SVM, and Random Forest. The Random Forest classifier was applied and acquired an accuracy rate of 91.10%. A method is suggested by Kaur et al. (2020) using the Run Length Smearing Algorithm (RLSA) for finding boundary lines. The technique achieved a segmentation accuracy rate of 96.15% for text and images in Gurumukhi newspapers. The efficacy of a variety of parameters and classifiers for the detection of Gurumukhi newspaper text was assessed in an investigation carried out by Kaur et al. (2022). The strengths and weaknesses of several feature extraction techniques that rely on a variety of classifications are examined. In summary, the multi-layer perceptron achieved its finest recognition accuracy, 96.5%, by employing orthogonal features and a splitting approach ranging from 70% to 30%.

A writer recognition solution (Nabi et al., 2024) for Urdu manuscripts, utilizing a deep learning methodology. The system originated on CNN's VGG-16 model. The dataset consisted of 318 different Urdu writers. The training phase obtained a proficiency of 98%, whereas the testing phase earned a proficiency

of 99.1%. A CNN model is suggested by Maaz & Issa (2020) using images of Arabic handwritten manuscripts from the ICFHR-2012 dataset of 202 writers. Each writer had three texts. The technique suggested had a classification efficiency of 98.24%. A method using FAST key points and a Harris corner detector is suggested by Semma et al. (2021) to identify handwritten points of interest. Deep CNN involves training small patches around vital spots, end-to-end classification, and encoding features from a convolution layer. Experimental investigation uses IAM, QUWI, and IFN/ENIT datasets. Frag-net is a deep neural network model developed by He & Schomaker (2020) that extracts significant writer-specific information based on which identification is performed using a feature pyramid. In a recent study (He & Schomaker, 2021), the authors utilized a RNN to analyse the geographical relationship between a series of items. **Table 1** provides a review of notable works on writer recognition, revealing that the majority of research focuses on methodologies tested on content drafted in dialects using Latin scripts. Although there aren't many studies on other scripts, such as Chinese, Arabic, and Devanagari (Hindi), they have garnered significant attention in recent times.

3. Datasets

Data serves as the fundamental support for any deep learning framework, and the precision of the algorithm is primarily dictated by the standard and nature of the information employed. This section covers a comprehensive overview of the datasets utilized in our exploratory investigation. These datasets consist of two benchmark datasets, CVL and IAM, as well as one in-house bilingual dataset. Specific information regarding each of these is following:

3.1 CVL Dataset

There are 311 authors whose handwritten images are included in the CVL database. These images are written in both German and English. Six of the texts were composed in English, while one was written in German by each participant. The sample paragraph text handwritten image is depicted in **Figure 1**. The database provides the general public with access to both the original version, which includes both printed and written documents, and a condensed version that exclusively contains handwritten texts.

Table 1. Comparative survey of writer recognition methodologies.

Author	Feature extraction approach	Classifier	Dataset(s)
Bulacu & Schomaker (2007)	PDF	Nearest neighbor	Firemaker, IAM
Jain & Doermann (2011)	KAS	Euclidean distance	IAM
Jain & Doermann (2014)	KAS with SIFT	Fisher vector & cosine distance	IAM
Wu et al. (2014)	SIFT	Manhatten distance	IAM, Firemaker
Xing & Qiao (2016)	CNN	Softmax	IAM
Christlein et al. (2017)	CNN & SIFT	VLAD & ESVM	ICDAR 2017
Chen et al. (2019)	ResNet	VLAD & nearest neighbor	ICDAR 2013, CVL
Li and Lima (2021)	ResNet-50	Softmax	In-House
Kumar et al. (2022)	SIFT & SURF	k-NN, SVM	Hindi Hollow Char Dataset
Nabi et al. (2023)	CNN VGG-16	Softmax	Urdu Dataset
Maaz and Issa (2020)	CNN	Softmax	ICFHR-2012
Semma et al. (2021)	CNN & FAST Key Points	VLAD & k-NN	IAM, QUWI
He and Schomaker (2020)	Frag-Net CNN	Softmax	IAM, CVL
He and Schomaker (2021)	GR-RNN	Softmax	IAM, CVL, CERUG-EN
He and Schomaker (2018)	Adaptive CNN	Softmax	CVL, IAM
Christlein et al. (2015)	CNN & GMM Super Vector Encoding	Softmax	ICDAR-2013, CVL
Christlein et al. (2017)	CNN	VLAD & ESVM	CVL, KHATT
Keglevic et al. (2018)	DenseNet	VLAD & nearest neighbor	ICDAR-2013
Adak et al. (2018)	Squeeze-Net CNN	Softmax	In-House Bengali
Sulaiman et al. (2019)	CNN & LBP	VLAD & nearest neighbor	CVL & IAM
Gumusbas & Yildirim (2019)	Capsule-Net	Softmax	CEDAR

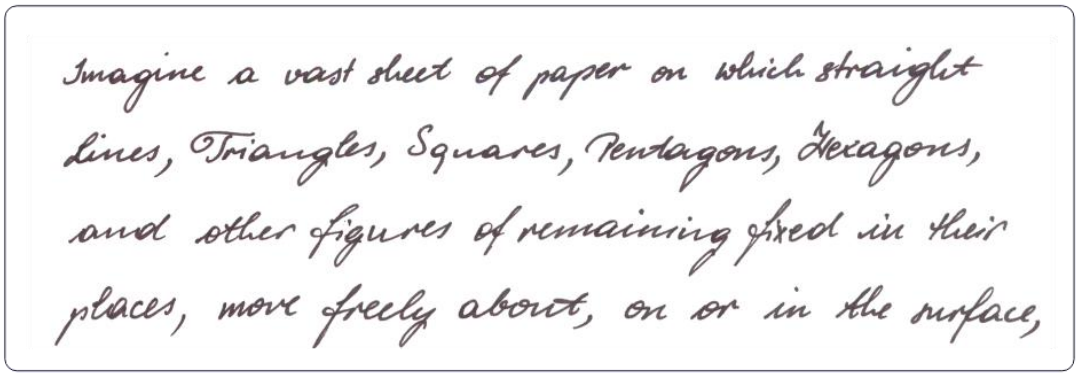

Figure 1. Handwriting sample from CVL dataset.

3.2 IAM Dataset

The IAM is a widely recognized and applied English handwritten dataset, containing 1539 papers and 657 authors. The pages in IAM range from 1 to 59 for each author. **Figure 2** illustrates the sample handwritten text-line data. However, the dataset's most captivating feature is its multifaceted availability of word, line, and page images, which provide an abundant number of methods for recognizing penmanship and identifying words.

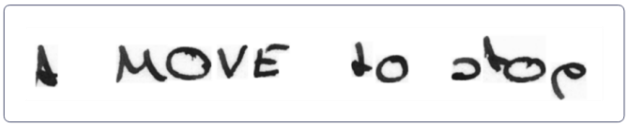

Figure 2. Handwriting sample from IAM dataset.

3.3 In-house Bilingual Dataset

A bilingual handwritten database is created in-house with the collaboration of 50 writers. The sample handwritten paragraph text data is shown in **Figure 3**. Each writer provides six paragraphs. Each paragraph typically consists of 5 to 6 lines of text. The handwritten paragraph content includes phrases in both Hindi and English languages.

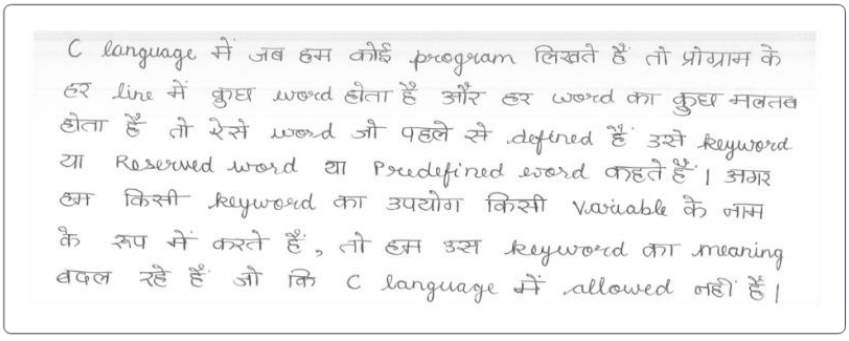

Figure 3. Handwriting sample from In-house bilingual dataset.

4. Proposed Methodology

The outlined approach comprises two fundamental phases: data pre-processing and a hybrid CNN with Bi-LSTM deep neural network. The process flow is depicted in the block diagram shown in **Figure 4**. We employ text-line and word-level segmentation methods to analyse handwritten paragraph datasets and subsequently implement data augmentation to produce a substantial quantity of image patches. The proposed model discovers the patterns for every individual writer in order to perform classification by extracting features from these image segments. Information regarding model architecture and data pre-processing is provided in the subsequent subsections.

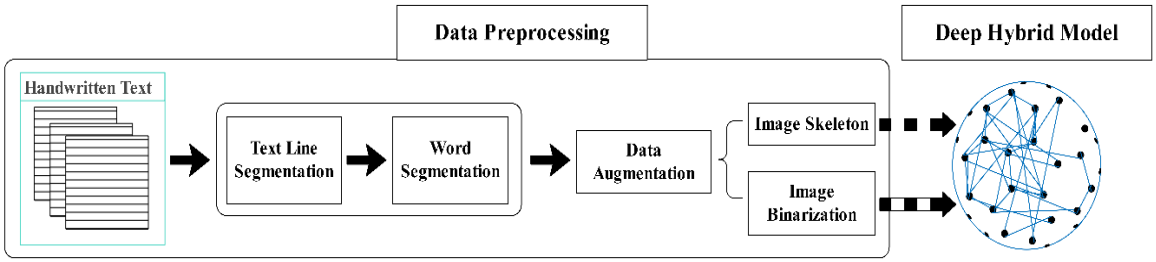

Figure 4. Block schematic of the suggested setup.

4.1 Data Pre-Processing

Data pre-processing is designed to improve information legibility and eliminate irrelevant details in order to obtain visuals that are adequate. In order to build an efficient deep learning model, it is crucial to possess a substantial quantity of training data. Thus, during this stage, a data augmentation method is employed to produce supplementary data from three datasets: IAM, CVL, and an in-house bilingual dataset. The suggested system is trained through augmented paragraph image patches that originate from word images. The CVL dataset incorporates initially present word images. To process the in-house and IAM datasets, a word extraction method was initially implemented to separate the words from the provided handwritten paragraphs or text-line images. **Figure 5** illustrates the workflow of word extraction.

The Otsu (Xu et al., 2011) approach is employed to convert a text image into a binary format, as illustrated in **Figure 5**. The binary text image is then subjected to a Gaussian filter (Wang et al., 2014), which recognizes connected components as words. Subsequently, the binarization process is applied to the filtered, blurry image, identifying the boundary region of the segmented word with a bounding box. Image thresholding is employed to convert the image into a binary format by considering the intensity of each pixel. An issue with basic thresholding is the requirement for manual specification of the threshold amount. Therefore, Otsu's technique is employed to autonomously ascertain the threshold. Otsu's method consists of the following main steps:

- Initially, the inter-class variance is calculated, which quantifies the degree of distinction between the foreground and background areas. In order to determine the between-class variance for the threshold value, the subsequent formula is applied:

$$\text{var}(Th) = P_f . P_b . (m_f - m_b)^2 \quad (1)$$

In Equation (1), P_f and P_b denote the probabilities, while m_f and m_b denote the average gray-scale magnitude values of the background and foreground regions, respectively.

- Compute the average gray-scale magnitude values m_f and m_b in Equation (1) using the following method:

$$m_f = \frac{\sum_{i=0}^{Th-1} i.Hist(i)}{P_f.W.H} \quad (2)$$

$$m_b = \frac{\sum_{i=Th}^{255} i.Hist(i)}{P_b.W.H} \quad (3)$$

In Equation (2) & Equation (3), W and H denote the width and height of the image and $Hist(i)$ represents the gray-scale histogram.

- The best possible threshold value, represented as Th_{opt} in Equation (4), is selected based on the threshold value that produces the highest between-class variation.

$$Th_{opt} = \arg\max_{Th}(\text{var}(Th)) \quad (4)$$

Gaussian blurring involves calculating a weighted average, where neighboring pixels that are more adjacent to the core pixel have a greater influence on the overall average. The Gaussian blur technique is used with a kernel size of 121*121. The mathematical expression for a Gaussian function in one dimension is shown in Equation (5):

$$G(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}} \quad (5)$$

where, σ is the Gaussian kernel's standard deviation and x is the distance to the kernel's horizontal center.

The process of producing newly formed training data from pre-existing training data is known as data augmentation. Without essentially gathering additional data, it enables us to expand the collection and add variability to it. The sequential procedure for the data augmentation strategy is shown in Algorithm 1. The data augmentation procedure has following steps:

(1) In the initial step, the data enhancement technique comprises an array of segmented words from the dataset.
(2) A text line is formed by employing a permutation of 35 arbitrary words. A total of 120 text lines are generated.
(3) To construct an augmented paragraph, six random text lines are permuted. A total of 100 augmented paragraphs are generated for each writer class.
(4) Consequently, a patch cropping strategy is implemented to address augmented paragraph images that are not symmetrical in size. Initially, all paragraph images are set to a specific width and height of 224 pixels, in addition to retaining the aspect ratio. Subsequently, crop the 224*224 image segments from the revised paragraph image data.
(5) Every cropped augmented paragraph image patch is transformed into a single channel by using Otsu's method.
(6) The skeleton image patch is constructed by these binary image patches. Skeletonization (Abu-Ain et al., 2013) is known as a morphological procedure since it provides relevant insights concerning the contours in the image.

Figure 6 illustrates the entire data augmentation process.

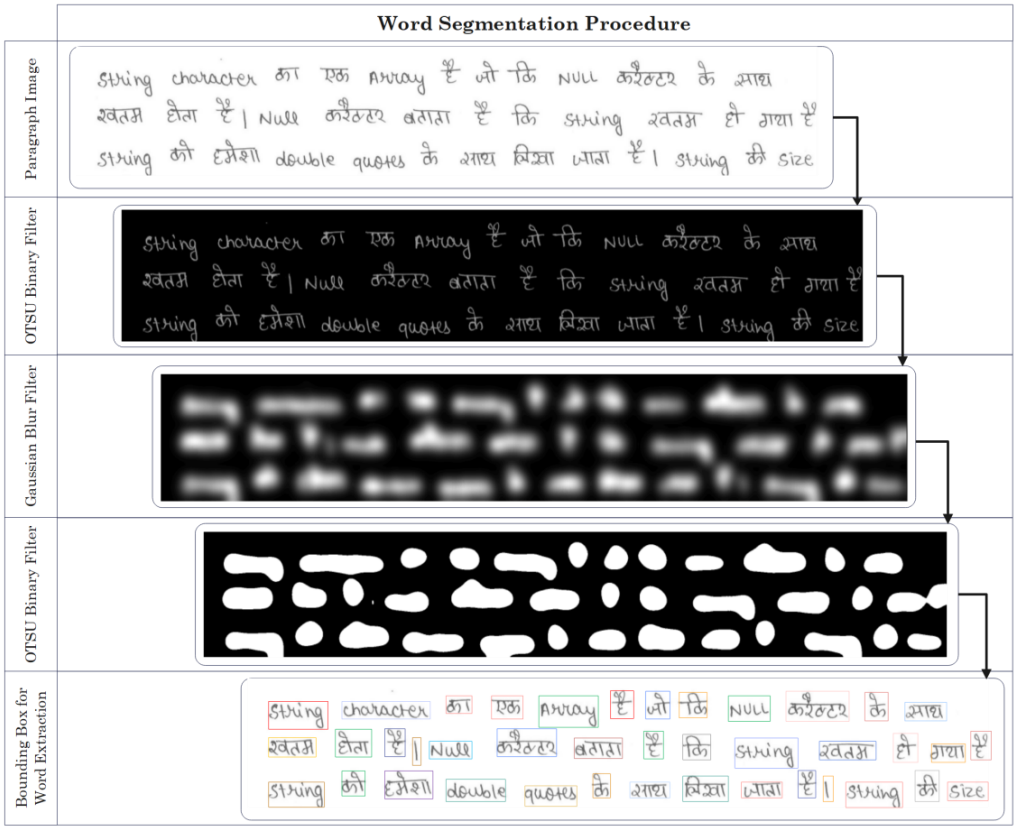

Figure 5. Workflow of word extraction.

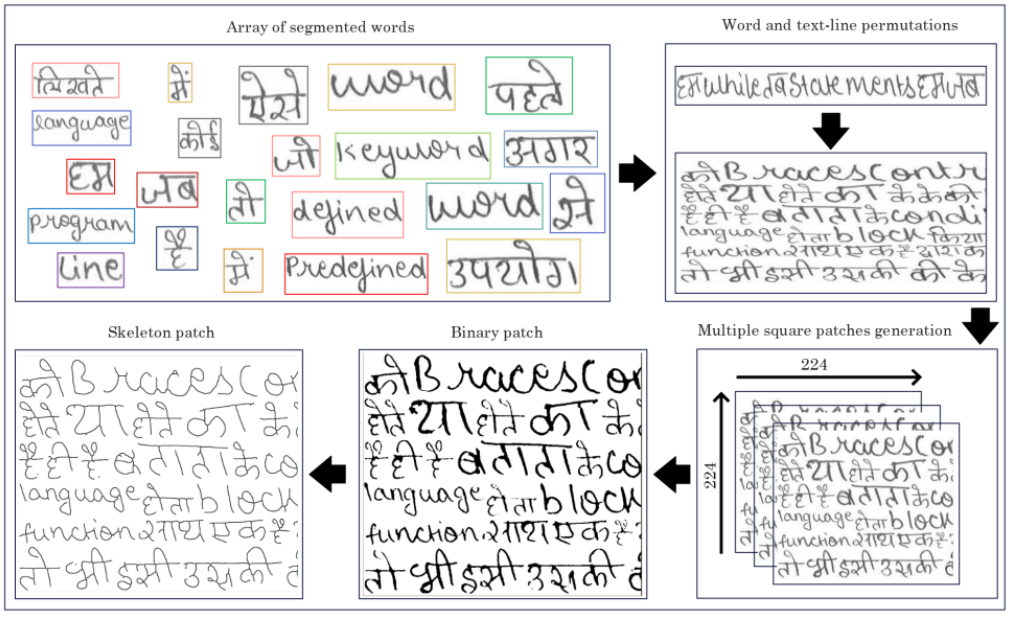

Figure 6. Data augmentation procedure.

Algorithm 1: Data Augmentation

```
for <each writer-class> do
        Input ⇐ SegmentedWords
        for <i=1; i≤120; i++> do
                TextLine ⇐ WordPermutation
        end for
        for <i=1; i≤100; i++> do
                Paragraph ⇐ TextlinePermutation
        end for
        for <each Paragraph> do
                h ⇐ ParagraphHeight
                w ⇐ ParagraphWidth
                h_factor ⇐ 224 / h
                w_new ⇐ (w ∗ h_factor )
                ResizeParagraph ⇐ (w_new, 224)
                crop patches of size (224,224)
                for <for each patch> do
                        apply binarization
                        apply skeletonization
                end for
        end for
end for
```

*__TextlinePermutation__: Permute random 6 text-lines from **TextLine** array

*__WordPermutation__: Permute random 35 words from **Input** array of segmented words

4.2 Multi-Input CNN Bi-LSTM Architecture

The AlexNet (Alom et al., 2018) layout serves as the inspiration for the proposed CNN architecture. It demonstrated outstanding proficiency in the 2012 ImageNet (Krizhevsky et al., 2017) contest. The proposed architecture comprises two paths (Path-A and Path-B), as illustrated in **Figure 7**, since it is a multi-input CNN. Both paths accept input in the form of binary image patches and skeleton image patches, respectively. Convolution and pooling layers in the conventional CNN structure facilitate the easier extraction of features from the input image data. We blend the Bi-LSTM network module, which is a model for processing sequences, with local feature maps generated from a CNN. Bi-LSTM networks are skilled at processing information both in forward and reverse directions, enabling the model to get a deeper comprehension of the interdependencies within sequences. Finally, the network model uses the classification phase to identify the owner of the queried handwritten content. Subsequent sections provide a comprehensive analysis of the model's architectural pipeline.

4.2.1 Multi-Input CNN

Figure 7 illustrates that the proposed model utilizes two routes for feature extraction, both of which share the same structure. The model takes the augmented squared handwritten paragraph image patches from the data pre- processing stage as its input. There are three variations of convolution-pooling layers, as shown in **Figure 8**, each differing in the quantity of filters and sizes. The convolution layer utilizes a collection of adaptable filters, referred to as kernels, to process the input pictures.

The filters or kernels are smaller matrices. Specifically, 5*5, 3*3, and 3*3 shaped filters are employed on three sequential convolution layers. The result generated by this layer is commonly known as feature maps.

This procedure involves sliding over the input picture data and calculating the dot product between the weight of the kernel and the corresponding patch of the input image, as shown in following equation:

$$F_{conv}[i, j] = (I \otimes K)_{[i,j]} \tag{6}$$

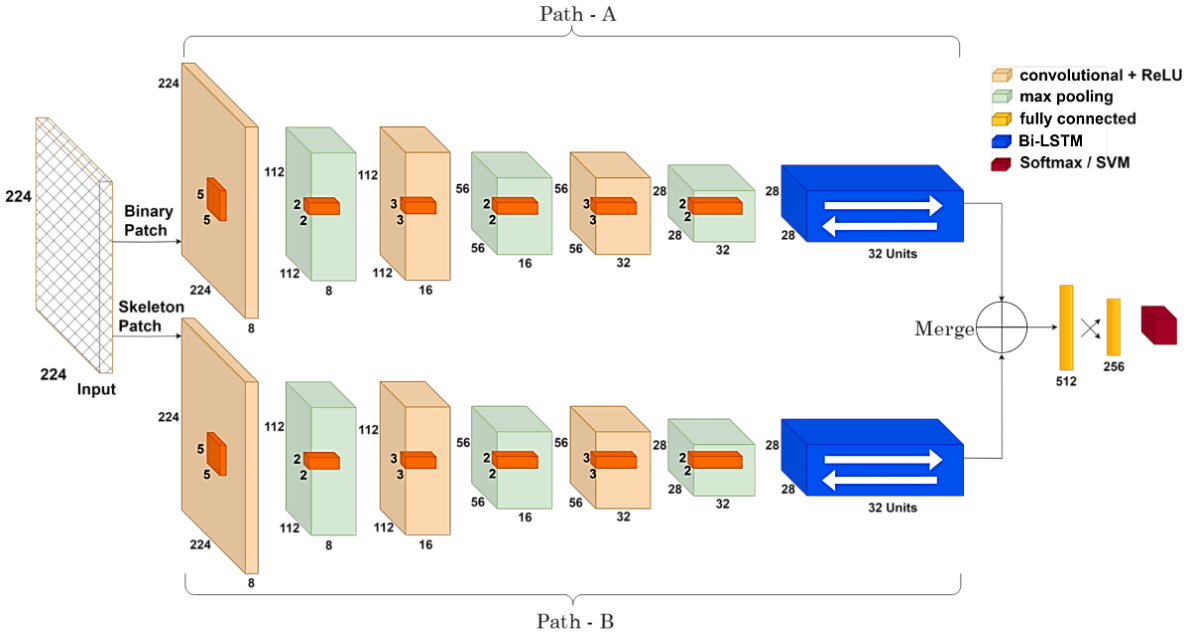

Figure 7. Proposed CNN Bi-LSTM architecture.

In Equation (6), the symbol ⊗ represents a two-dimensional discrete convolution operator. This operator involves sliding the convolution kernel *K* around the input image *I* to perform element-wise multiplication and summation. The outcome is identified as the convolution feature map $F_{(conv)}[i,j]$. Since the input images are squared, their height and width are uniform. Equation (7) delivers the formula utilized for estimating the dimensions of feature maps resulting from the convolution process,

$$D = \frac{(I - F + 2 * P)}{S} + 1 \tag{7}$$

The dimension *D* of the feature maps can be determined by considering the dimensions of the kernel *K*, input *I*, padding *P*, and strides *S*. The imposed kernels prioritize the central area of the image instead of its edges. It could be mitigated by adding padding. Conventionally, zero padding is implemented by adding a column and row of zeros on both edges of the input vector. The rate of motion of the convolution kernel on the input image is controlled by the stride, which demonstrates the number of positions the kernel slides in on the image. The activation function is applied across every outcome of the convolution layer. The ReLU activation function is employed in the output of the convolution layer, and the output ReLU operation is stored in the feature-mapped array $\boldsymbol{F_{acti}}$ in Equation (8).

$$F_{acti} = \mathrm{Re}\,LU\{F_{conv}[i, j]\} \rightarrow \max\{0, F_{conv}[i, j]\} \tag{8}$$

Pooling is a further step in the convolution layer that is employed to reduce the dimensionality of feature maps. Its primary purpose is to reduce the size of the feature maps and enhance computational efficiency. There are consecutive max-pooling layers following each of the three convolution layers, as shown in

Figure 8, each having a receptive field of 2*2 and a stride size of 2. **Figure 7** provides a graphical overview of the dimensions and amounts of kernels and feature maps for each convolution and pooling layer. In the initial convolution layer, we set up eight distinct kernels with a size of 5*5 on the input image's 2D vector, which has a size of 224*224. The model produces a total of eight feature maps measuring 112*112 after the initial convolution using Equation (7).

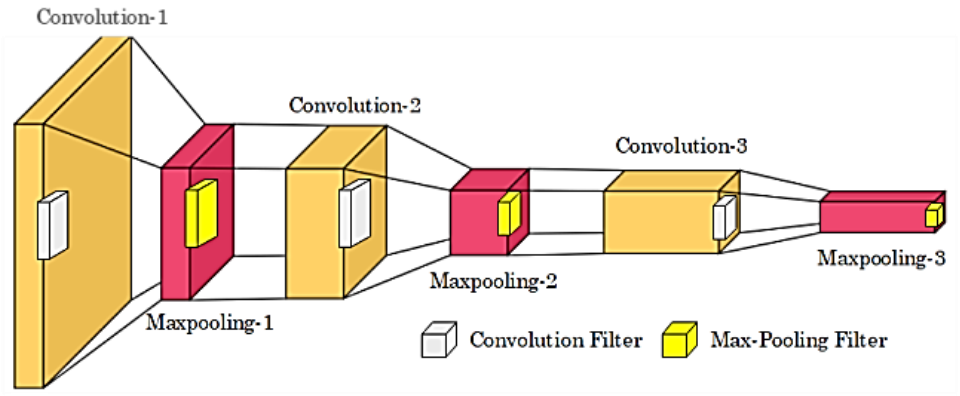

Figure 8. Local feature maps extraction in CNN.

The next pooling layer down-samples these eight feature maps. **Figure 9** illustrates the visualization of the entire process of the initial convolution and pooling layers. Following the three sets of the convolution-pooling layer, the obtained local feature vector with dimensions of 28*28 is combined with the Bi-LSTM network, which will be further elaborated in the subsequent sub-section.

4.2.2 Bi-LSTM

LSTM (Long Short Term Memory) cells are commonly employed to evaluate a portion of formerly acquired features. LSTM networks have the ability to acquire and understand relationships that persist in sequential data. **Figure 10** illustrates the building blocks of a LSTM cell. The input gate, forget gate, and output gate are the three gates that control the memory cell. The input gate controls how data is incorporated into the memory cell. The forget gate controls how information is removed from the memory cell. The output gate manages the data that is emitted from the memory cell. The proposed model incorporates a bi-directional LSTM that efficiently handles sequential data by computing it in both the front and reverse directions. This enables the model to grab the complete context underlying the input sequence, including both past and future information.

To begin, a LSTM cell is utilized to process the current input vector x_k and previous hidden state data vector h_{k-1}, which are passed via the forget gate, multiplied by weight matrices, and then added to a bias, as shown in Equation (9):

$$f_k = sigma(W_f.[h_{k-1}, x_k] + bias_f) \quad (9)$$

where, sigma is sigmoid activation function
W_f is forget gate weight vector
$h_{k-1}$is previous hidden state
x_k is current input
$bias_f$ is forget gate bias

The input gate has accountability for facilitating the entry of essential data into the LSTM cell state. To regulate the state of the cell, Equation (10) illustrates the output i_k of the input gate.

$$i_k = sigma(W_i.[h_{k-1}, x_k] + bias_i) \quad (10)$$

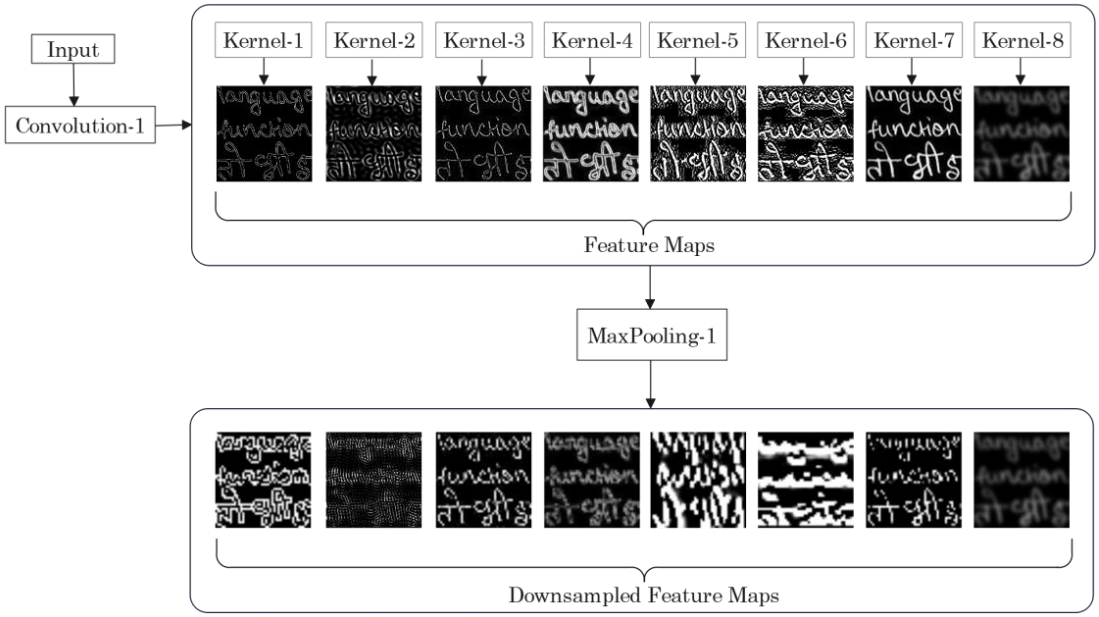

Figure 9. Feature maps visualization - First phase of convolution & max-pooling.

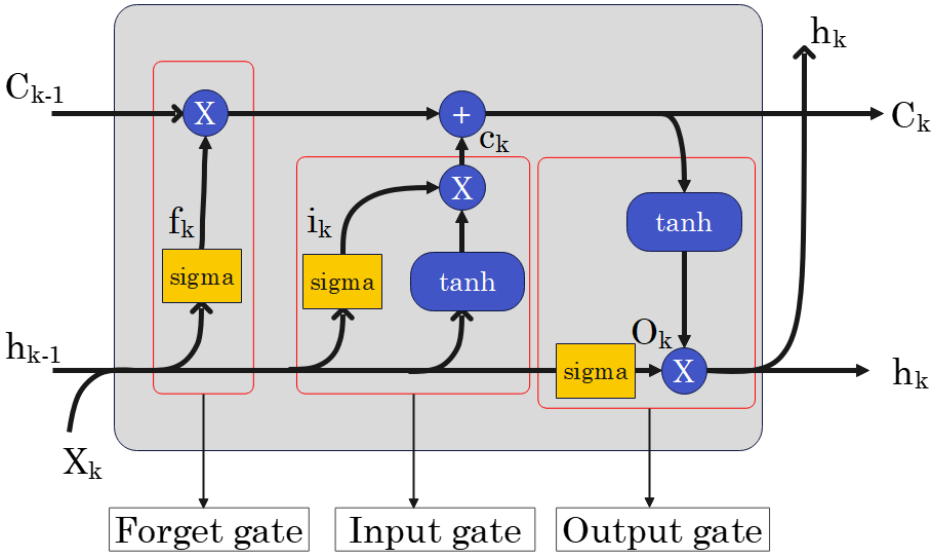

Figure 10. LSTM cell architecture.

Finally, the insightful data $\boldsymbol{c_k}$ is obtained by Equation (11) and Equation (12) by multiplying the vector values by the regulated values.

$$\hat{c} = \tau(W_c.[h_{k-1}, x_k] + bias_c) \tag{11}$$

$$c_k = f_k.c_{k-1} + i_k.c_k \tag{12}$$

where, τ is the hyperbolic tangent (tanh) activation function. The output gate retrieves the hidden state h_k of the current time stamp. The output gate consists of two parts. The *sigma* function, similar to the other two gates, ascertains the amount of pertinent information required. Afterwards, the revised cell state is combined using the τ activation function, which takes the output of the *sigma* function as input $\boldsymbol{o_k}$, as shown in Equation (13) and Equation (14):

$$o_k = \tau(W_0.[h_{k-1}, x_k] + bias_0) \tag{13}$$

$$h_k = o_k.\tau(c_k) \tag{14}$$

From Equation (10) to Equation (14)
[W_i, W_c, W_o] are weight matrices of input gate, output gate respectively.

[$bias_i$, $bias_c$, $bias_o$] are biases of input gate, output gate respectively.

The LSTM is capable of operating in a backward manner, allowing the retention and transfer of characteristics from state-n to state-2 and then to state-1. Bi-LSTM model is created by merging these two methods, as shown in **Figure 11**. Bi-LSTM is preferred, since can capture contextual information from both past and future time steps in a sequence. By capturing temporal dependencies in forward and backward directions, making the combination of CNN Bi-LSTM robust for writer specific patterns. The local features retrieved from the proposed CNN are combined with the Bi-LSTM cell units, as shown in **Figure 7**. The input size of the 32 units of Bi-LSTM cells is 28*28, which corresponds to the feature maps of the last pooling layer of CNN. As depicted in **Figure 11**, it is composed of two LSTM layers, one of which processes the sequence forward and the other backward. Every layer preserves its own memory cells and hidden states.

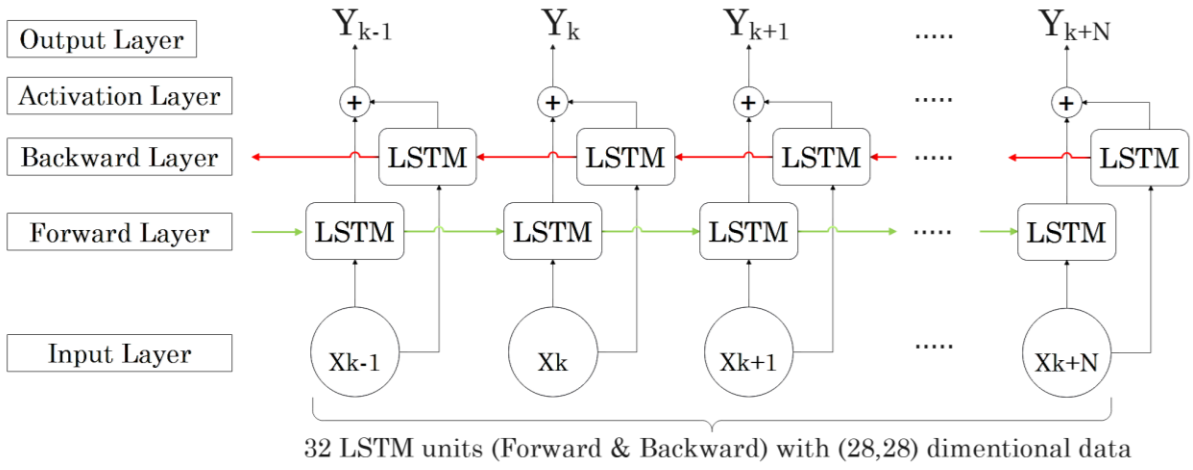

Figure 11. Bi-LSTM configuration.

The sequence-to-sequence data processing in Bi-LSTM involves the following phases:

- **Forward Pass:** The set of inputs is sent from the initial to the finalized time step to the forward LSTM layer. At every interval, the forward LSTM employs the present input value, the prior hidden state, and the memory cell to figure out its concealed state and modify its memory cell.
- **Backward Pass:** Concurrently, the supplied sequences are sent to the reverse LSTM layer in the opposite direction, beginning with that final step and progressing to the initial one. The reverse LSTM, like the forward pass, calculates its concealed state and updates its storage cell with the present input and the previous concealed state and storage cell.
- After the forward and reverse passes, the concealed states from both LSTM layers are combined at the activation layer for every step.
- To calculate the output Y_k at time k, both activations forward $\overrightarrow{x^{\{k\}}}$ and backward $\overleftarrow{x^{\{k\}}}$ are utilized with bias b_y as shown in Equation (15).

$$Y_k = W_y[\overrightarrow{x^{\{k\}}}, \overleftarrow{x^{\{k\}}}] + b_y \tag{15}$$

The output vector Y_k obtained by utilizing 32 Bi-LSTM units for both paths A and B, as illustrated in **Figure 7**, and subsequently blended together. The blended data vector is transformed into a one-dimensional vector and then moved to the next module. This module consists of fully connected layers that are utilized for the assignment of writer classification. The details of this assignment are extensively explained in the subsequent part.

4.2.3 Classification

Thus far, feature extraction procedures have been completed for the model. Next is the writer classification phase. The Fully Connected Layer sometimes referred to as the Hidden or Dense Layer, serves as the final layer in the suggested model. A prior Bi-LSTM layer obtains a one-dimensional combined data vector as the input. The output of the dense layer is generated by applying an affine function in Equation (16) followed by a non-linear function ReLU.

$$f(affine) = W.x + b \tag{16}$$

where, W, x and b are the weight vector, input vector, and bias, respectively, for each node in a dense layer. As depicted in **Figure 7** the model has two dense layers with 512 and 256 nodes subsequently for model training. The information coming from the last dense layer is separately fed into Softmax and SVM classifiers to analyze the probability distribution across the entire collection of writer classes. SVM classifier is employed with the RBF kernel to effectively deal with data that cannot be separated linearly by transforming the input into higher dimensions. The RBF kernel is defined by Equation (17):

$$RBF(a,b) = \exp\left(-\frac{(a-b)^2}{2\sigma^2}\right) \tag{17}$$

where, σ indicates the variance and $(a\text{-}b)^2$ symbolizes the Euclidean distance between two points a & b. The comprehensive procedure of the suggested model is provided in Algorithm 2.

5. Experiments & Results

The proposed approach examines the impact of each individual stage. The local features obtained from a multi-input CNN are combined with Bi-LSTM model. In addition, the impact of utilizing other classifiers, namely Softmax and SVM, was also investigated on the suggested model for the writer recognition challenge. Two benchmarks and one in-house-produced bilingual dataset were employed for model

evaluation. IAM and CVL are two benchmark datasets that contain handwritten text images with the contributions of more than 300 writers with English and German scripts. The in-house bilingual dataset is made up of a mixture of English and Hindi scripts, with the contribution of 25 writers. To evaluate the proposed model's performance and model training, initially 224*224 sized augmented paragraph image patches are constructed from 20 writers, as shown in **Table 2**. The augmented image patches are split into 70:30 ratios for training and evaluation of the model.

Algorithm 2: Proposed Model

for <each writer-class> **do**
    Path(A) ⇐ BinaryInput
    Path(B) ⇐ SkeletonInput
    for <each bainay-patch in Path-A> **do**
        F1 ⇐ FE(BinaryInput)
    end for
    for <each skeleton-patch in Path-B> **do**
        F2 ⇐ FE(SkeletonInput)
    end for
    FusedFeatures ⇐ F1 ⊕ F2
    HiddenLayer1(512Nodes) ⇐ FusedFeatures
    HiddenLayer2(256Nodes) ⇐HiddenLayer1(512Nodes)
    Softmax ⇐ HiddenLayer2(256Nodes)
    SVM ⇐ HiddenLayer2(256Nodes)
end for
procedure FE (Input)
    fmap1 = (Input) ⊙ kernel(5, 5)
    down1 = maxpooling(fmap1)
    fmap2 = (down1) ⊙ kernel(3, 3)
    down2 = maxpooling(fmap2)
    fmap3 = (down2) ⊙ kernel(3, 3)
    down3 = maxpooling(fmap3)
    FLSTM = LSTM (down3) ⊗ 32
    BLSTM = LSTM (down3) ⊗ 32
    BiLSTM = (FLSTM) ⊕ (BLSTM)
    return (BiLSTM)
end procedure
*⊕ : Merging
* ⊙ : Convolution
* FLSTM : Forward LSTM
* BLSTM : Backward LSTM
* BiLSTM : Bidirectional LSTM

Table 2. Patch generation for datasets.

Dataset	Total no. of image patches	Training patches	Testing patches
CVL	14000	9861	4227
IAM	11263	7883	3379
Bilingual	12724	8906	3817

The top-N performance of model is employed to assess the frequency with which the predicted class aligns with the top-N values of the classifier's distribution for each dataset. Top-1, Top-2, and Top-3 performance were measured for all datasets with Softmax and SVM classifiers as shown in **Table 3** and **Table 4**. Since the model consists of two input paths, Path-A and Path-B. Path-A examines binarized image patches, while Path-B utilizes skeletonized image patches. Once the skeleton data is located, a dilation operation is employed using a 3*3 kernel to fine tune the image patch. The target or label class vector is subsequently transformed into a binary class matrix. The arrangement of convolution and pooling layers in both pathways is identical to the one shown in **Figure 7**. Both layers are tuned for two hyper-parameters, stride and padding. The padding for all three convolution layers is set to 'SAME', which means that the input is half-padded to assure that the filter is applied to all parts of the input. Stride is set to one for all convolution layers. The Bi-LSTM is constructed by arranging 32 LSTM cells in both forward and backward orientations following the previous pooling layer. Two fully connected layers are deployed after Bi-LSTM module with 512 and 256 nodes, respectively. Given that this assignment involves classifying many classes, integer values are assigned to the target labels. The sparse categorical cross-entropy loss approach is utilized to train the proposed model, as depicted in **Figure 15**.

Table 3. Accuracies using SoftMax classifier.

Dataset	Top-1(%)	Top-2(%)	Top-3(%)
CVL	98.50	99.95	100
IAM	98.75	99.70	99.97
Bilingual	99.00	99.92	100

Table 4. Accuracies using SVM classifier.

Dataset	Top-1(%)	Top-2(%)	Top-3(%)
CVL	98.55	99.12	99.47
IAM	98.78	98.16	97.45
Bilingual	98.97	98.89	99.1

SoftMax and SVM, two classifiers applied separately for writer classification. The model performance for the top-1, top-2, and top-3 combinations on each epoch is illustrated in **Figures 12, 13,** and **14**. The model's efficiency is assessed by employing a confusion matrix to determine the top-N accuracies across all applied datasets, as depicted in **Figure 16**. We also conducted a comparative analysis of our proposed technique with other existing methods using the IAM and CVL datasets. The results of this analysis are presented in **Table 5** and **Table 6** for the relevant datasets. The proposed hybrid CNN with Bi-LSTM learning pipeline has exceptional results compared to the majority of other supervised methods.

Table 5. Performance comparison with state-Of-The-Art on IAM dataset.

Author	Feature extraction approach	Top-1(%)
Khalifa et al. (2015)	Graphemes with codebook	92
Hadjadji & Chibani (2018)	LPQ, RL and oBIF with OC-K-Means	94.5
He and Schomaker (2020)	CNN with word images and fragments	96.3
Nguyen et al. (2019)	CNN	90.12
He and Schomaker (2018)	Deep Adaptive CNN	69.5
Sulaiman et al. (2019)	CNN + LBP	80.81
Xing & Qiao (2016)	CNN	97.3
Proposed	**Multi-Input CNN Bi-LSTM**	**98.78**

Table 6. Performance comparison with state-of-the-art on CVL dataset.

Author	Feature extraction approach	Top-1(%)
Kleber et al. (2013)	CS-UMD	97.9
Kleber et al. (2013)	QUQA-B	92.9
Kleber et al. (2013)	TEBESSA-c	97.6
Kleber et al. (2013)	TSINGHUA	97.7
Fiel & Sablatnig (2013)	Gaussian Mixture Model and Fisher kernel	97.8
He and Schomaker (2018)	Deep Adaptive CNN	79.1
Sulaiman et al. (2019)	CNN + LBP	89.32
Proposed	**Multi-Input CNN Bi-LSTM**	**98.55**

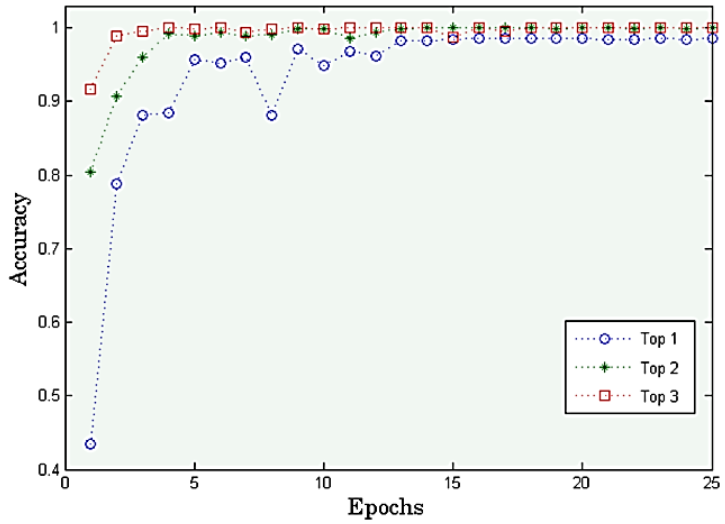

Figure 12. Top-1, Top-2, Top-3 performance of the proposed method with CVL dataset.

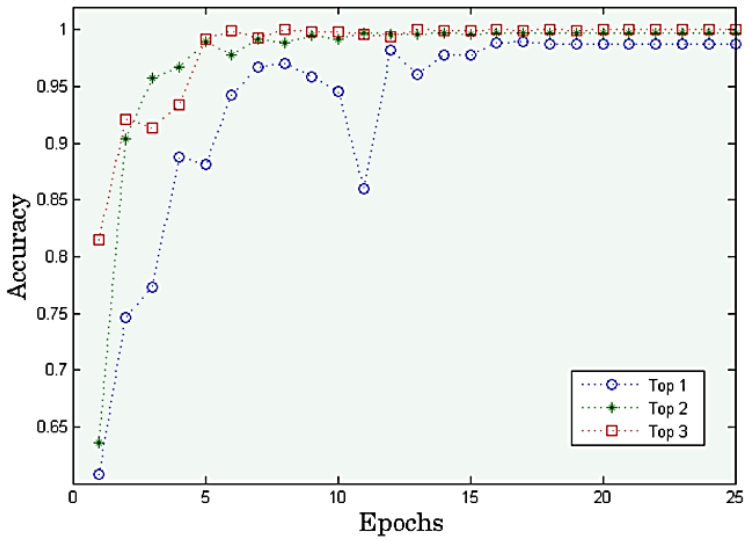

Figure 13. Top-1, Top-2, Top-3 performance of the proposed method with IAM dataset.

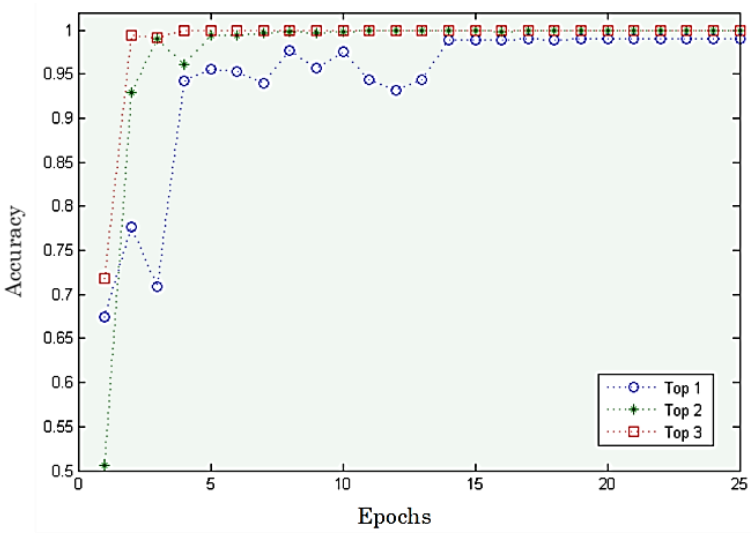

Figure 14. Top-1, Top-2, Top-3 performance of the proposed method with in-house dataset.

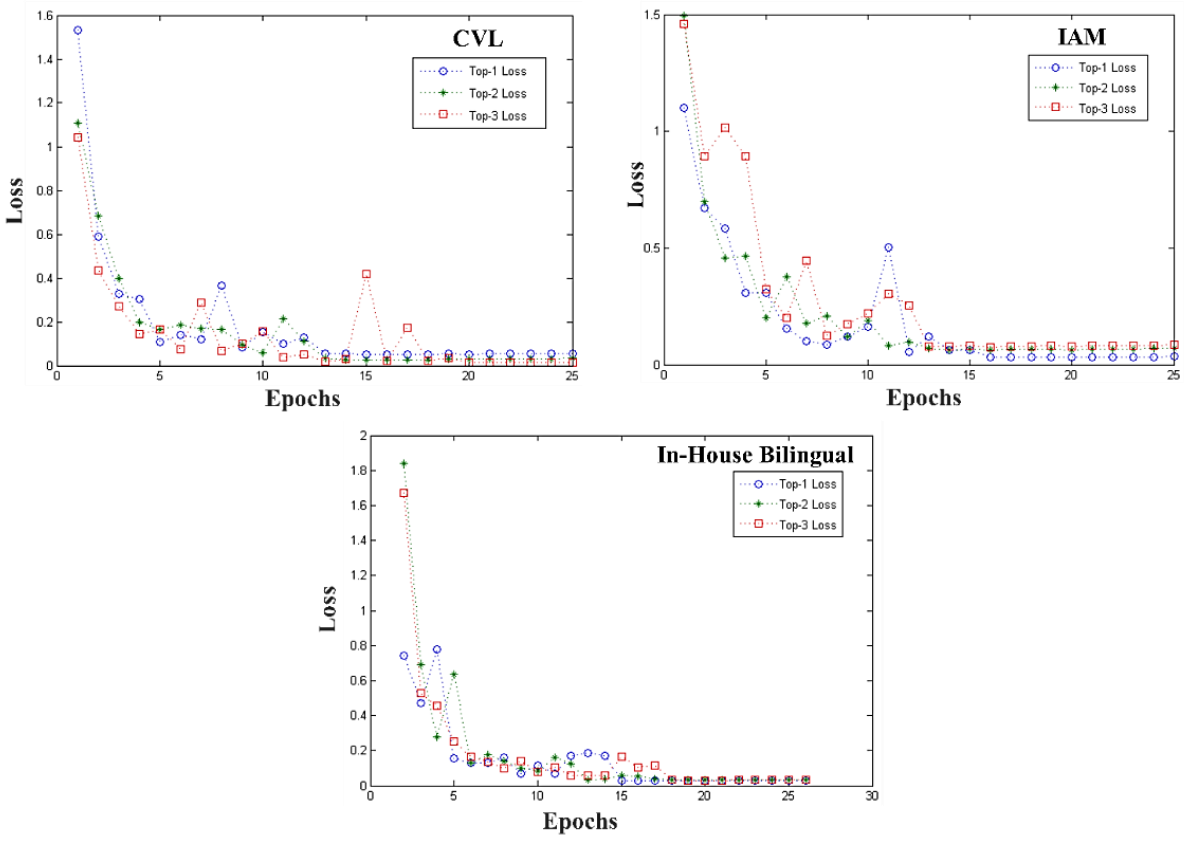

Figure 15. Loss visualization for all datasets.

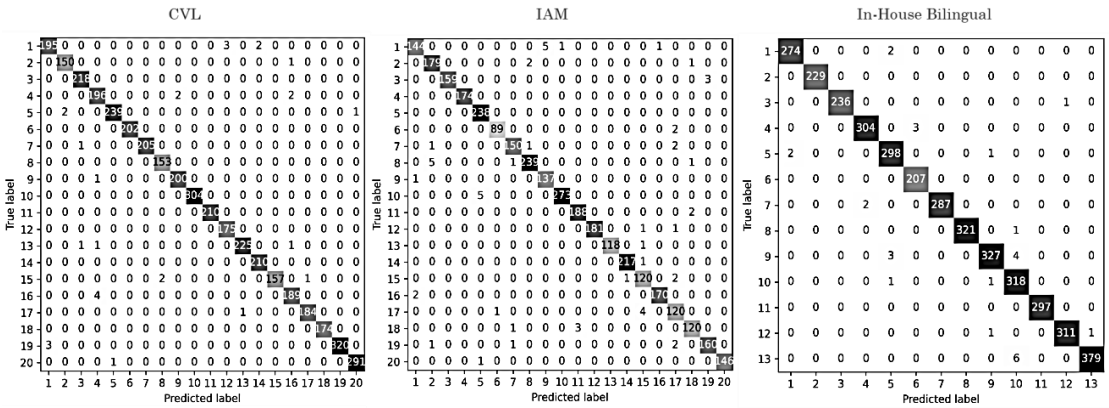

Figure 16. Confusion matrix for all datasets.

While the proposed model demonstrates strong performance, certain inherent challenges in offline writer recognition contribute to the observed errors:

1) Variability in handwriting styles:

- ***Personalized Handwriting Traits*:** Each individual has unique writing habits, such as irregular letter formations, inconsistent spacing, and variability in slant angles, which can confuse the model's feature extraction process.
- ***Writing Speed and Pressure Variations*:** Differences in writing pressure and speed lead to inconsistent stroke thickness, causing the model to misinterpret certain characters or features.
- ***Cursive Writing Challenges*:** For writers who use cursive styles, connected letters sometimes blend together, making segmentation and feature extraction more difficult.

2) Dataset-Related Noise:

- ***Artifacts in Scanned Images*:** The IAM and CVL datasets, as well as the in-house bilingual dataset, contain samples with noise like smudges, blurs, and irregularities caused by scanning artifacts or document wear.
- ***Multilingual Challenges in Bilingual Dataset*:** In the bilingual dataset, scripts from different languages (Hindi and English) occasionally overlap or blend, leading to confusion in distinguishing stylistic features unique to each script.

3) Model Limitations in Feature Extraction:

- ***HoG Features in Hybrid Model*:** While HoG features enhance the hybrid model's performance, they can sometimes emphasize irrelevant details in highly noisy or distorted samples, leading to misclassification.
- ***Difficulty Capturing Global Context*:** The multi-input CNN is designed to capture localized features effectively. However, in cases where global context is crucial (e.g., long sequences of text), errors may arise.
- ***Sensitivity to Stroke Variations*:** Subtle stroke differences (e.g., minor variations in loop size or tail length) between writers are sometimes missed by the CNN, leading to errors in distinguishing closely related handwriting styles.

While the current evaluation focuses on the IAM, CVL, and an in-house bilingual dataset (English and Hindi), we recognize the value of extending the evaluation to encompass additional scenarios. Below, we outline potential use cases that could enhance the robustness and comprehensiveness of our study:

1) *Arabic handwriting recognition*: Arabic handwriting introduces unique challenges with its cursive structure, positional character changes (initial, medial, and terminal forms), and high variability in stroke styles. Testing on datasets like KHATT or the Arabic Handwriting Dataset will demonstrate the model's adaptability to scripts that differ significantly from Latin-based languages.

2) *Indic handwriting recognition*: Expanding beyond Hindi to other Indic scripts such as Tamil, Bengali, or Kannada, which have unique orthographic structures with rounded characters and ligatures, would highlight the model's ability to generalize across languages with distinct script characteristics. This would further strengthen its application in multilingual regions.

3) *Forensic writer identification*: Forensic handwriting analysis often requires identifying or verifying writers from limited samples. Simulating such scenarios with small, diverse datasets (e.g., datasets mimicking forensic document collections) would demonstrate the model's utility in real-world applications.

4) *Unseen dataset validation*: Testing the model on a completely unseen dataset, such as RIMES (French handwritten documents) or FIREMAKER (German handwriting), would emphasize its generalization capability.

6. Conclusion

In this research, we proposed a novel multi-input hybrid CNN Bi-LSTM model for offline writer recognition, leveraging the strengths of convolutional neural networks (CNN) for feature extraction and bidirectional long short-term memory networks (Bi-LSTM) for sequence modeling. The system exhibited the highest top-1 accuracy of 98.55% with CVL, 98.78% with IAM, and 99% with in-house data, as reported by the results of the experiments. The model demonstrated superior accuracy compared to state-of-the-art methods, showcasing its efficacy in capturing both spatial and sequential features critical for robust writer identification. To enhance the performance and robustness of the model, we introduced a comprehensive data augmentation strategy. This involved constructing paragraph image patches from a combination of publicly available datasets (IAM and CVL) and an in-house developed bilingual dataset comprising English and Hindi scripts. The inclusion of bilingual scripts represents a significant contribution to the field, addressing the growing need for multilingual writer recognition systems in diverse linguistic contexts.

The results obtained from extensive experimentation indicate that our model not only achieves high accuracy across multiple datasets but also generalizes effectively across different writing styles and scripts. The hybrid architecture, combined with data augmentation, has proven to be a robust solution, paving the way for advancements in writer recognition, especially in multilingual scenarios. Although the proposed model demonstrates strong performance in offline writer recognition, it is with some potential limitations. One significant challenge is the computational cost associated with training and inference, as Bi-LSTM requires processing sequences in both forward and backward directions, which can increase memory and runtime requirements. Additionally, the presence of strikethrough handwritten text in datasets, which can introduce noise and ambiguity, making it difficult for the model to accurately extract meaningful features. Future work could focus on extending the dataset to include more languages and exploring further enhancements to the model architecture to tackle more complex and diverse handwriting datasets. Overall, this work represents a significant step forward in offline writer recognition, offering a powerful tool for applications in document forensics, authentication, and beyond.

Conflicts of Interest

The authors confirm that there is no conflict of interest to declare for this publication.

AI Disclosure
The author(s) declare that no assistance is taken from generative AI to write this article.

Acknowledgments
We sincerely appreciate the valuable feedback and constructive comments provided by the reviewers. Their insights have significantly contributed to improving the quality of our manuscript.

References

Abu-Ain, W., Abdullah, S.N.H.S., Bataineh, B., Abu-Ain, T., & Omar, K. (2013). Skeletonization algorithm for binary images. *Procedia Technology, 11*, 704-709. https://doi.org/10.1016/j.protcy.2013.12.248.

Adak, C., Chaudhuri, B.B., & Blumenstein, M. (2018). A study on idiosyncratic handwriting with impact on writer identification. In *2018 16th International Conference on Frontiers in Handwriting Recognition* (pp. 193-198). IEEE. Niagara Falls, New York. https://doi.org/10.1109/icfhr-2018.2018.00042.

Alom, M.Z., Taha, T.M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M.S., & Asari, V.K. (2018). The history began from AlexNet: a comprehensive survey on deep learning approaches. *Computer Vision and Pattern Recognition*. https://doi.org/10.48550/arxiv.1803.01164.

Bulacu, M., & Schomaker, L. (2007). Text-independent writer identification and verification using textural and allographic features. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(4), 701-717. https://doi.org/10.1109/tpami.2007.1009.

Chen, S., Wang, Y., Lin, C.T., Ding, W., & Cao, Z. (2019). Semi-supervised feature learning for improving writer identification. *Information Sciences, 482*, 156-170. https://doi.org/10.1016/j.ins.2019.01.024.

Christlein, V., Bernecker, D., Maier, A., & Angelopoulou, E. (2015). Offline writer identification using convolutional neural network activation features. In: Gall, J., Gehler, P., Leibe, B. (eds) *Pattern Recognition*. Springer International Publishing, Cham, pp. 540-552. https://doi.org/10.1007/978-3-319-24947-6_45.

Christlein, V., Gropp, M., Fiel, S., & Maier, A. (2017). Unsupervised feature learning for writer identification and writer retrieval. In *2017 14th IAPR International Conference on Document Analysis and Recognition* (pp. 991-997). IEEE. Kyoto, Japan. https://doi.org/10.1109/icdar.2017.165.

Christlein, V., Gropp, M., Fiel, S., & Maier, A. (2017). unsupervised feature learning for writer identification and writer retrieval. In *2017 14th IAPR International Conference on Document Analysis and Recognition* (pp. 991-997). IEEE. Kyoto, Japan. https://doi.org/10.1109/icdar.2017.165.

Elbarawy, Y.M., & Ghonaim, W.A. (2021). Hybridized convolution neural network and multiclass-SVM model for writer identification. *Engineering Letters, 29*(1), 1-10.

Fiel, S., & Sablatnig, R. (2013). Writer identification and writer retrieval using the fisher vector on visual vocabularies. In *2013 12th International Conference on Document Analysis and Recognition* (pp. 545-549). IEEE. Washington, DC, USA. https://doi.org/10.1109/icdar.2013.114.

Gumusbas, D., & Yildirim, T. (2019). Offline signature identification and verification using capsule network. In *2019 IEEE International Symposium on Innovations in Intelligent Systems and Applications* (pp. 1-5). IEEE. Sofia, Bulgaria. https://doi.org/10.1109/inista.2019.8778228.

Hadjadji, B., & Chibani, Y. (2018). Two combination stages of clustered one-class classifiers for writer identification from text fragments. *Pattern Recognition, 82*, 147-162. https://doi.org/10.1016/j.patcog.2018.05.001.

He, S., & Schomaker, L. (2018). Deep adaptive learning for writer identification based on single handwritten word images. *Pattern Recognition, 88*, 64-74. https://doi.org/10.1016/j.patcog.2018.11.003.

He, S., & Schomaker, L. (2020). FragNet: writer identification using deep fragment networks. *IEEE Transactions on Information Forensics and Security, 15*, 3013-3022. https://doi.org/10.1109/tifs.2020.2981236.

He, S., & Schomaker, L. (2021). GR-RNN: global-context residual recurrent neural networks for writer identification. *Pattern Recognition, 117*, 107975. https://doi.org/10.1016/j.patcog.2021.107975.

Jain, R., & Doermann, D. (2011). Offline writer identification using K-adjacent segments. In *2011 International Conference on Document Analysis and Recognition* (pp. 769-773). IEEE. Beijing, China. https://doi.org/10.1109/icdar.2011.159.

Jain, R., & Doermann, D. (2014). Combining Local Features for Offline Writer Identification. In *2014 14th International Conference on Frontiers in Handwriting Recognition* (pp. 583-588). IEEE. Hersonissos, Greece. https://doi.org/10.1109/icfhr.2014.103.

Kaur, R.P., Jindal, M.K., & Kumar, M. (2020). Text and graphics segmentation of newspapers printed in Gurmukhi script: a hybrid approach. *The Visual Computer, 37*(7), 1637-1659. https://doi.org/10.1007/s00371-020-01927-0.

Kaur, R.P., Kumar, M., & Jindal, M.K. (2023). Performance evaluation of different features and classifiers for Gurumukhi newspaper text recognition. *Journal of Ambient Intelligence and Humanized Computing, 14*(8), 10245-10261. https://doi.org/10.1007/s12652-021-03687-8.

Keglevic, M., Fiel, S., & Sablatnig, R. (2018). Learning features for writer retrieval and identification using triplet CNNs. In *2018 16th International Conference on Frontiers in Handwriting Recognition* (pp. 211-216). IEEE. Niagara Falls, New York, USA. .https://doi.org/10.1109/icfhr-2018.2018.00045.

Khalifa, E., Al-Maadeed, S., Tahir, M.A., Bouridane, A., & Jamshed, A. (2015). Off-line writer identification using an ensemble of grapheme codebook features. *Pattern Recognition Letters, 59*, 18-25. https://doi.org/10.1016/j.patrec.2015.03.004.

Kleber, F., Fiel, S., Diem, M., & Sablatnig, R. (2013). CVL-DataBase: An off-line database for writer retrieval, writer identification and word spotting. In *2013 12th International Conference on Document Analysis and Recognition* (pp. 560-564). IEEE. Washington, DC, USA. https://doi.org/10.1109/icdar.2013.117.

Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM, 60*(6), 84-90. https://doi.org/10.1145/3065386.

Kumar, M., Jindal, M.K., & Kumar, M. (2022). Distortion, rotation and scale invariant recognition of hollow Hindi characters. *Sadhana, 47*(2), 92. https://doi.org/10.1007/s12046-022-01847-w.

Kumar, V., & Sundaram, S. (2024). Utilization of information from CNN feature maps for offline word-level writer identification. *Expert Systems with Applications, 238*(Part A), 121709. https://doi.org/10.1016/j.eswa.2023.121709.

Li, B., & Lima, D. (2021). Facial expression recognition via ResNet-50. *International Journal of Cognitive Computing in Engineering, 2*, 57-64. https://doi.org/10.1016/j.ijcce.2021.02.002.

Maaz, S., & Issa, H. (2020). Using deep learning for Arabic writer identification. *International Journal of Computer Applications, 175*(25), 1-7. https://doi.org/10.5120/ijca2020920783.

Marti, U.V, & Bunke, H. (2002). The IAM-database: an English sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition, 5*(1), 39-46. https://doi.org/10.1007/s100320200071.

Nabi, S.T., Kumar, M., & Singh, P. (2023). DeepNet-WI: a deep-net model for offline Urdu writer identification. *Evolving Systems, 15*(3), 759-769. https://doi.org/10.1007/s12530-023-09504-1.

Nabi, S.T., Kumar, M., & Singh, P. (2024). DeepNet-WI: a deep-net model for offline Urdu writer identification. *Evolving Systems, 15*(3), 759-769. https://doi.org/10.1007/s12530-023-09504-1.

Nguyen, H.T., Nguyen, C.T., Ino, T., Indurkhya, B., & Nakagawa, M. (2019). Text-independent writer identification using convolutional neural network. *Pattern Recognition Letters, 121*, 104-112. https://doi.org/10.1016/j.patrec.2018.07.022.

Semma, A., Hannad, Y., Siddiqi, I., Djeddi, C., & Kettani, M.E.Y.E. (2021). Writer identification using deep learning with FAST keypoints and Harris corner detector. *Expert Systems with Applications, 184*, 115473. https://doi.org/10.1016/j.eswa.2021.115473.

Sulaiman, A., Omar, K., Nasrudin, M.F., & Arram, A. (2019). Length independent writer identification based on the fusion of deep and hand-crafted descriptors. *IEEE Access, 7*, 91772-91784. https://doi.org/10.1109/access.2019.2927286.

Sunny, M.A.I., Maswood, M.M.S., & Alharbi, A.G. (2020). Deep learning-based stock price prediction using LSTM and bi-directional LSTM Model. In *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference* (pp. 87-92). IEEE. Giza, Egypt. https://doi.org/10.1109/niles50944.2020.9257950.

Wang, M., Zheng, S., Li, X., & Qin, X. (2014). A new image denoising method based on Gaussian filter. In *2014 International Conference on Information Science, Electronics and Electrical Engineering* (pp. 163-167). IEEE. Sapporo, Japan. https://doi.org/10.1109/infoseee.2014.6948089.

Wu, X., Tang, Y., & Bu, W. (2014). Offline text-independent writer identification based on scale invariant feature transform. *IEEE Transactions on Information Forensics and Security, 9*(3), 526-536. https://doi.org/10.1109/tifs.2014.2301274.

Xing, L., & Qiao, Y. (2016). DeepWriter: a multi-stream deep cnn for text-independent writer identification. In *2016 15th International Conference on Frontiers in Handwriting Recognition* (pp. 584-589). IEEE. Shenzhen, China. https://doi.org/10.1109/icfhr.2016.0112.

Xu, X., Xu, S., Jin, L., & Song, E. (2011). Characteristic analysis of Otsu threshold and its applications. *Pattern Recognition Letters, 32*(7), 956-961. https://doi.org/10.1016/j.patrec.2011.01.021.