



An Unsupervised Writer Identification Based on Generating Clusterable Embeddings

M. F. Mridha¹, Zabir Mohammad², Muhammad Mohsin Kabir², Aklima Akter Lima²,
Sujoy Chandra Das², Md Rashedul Islam^{3,*} and Yutaka Watanobe⁴

¹Department of Computer Science and Engineering, American International University Bangladesh, Dhaka, 1229, Bangladesh

²Department of Computer Science & Engineering, Bangladesh University of Business & Technology, Dhaka, 1216, Bangladesh

³Department of Computer Science and Engineering, University of Asia Pacific, Dhaka, 1216, Bangladesh

⁴Department of Computer Science and Engineering, University of Aizu, Aizu-Wakamatsu, 965-8580, Japan

*Corresponding Author: Md Rashedul Islam. Email: rashed.cse@gmail.com

Received: 03 June 2022; Accepted: 15 December 2022

Abstract: The writer identification system identifies individuals based on their handwriting is a frequent topic in biometric authentication and verification systems. Due to its importance, numerous studies have been conducted in various languages. Researchers have established several learning methods for writer identification including supervised and unsupervised learning. However, supervised methods require a large amount of annotation data, which is impossible in most scenarios. On the other hand, unsupervised writer identification methods may be limited and dependent on feature extraction that cannot provide the proper objectives to the architecture and be misinterpreted. This paper introduces an unsupervised writer identification system that analyzes the data and recognizes the writer based on the inter-feature relations of the data to resolve the uncertainty of the features. A pairwise architecture-based Autoembedder was applied to generate clusterable embeddings for handwritten text images. Furthermore, the trained baseline architecture generates the embedding of the data image, and the K-means algorithm is used to distinguish the embedding of individual writers. The proposed model utilized the IAM dataset for the experiment as it is inconsistent with contributions from the authors but is easily accessible for writer identification tasks. In addition, traditional evaluation metrics are used in the proposed model. Finally, the proposed model is compared with a few unsupervised models, and it outperformed the state-of-the-art deep convolutional architectures in recognizing writers based on unlabeled data.

Keywords: Writer identification; pairwise architecture; clusterable embeddings; convolutional neural network



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

Handwriting is considered a productive and dependable segment of behavioral biometrics. However, it provides more specific information regarding the individual who assembles it. Furthermore, the interclass variance varies extensively in handwriting characteristics from one individual to another. Psychologists, graphologists, forensic experts, and historians have spent the last few decades researching handwriting analysis, which is an intriguing and complex area of pattern-recognition exploration.

Advancing technology in handwriting recognition has several applications. This study aims to develop new concepts and methodologies for handwriting analysis and construct dependable recognition algorithms to understand and incorporate various writing styles. Numerous systems have been designed to identify writers, including online and offline identification based on handwriting [1]. Identifying a writer based on a text that has already been uploaded, scanned, and translated into a computer as a sample image is known as offline writer identification. The method of determining the writer while writing is recorded with electronic devices such as a stylus pen or tablet, known as online writer identification. Because of variances in handwriting style, paper quality, and preprocessing processes, identifying a writer based on offline methods is more complex than identifying an online writer identification system, as offline methods work with images of handwritten text. Compared to offline data, working with online data is simpler.

The computer-based writer identification (WI) approaches have been produced since the mid-1960s. State-of-the-art writer verification and identification methods from the 1960s to the 1980s were discussed in [2]. Earlier investigations by De Bruyne et al. [3] considered signatures as individual signs, concentrating on using offline signatures for identification purposes. Yoshimura et al. [4,5] devised a text-dependent approach to identify writers employing handwritten text. However, considerable studies have been conducted only on handwritten text, with some limitations in gathering signature patterns and massive text databases since the early 2000s. Srihari et al. [6] confirmed that the distinctiveness of handwritten text is acceptable for writer identification, and machine learning (ML) algorithms teach this task to many writers.

Furthermore, WI can be categorized into two approaches: text-dependent and text-independent [7]. A writer must produce the exact text used for recognition using the text-dependent methodology. However, text-dependent writer identification can achieve higher accuracy. In contrast, the practical implementation of this approach is confined and specifies the requirement of writing with fixed text content. Therefore, text-independent writer identification is generally more general and efficient for real-world applications. Nonetheless, this system must capture the slight differences in distinct writers' handwriting styles and avoid the influence of significant contrasts in handwriting content. Therefore, text-independent writer identification is a complex task.

Nearly every applied algorithm is utilized on labeled datasets among the various writer identification systems. The supervised learning algorithm acquires approximately perfect accuracy on a performance scale. However, these supervised learning algorithms require a large amount of labeled data. A few unsupervised methods have been proposed, most of which depend on feature extraction methods such as scale-invariant feature transform (SIFT) [8]. Thus, we were eager to establish an unsupervised approach that did not require external feature extraction. This study proposes a text-independent offline writer identification method for detecting a writer's handwriting using unlabeled data. The proposed approach is unsupervised and utilizes genuine feature relationships of the data. This method requires handwritten sentence segments and ensures that a sentence contains only one user's handwriting. Correspondingly, this technique employs paired architectures to construct clusterable embeddings from handwriting [9]. This method makes the following contributions:

1. This paper introduces an unsupervised writer identification method capable of identifying unlabeled data for handwritten datasets using an openly accessible “IAM dataset.”
2. The paper operates a pairwise architecture that constructs clusterable embeddings based on feature similarities and dissimilarities of the handwritten image frame.
3. This study exploits a baseline architecture for training and evaluating data and compares it with other popular baseline architectures. The architecture validated that the proposed approach performed better than different unsupervised methodologies.

The remainder of this paper is organized as follows: Section 2 examines relevant work in the writer identification domain. Section 3 explains the technique and structure of the training phase, as well as the baseline structures of the proposed method, including challenges and modifications. Section 4 presents the process, experimental setup, evaluation approach, datasets, and performance analysis architecture. Section 5 concludes the paper and discusses the future endeavors and applications of the proposed method.

2 Related Work

Last few decades, much research have been done on hand gesture recognition and handwriting recognition to identify users and other HCI application [1,3,10,11]. Writer identification has become a prominent topic in recent decades, and various strategies have been proposed to address this issue. However, some methodologies employed for extracting writer-specific information from handwritten patterns were based on previous research. Statistics-based methods, such as Scale-Invariant Feature Transform (SIFT), RootSIFT, contour, hinge, path signature, and several handcrafted techniques have been used in previous studies [11]. Earlier methods relied on supervised learning using a single or more constructed feature extractor. Dargan et al. [12] developed a novel study to identify writers based on the Devanagari script’s pre-segmented characters. The experiment used a corpus containing five copies of each Devanagari script character authored by 100 different writers, totaling 24,500 samples. The study employed four feature extraction methodologies: zoning, diagonal, transition, and peak extent-based features and classification methods, including k-NN and linear Support Vector Machine (SVM) [12,13]. The identification accuracy was 91.53% when using a linear SVM classifier with zoning, transition, and peak-extent-based features. However, autoencoders and deep-learning methods can better classify overlapping characters and noisy data.

Recently, the objective has been to mitigate supervised learning methods, operate them on unsupervised datasets, and achieve better performance. Researchers have started their work on unsupervised writer identification techniques to construct an automated writer identification system and have extensively investigated Deep Learning (DL) architectures. Semma et al. [14] described a writer identification system that uses handwriting to extract important points and input the small patches surrounding these critical points. They used the FAST vital points and Harris corner detector to identify the points of interest in handwriting. A deep convolutional neural network (CNN) is trained using small patches placed around it to perform end-to-end classification and encode features from convolutional layers. In addition, an experimental study was conducted on three separate datasets: IAM, QUWI, and IFN/ENIT, with an identification rate of 99.8%. They identified the writer’s handwritten text using small patches, and their performance was almost perfect.

Several studies have focused on different convolutional neural network architectures with a combination of segmentation algorithms, innovative denoisers, customized CNNs, and handcrafted features. Convolutional Neural Network (CNN) is trained with a global feature aggregator that aggregates

locally retrieved features in various ways [15]. Experimenting with datasets such as Fire Marker and IAM with 900 writers, the identification rate of the proposed method was 91.81%. Kang et al. [16] proposed an unsupervised writer adaptation strategy for recognizing generic handwritten text trained with synthetic fonts for a fresh incoming writer. Five distinct datasets were used to test and validate the proposed method. It has a character error rate of 6.75 and a word error rate of 17.26 and provides a practical and generic strategy to deal with new document collections. The system has many dimensions that are unrestricted for improvement; thus, it can be used as a solution for generic data.

Deep convolutional neural networks (CNNs) [17] are denoised with handcrafted descriptor features by combining deep learning and shared computer vision approaches. They outperformed the state-of-the-art methods in identifying noisy manuscripts by over 10%. This system emphasizes various types of noise. He et al. [18] proposed a deep adaptive learning method for writer identification using multi-task learning. In addition, an additional task was added to the training process to enforce the emergence of reusable features. DeepWriter is a multi-stream CNN proposed by Xing et al. [19] to learn a deep, powerful representation for writer identification. To improve the performance of DeepWriter, they created a multi-stream structure, optimized it, and added data augmentation learning. Furthermore, He et al. [20] proposed the fragNet architecture using two pathways: The feature pyramid is utilized to extract feature maps, and the fragment route is trained to predict the writer's identity. In [21], authors proposed an end-to-end writer identification system that relies on an extended version of ResNet by conjugating deep residual networks and very efficient feature extraction, emphasizing feature engineering.

However, few unsupervised writer identification methods have been proposed. Christlein et al. [22] and Chammas et al. [23] used SIFT as a key point descriptor, extracted image patches from handwritten documents, and clustered the descriptors as surrogate classes. Furthermore, a deep CNN (such as ResNet) is trained to identify writers using the extracted image patches of SIFT and cluster information (CI) from SIFT descriptors. The results were encoded using a Vector of Locally Aggregated Descriptors (VLAD) [24] and applied to an Exemplar Support Vector Machine (E-SVM) at the end to compare the results. The method should increase the number of input patches from the critical point to multiple written sentences and employ the system for improvement. In terms of embeddings, Sun et al. [25] proposed a three-branch embedding network (TBE-Net) with complementary feature learning and part-aware ability. In [26], the authors presented an unsupervised writer identification method using autoencoders to generate embeddings. The author employed two types of autoencoders: a Convolutional autoencoder (CAE) and a Variational autoencoder (VAE). The author utilized SIFT key points to extract image patches and trained the autoencoders to encode and decode the input image patches. The feature vector is obtained by a forward pass of image patches over the trained encoder, and multi-VLAD (m-VLAD) was applied to classify the writers. Table 1 illustrates a comparison of state-of-the-art unsupervised writer identification.

Table 1: The table compares of state-of-the-art unsupervised writer identification methods

Authors	Feature extraction	Approach	Classifier	Accuracy
[16]	SIFT/R-SIFT	ResNet20	VLAD + Example SVM	88.3
[22]	SIFT	CNN + VLAD	Cosine distance + E-SVM	91.6
[26]	SIFT	Autoencoders	m-VLAD encoding	87.1

Current state-of-the-art writer identification techniques utilize clustering to construct unsupervised attributes of historical documents [22]. However, they used handcrafted features to converge with SVMs. Furthermore, the effectiveness relies on previously learned characteristics; thus, performance may suffer owing to data inefficiencies and domain diversity in the training and testing data. Consequently, we present a system that is independent of the CNN architecture domain adaptation to learn unsupervised features using an embedding technique. A paired architecture generates pair-related data flows belonging to classes from a chosen dataset for the task. The proposed method utilizes the automated identification of writers by employing an accessible neural network.

3 Methodology

The proposed unsupervised writer identification method uses a deep CNN (baseline architecture) as a shared network of a Siamese network-based AutoEmbedder [27] architecture to generate embeddings based on hypothetical constraints. This section describes the overall methodology of the proposed unsupervised writer identification system. Section 3.1 demonstrates the data preprocessing steps and segmentation. Section 3.2 represents pairwise constraints where inputs are differentiated and separated in the feature space. A description of the data augmentation processes and tabulation of the augmentation pipeline are presented in Section 3.3. Finally, Sections 3.4 and 3.5 illustrate the AutoEmbedder architecture and the methodology's overall architecture.

3.1 Data Preprocessing

Preprocessing is a data cleansing stage in which irrelevant data are removed. In this phase, image processing techniques, such as binarization, normalization, and noise removal [28] are applied to handwritten images. In addition, the handwritten dataset was considered for sentence segmentation. The IAM dataset provides a sentence segmentation schema at the line segmentation level. However, background noise issues were observed in the images from the dataset. First, we applied a supplementary OpenCV-based Python script [29] to eliminate unwanted data such as noise removal, background elimination, and others. Subsequently, we resized the line-segmented images to a height of 112 pixels while maintaining the aspect ratio. Furthermore, we segmented the line images into smaller image frames using a non-overlapping sliding window approach and assigned a pseudo label depending on the sentence segments. Fig. 1 illustrates the sentence segmentation schema from the documents and image frame segments of sentences from the IAM dataset. Finally, the dataset is scaled to [0, 1].

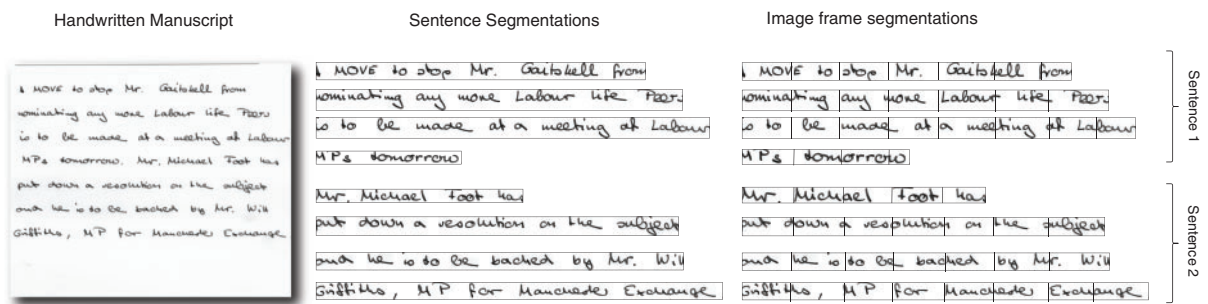


Figure 1: The figure represents the sentence segmentation schema of the IAM dataset and image frame segments from sentence segments

3.2 Pairwise Constraints

The proposed method solves unsupervised writer identification problems based on specific constraints. Generally, when an author starts writing, he or she tries to complete a statement such as a sentence. Therefore, smaller text blocks (in our study, smaller image frames) of the sentence were written by the same writer. Based on this intuition, we constructed pairwise constraints between two text images. If the image frames belong to the same sentence segment, they belong to the same class; otherwise, they belong to different classes.

In the context of the current problem, the ground truth of the writer's label for each image frame is unknown. We considered each sentence segment to belong to a different individual, resulting in several distinct pseudo-labels equaling $|S|$. We constructed pairwise constraints from sentence segments and used them to train DL frameworks to generate embeddings. A pairwise constraint represents and differentiates the relationship between two inputs. Consider two random image frames x_i and x_j , which have two possibilities: (a) the image frames belong to the same sentence segment X_k , or (b) they are related to a distinct sentence segment. Furthermore, C is a set of clusters, c_i is a specific cluster of smaller image frames of a sentence segment, and X_k is a specific sentence segment. The pairwise criterion can be mathematically expressed as,

$$\forall x_i \in X_k \text{ and } \forall x_j \in X_k: x_i, x_j \in c_k \quad (1)$$

$$\forall x_i \in X_k \text{ and } \forall x_j \notin X_k: x_i, x_j \notin c_k$$

The cluster restrictions defined in Eq. (1) is used to train the DL framework. To correctly introduce inter-cluster and intra-cluster relationships into a DL framework, we developed a ground regression function based on the paired criterion stated in Eq. (1). This function is expressed as follows:

$$P_c(x_i, x_j) = \begin{cases} 0, & \text{if } x_i, x_j \in c_p \\ \alpha, & \text{if } x_i \in c_p \text{ and } x_j \in c_q \end{cases} \quad (2)$$

Generally, $P_c(.,.)$ returns the distance constraints each pair for embeddings (generated from image frames) holds. If two embedding pairs belong to the same cluster, the function implies that they must be within a certain distance. However, embedding pairings from distinct clusters may be separated by more than α , as determined using AutoEmbedder architecture.

3.3 Data Augmentation

Data augmentation is a systematic strategy for boosting the number and variety of labeled training sets by utilizing input changes such as flipping, rotating, scaling, and cropping [30]. In the proposed pipeline, we used a non-generative online augmentation schema. Using other translation block probabilities, the augmentation schema randomly selected half of the batch data and applied different translations to the input image. The translation probabilities were normalized, and translation was applied based on probability normalization within a block. An augmentation process was used to resist overfitting in the DL framework. In addition, the proposed model can generate faulty assumptions about writers, as we consider each document to belong to a different individual. However, an author can contribute multiple documents indicating that our assumption of pairwise constraints can be faulty or incorrect in some cases. Data augmentation efficiently tricks DL networks and attempts to eliminate faulty assumptions. Further, Table 2 shows the translation probabilities and blocking probability.

Table 2: The table represents a detailed tabulation of the data augmentation pipeline along with their translation and block probability

Translations	Translation probability	Block probability
Random Crop	0.3	0.5
Flip	0.3	
Crop and Pad	0.3	
Gaussian Blue	0.5	0.5
Downscale	0.5	
Random Brightness & Contrast	0.5	0.5
Gaussian Noise	0.5	

3.4 AutoEmbedder Architecture

To re-cluster the data, we deployed a Siamese network constraint-based AutoEmbedder framework as a DL architecture [27]. However, further changes to the network's overall training process are required to improve the learning progress. The AutoEmbedder architecture was trained using the paired restrictions defined by the function $P_c(\cdot)$ in Eq. (2). The architecture adheres to the Siamese network constraints, which can be summarized as follows:

$$S(x_i, x_j) = \text{ReLU}(\|E_\phi(x_i) - E_\phi(x_j)\|, \alpha) \quad (3)$$

In Eq. (3), $S(x_i, x_j)$ denotes a two-input Siamese network function. $E_\phi(x_i)$ in the framework transfers higher-dimensional information to lower-dimensional clusterable embeddings using a shared DNN network (baseline architecture). Euclidean distance was used to calculate the distance between the embedding pairs. Furthermore, the embedding distance is passed via a threshold rectified linear unit (ReLU) activation function. The $\text{ReLU}(\cdot, \cdot)$ function used in Eq. (3) is a threshold ReLU function, such that,

$$\text{ReLU}(x, \alpha) = \begin{cases} 0, & \text{if } 0 \leq x < \alpha \\ \alpha, & \text{if } x \geq \alpha \end{cases} \quad (4)$$

where α (a hyperparameter) denotes the cluster margin between the embeddings. The Autoembedder design always provides outputs in the range $[0, \alpha]$ owing to the threshold.

3.5 Overall Proposed Strategy

The proposed unsupervised writer identification method was based on the inter-structure relationship of the data. First, to establish the structural relationship between the data, every sentence segment of a handwritten dataset is considered a distinct writer.

Furthermore, smaller image patch segmentation (such as a small block of text) was applied to every sentence segment of the handwritten dataset and a pseudo label is assigned according to segments. Furthermore, the image patches were placed into a cluster network to construct pairwise constraints. The pairwise constraint defines the input pair as belonging to the same class if they belong to the same sentence segment; otherwise, they belong to a different class. The pairwise architecture generates clusterable embeddings of the image frames based on hypothetical constraints. However, the Autoembedder-based pairwise architecture generates embeddings based on feature similarities rather

than hypothetical assumptions [31]. The baseline architecture was extracted from the DL network, and the previously generated smaller image patch was passed through the baseline architecture to generate embedding.

Correspondingly, the generated embeddings of the image frames were clustered using the K-means algorithm based on the number of writers. The K-means algorithm generates cluster or class indices of image frames. The clustering algorithm aggregates image frames based on feature similarities and is regarded as an individual's handwriting to evaluate with metrics, such as accuracy, recall, precision, and F1 score. Fig. 2 illustrates the workflow of the proposed unsupervised writer identification training and assessment procedure.

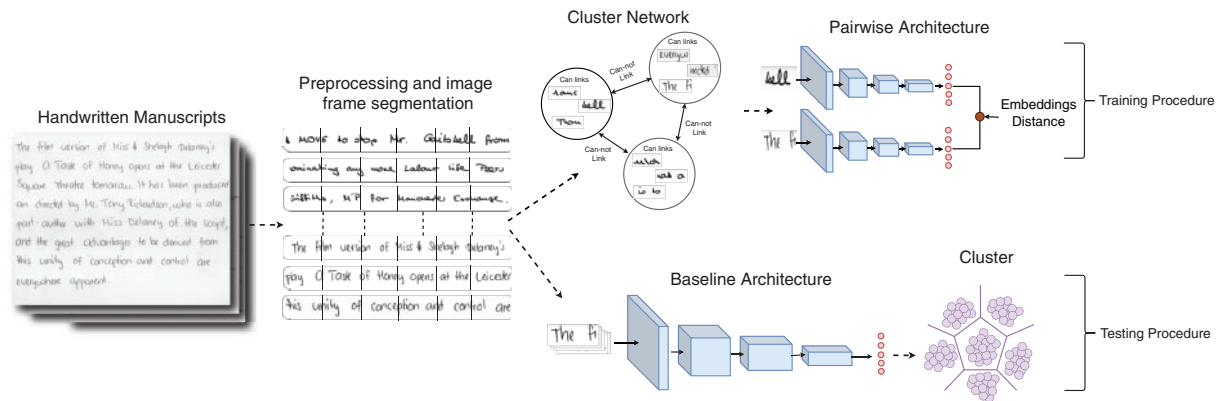


Figure 2: The figure illustrates the overall workflow of the proposed writer identification system

4 Evaluation

In this section, the performance of the proposed architecture is compared with that of state-of-the-art systems. Then, the datasets and training techniques are examined in the following sections, along with a summary of the experimental results.

4.1 Dataset

The proposed model was applied to the IAM [32] dataset. There are 13,353 photos of handwritten text lines in the IAM database, authored by 657 writers. The writings transcribed by these writers were from the British English Lancaster-Oslo/Bergen corpus. It is considered part of the modern collection and comprises contributions from 657 writers, totaling 1,539 handwritten pages with 115,320 words [33]. The database was labeled at the sentence, line, and word levels. However, the dataset was inconsistent because there was a lack of proper distribution between writers. Some writers have contributed a healthy number of writing pages, while others have contributed less. Of the 657 writers, 301 contributed more than two pages of handwriting. Therefore, we follow a recursive method to take the pages of writers with at least two writing pages. In addition, writers who contributed more than two pages are sliced into their first two pages. Therefore, it is clear that this study focuses on data subsets rather than the entire dataset.

4.2 Evaluation Metric

The proposed method identifies writers defined by clusters in the feature space based on generating clusterable embeddings. To evaluate the proposed model, a K-means algorithm was used. However,

a contingency matrix must be created to analyze the clusters found by the algorithm. The process requires comparing the result of a clustering algorithm $C = \{C_1, C_2, \dots, C_n\}$ to a potentially different partition of data $P = \{P_1, P_2, \dots, P_n\}$, which might be considered as ground-truth clusters.

This contingency matrix includes the following terms such as true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). TP indicates the number of data pairs discovered in the same cluster in C and P . Additionally, FP specifies the number of data pairs found in the same cluster C , and the number of data pairs found in separate clusters P . FN acts as the opposite of FP, where the number of data pairs that were discovered in distinct clusters in C but in the same cluster in P . Finally, TN identifies the number of data pairs found in different clusters in C and P . Four well-known and conventional accuracy measures are utilized to assess the correctness of the proposed method.

Clustering accuracy is a statistic used to evaluate it. Accuracy is one of the most frequently used metrics in biometric identification systems to assess how well a system correctly distinguishes an individual from a group of objects. The fraction of accurate forecasts of overall predictions is quantified as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

The precision metric is used to calculate the relevance of a positive prediction, which is calculated as,

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

The recall metric is used to compute positive predictions over positive ground predictions, which are computed as,

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

Finally, the F1-score is used to indicate the projected result's balance between precision and recall, which is calculated as follows.

$$F1_{score} = 2 \times \frac{precision \times recall}{precision + recall} \quad (8)$$

Here, the evaluation metrics, including accuracy, precision, recall, and F1-score produce results in the $[0, 1]$ range. The greater the value of these indices, the better the match between the cluster and ground truth.

4.3 Experimental Setup

Python was used to collect, gather, process, experiment, and evaluate data. Keras is used to implement the deep-learning methods. Individual mathematical calculations and image processing were performed using Numpy and OpenCV. In addition, a quick and flexible open-source image augmentation library with many different image transformation operations is presented, which is an easy-to-use wrapper over existing augmentation libraries called Albumentations. The pairwise architecture was trained using the mean square error as a loss function with a batch size of 64. To train the DL framework, we used the Adam optimizer with a learning rate of 0.0005.

4.4 Result Analysis

The result analysis is divided into two subsections representing the various architectures and unsupervised methodology comparisons. Firstly, we compare our proposed approach with diverse computer vision-based architecture on the “IAM” dataset. In addition, we include existing unsupervised writer identification methods that operate using handwritten text images.

Fig. 3 illustrates the F1-score on the validation dataset while training different deep CNN architectures. The Xception architecture performed better on the validation set according to the graphs. Theoretical approaches suggest that increasing the depth of a deep learning architecture should enhance classification performance. However, increasing the depth of the architecture often results in vanishing-gradient problems. The vanishing gradient problem causes the VGG architectures to perform poorly within the framework.

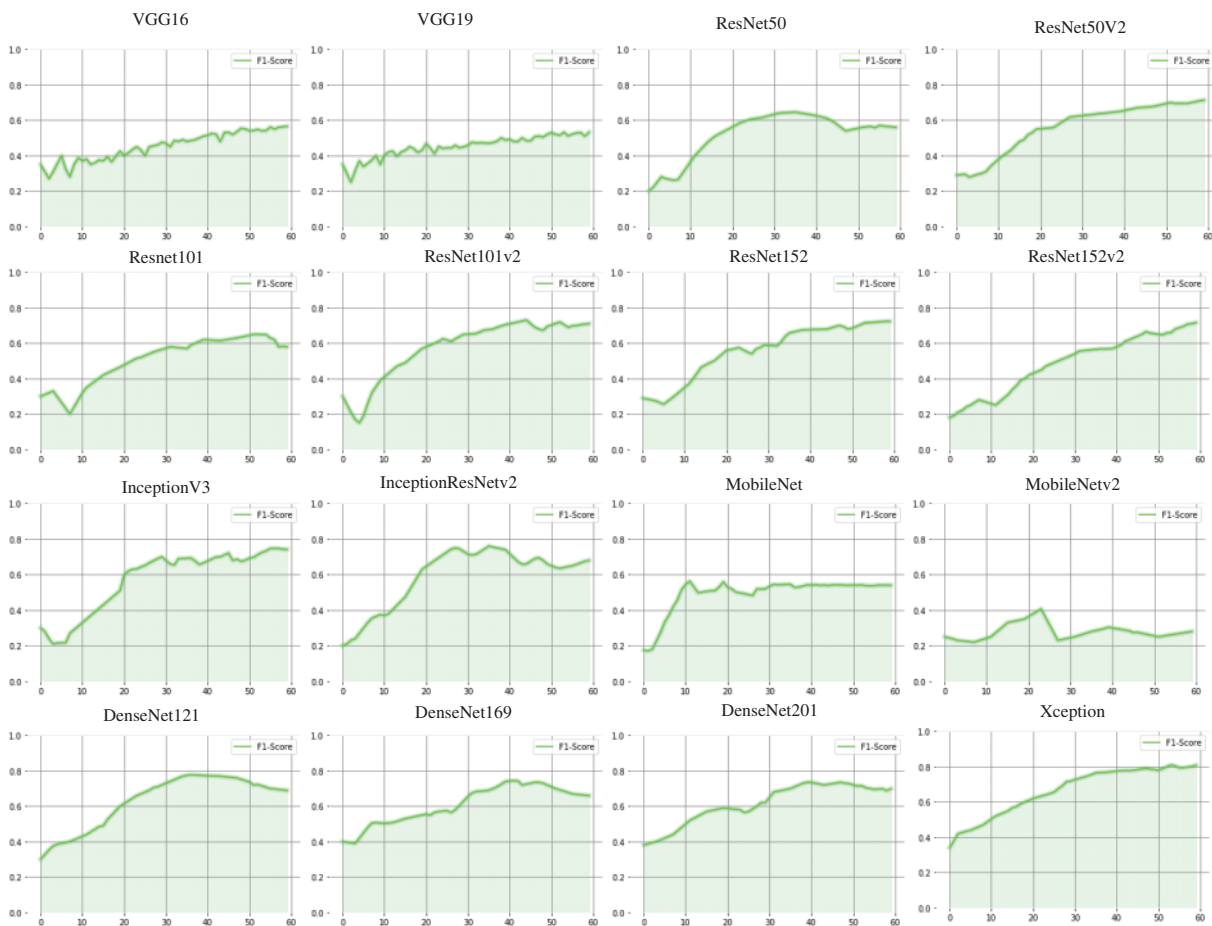


Figure 3: F1 score illustrations on the validation dataset. The epochs are represented by the horizontal axis, and the metrics score is illustrated by the vertical axis. As the metrics calculation requires quadratic time complexity, we saved the metrics after every five epochs

Nonetheless, a skipped connection may solve the vanishing gradient problem. ResNet uses a residual connection to resolve the vanishing gradient problem. However, they may not achieve good outcomes owing to overfitting. The inceptionResNet architecture is appropriately integrated with the

residuals and inception blocks, producing better results, but not adequately. DenseNet is based on the idea that convolutional networks perform better when the linkages between the input and output layers are shorter.

This strategy is frequently used to help architectures avoid overfitting and identify appropriate neural activation combinations for various input patterns. DenseNets has fewer parameters than Xception. However, the notion of a shorter link between the input and output does not improve performance. MobileNet designs have the fewest parameters of all the models that have been implemented, but they cannot identify writers at a satisfactory rate. MobileNets' low F1 score may imply that the problem is not related to training parameters but the need for an appropriate network architecture for writer identification.

In contrast, Xception architecture combines VGG and inception architectural methodologies. The Xception architecture follows the strategy of performing pointwise convolution before depth-wise convolutions. In addition, Xception employs a skip connection method. However, similar to the DenseNet architecture, this prevents extensive parallel connections. Xception architecture supports a single pair of parallel flows and performs spatial convolutions rather than general convolutions. Spatial convolutions that work on numerous filters simultaneously tend to better understand texture characteristics, with excellent convergence speed, minor accuracy improvement, and fewer trainable parameters than classic convolutions.

Moreover, separable convolution provides better nonlinearity recognition capability than any other architecture. Hence, apart from Xception, any other baseline fails to learn stroke patterns, which is considered an essential handwriting characteristic of individuals. Table 3 illustrates the comparison of DCNN architectures reported with accuracy, precision, recall, and F1-Score of all architectures calculated on the training, validation, and test datasets.

Table 3: The table compares various CNN architectures on the IAM dataset. Furthermore, the parameter column presents the number of parameters used in the architectures. The highest score is marked in bold

Architecture	Parameters	Train				Validation				Test			
		Precision	Accuracy	Recall	F1-score	Precision	Accuracy	Recall	F1-score	Precision	Accuracy	Recall	F1-score
VGG16	138	0.72	0.65	0.59	0.60	0.70	0.62	0.57	0.59	0.71	0.62	0.56	0.58
VGG19	143	0.75	0.62	0.56	0.59	0.71	0.60	0.52	0.57	0.69	0.61	0.53	0.55
ResNet50	25	0.79	0.67	0.59	0.63	0.77	0.65	0.59	0.62	0.72	0.64	0.58	0.60
ResNet50V2	25	0.78	0.72	0.60	0.68	0.80	0.69	0.62	0.67	0.79	0.67	0.58	0.65
ResNet101	44	0.76	0.68	0.56	0.62	0.74	0.65	0.57	0.63	0.72	0.63	0.54	0.60
ResNet101V2	44	0.85	0.74	0.71	0.77	0.86	0.77	0.70	0.75	0.82	0.74	0.67	0.71
ResNet152	60	0.71	0.75	0.62	0.74	0.78	0.74	0.68	0.72	0.77	0.73	0.69	0.74
ResNet152V2	60	0.87	0.82	0.70	0.73	0.84	0.80	0.71	0.77	0.82	0.78	0.70	0.76
DenseNet121	8	0.86	0.83	0.78	0.81	0.79	0.82	0.76	0.80	0.80	0.84	0.77	0.79
DenseNet169	14	0.87	0.78	0.68	0.79	0.89	0.81	0.72	0.78	0.87	0.78	0.69	0.76
DenseNet201	20	0.87	0.79	0.72	0.76	0.85	0.79	0.70	0.75	0.83	0.81	0.70	0.76
InceptionV3	23	0.78	0.81	0.74	0.79	0.76	0.80	0.71	0.77	0.75	0.81	0.73	0.78
Inception ResNetV2	55	0.88	0.82	0.74	0.80	0.87	0.80	0.73	0.78	0.85	0.78	0.71	0.76
MobileNet	4	0.74	0.67	0.55	0.63	0.72	0.64	0.52	0.59	0.69	0.64	0.52	0.60
MobileNetV2	3	0.55	0.49	0.37	0.45	0.53	0.45	0.34	0.41	0.50	0.42	0.31	0.42
Xception	22	0.93	0.84	0.79	0.82	0.91	0.86	0.77	0.83	0.90	0.87	0.75	0.84

Furthermore, we compare the proposed method with a state-of-the-art unsupervised writer identification method. Christlein et al. [22] and Chammas et al. [23] used SIFT or Restricted-SIFT (R-SIFT) as key point descriptors of image patches of handwritten documents and clustered the descriptors as surrogate classes. Furthermore, a deep Convolutional Neural Network (such as ResNet) was trained to identify writers using the extracted image patches of SIFT and cluster information (CI) from SIFT descriptors. The results were then encoded using a multi-vector of Locally Aggregated Descriptors (VLAD) and applied to an Exemplar Support Vector Machine (E-SVM) for comparison. SIFT is a well-known feature-extraction algorithm. However, it is a statistical approach for finding both minima and maxima in the Difference-of-Gaussian (DoG) scale space (known as a descriptor). A CNN was trained to identify writers based on clustering the descriptors considering each cluster as an individual writer. However, the architecture performs less efficiently because its accuracy depends on SIFT or R-SIFT, because deep neural networks can be trained on a faulty class. However, in our proposed unsupervised writer identification method, writers are identified using clusterable embeddings. The embedding generation is based on the actual feature relationships of an individual's handwriting. Therefore, the process does not require any feature extraction method before training a neural network, and embeddings are generated on the actual feature relation of the data. Thus, it produced significantly more efficient and adequate performance on the IAM dataset. In [26], the authors proposed an unsupervised writer identification method using autoencoders. They used two types of autoencoders: Convolutional autoencoder (CAE) and Variational autoencoder (VAE). SIFT key points were used to extract image patches and train the autoencoders to encode and decode the input image patches. The feature vector is obtained by a forward pass of image patches over the trained encoder, and multi-VLAD (m-VLAD) was applied to classify the writers. However, the autoencoder attempts to reconstruct the input image in the output layer. The architecture is trained using the reconstruction costs of the input and output. Therefore, an autoencoder learns to capture as much information as possible rather than as much relevant information as possible. Consequently, the autoencoder cannot determine the relevant information. Thus, the architecture fails to learn relevant information from handwritten text. However, the Autoembedder architecture generates clusterable embeddings based on feature similarities in a hyperspace. Thus, the architecture learns the relevant information from the data. The reason lies in Autoembedder's training strategy as the L2-loss learns aggregately from a batch of data. Therefore, the framework can obtain feature similarities because it is not precisely supervised using L2-loss. As a result, the architecture can re-cluster data in a hyperspace depending on feature similarities. Table 4 illustrates a comparison of state-of-the-art unsupervised writer identification.

Table 4: The table illustrates a performance comparison of existing unsupervised writer identification on the IAM dataset

Method	Accuracy	Precision	Recall	F1-score
CI-S (Gray/SIFT) [22]	0.81	0.87	0.75	0.78
CI-S + E-SVM-FE [22]	0.83	0.89	0.77	0.79
CIS + Resnet20 [16]	0.85	0.88	0.69	0.72
CAE m-VLAD [26]	0.69	0.68	0.61	0.65
VAE m-VLAD [26]	0.71	0.70	0.68	0.69
Proposed method	0.86	0.90	0.79	0.82

5 Conclusion

This study provides an unsupervised writer identification system for the IAM dataset that explicitly recognizes individuals. We offer paired architecture to categorize writers based on their handwriting. Pairwise architecture distinguishes writers by generating embeddings based on feature similarities instead of hypothetical assumption. In addition, this study includes a baseline architecture extracted from the DL network, which further assists in image frame clustering using K-means. This architecture assessed and improved the model's accuracy by focusing on the IAM dataset. The existing unsupervised writer identification methods depend on feature extraction, such as SIFT and R-SIFT. However, we eliminated the feature extraction process and used the data's feature relationship to identify the writers. The proposed method significantly outperforms the state-of-the-art unsupervised writer identification method. The entire experimental contribution and comparison would encourage researchers to examine the range of performances and variations in handwriting. However, the proposed approach depends on sentence segmentation. Although the IAM dataset provides a sentence segmentation schema, however that is not applicable in most cases. Therefore, an appropriate sentence segmentation procedure from handwritten documents can make the proposed approach more robust. Furthermore, this research contribution will facilitate researchers to investigate the range of standards, handwriting styles, and recognition approaches for language-dependent and language-independent writer identification.

Acknowledgement: We would like to thank the Bangladesh University of Business & Technology (BUBT), University of Asia Pacific (UAP), and University of Aizu (UoA) for supporting this research. Lots of gratitude and appreciation to the Advanced Machine Learning lab, BUBT; Computer Vision & Pattern Recognition Lab, UAP; Database System Lab, UoA; for research facilities and publication guidelines.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest regarding this present study.

References

- [1] S. Dargan, M. Kumar, A. Garg and K. Thakur, "Writer identification system for pre-segmented offline handwritten Devanagari characters using k-NN and SVM," *Soft Computing*, vol. 24, no. 13, pp. 10111–10122, 2020.
- [2] R. Plamondon and G. Lorette, "Automatic signature verification and writer identification-The state of the art," *Pattern Recognition*, vol. 22, no. 2, pp. 107–131, 1989.
- [3] P. De Bruyne and R. Forte, "Signature verification with elastic image matching," in *Proc. of Int. Carnahan Conf. on Security Technology*, Gothenburg, Sweden, pp. 113–118, 1986.
- [4] I. Yoshimora and M. Yoshimora, "Writer identification based on the arc pattern transformation," in *Proc. of 9th Int. Conf. on Pattern Recognition*, Rome, Italy, IEEE Computer Society, pp. 35–37, 1988.
- [5] M. Yoshimura, F. Kimura and I. Yoshimura, "Experimental comparison of two types of methods of writer identification," *IEICE Transactions (1976–1990)*, vol. 65, no. 6, pp. 345–352, 1982.
- [6] S. Srihari, S. H. Cha, H. Arora and S. Lee, "Individuality of handwriting: A validation study," in *Proc. of Sixth Int. Conf. on Document Analysis and Recognition*, Seattle, WA, USA, pp. 106–109, 2001.
- [7] S. K. Chan, Y. H. Tay and C. Viard-Gaudin, "Online text independent writer identification using character prototypes distribution," in *Proc. of 2007 6th Int. Conf. on Information, Communications & Signal Processing*, Singapore, pp. 1–5, 2007.

- [8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [9] G. Koch, R. Zemel and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. of ICML Deep Learning Workshop*, France, vol. 2, 2015.
- [10] F. A. Farid, N. Hashim, J. Abdullah, M. R. Bhuiyan, W. N. S. M. Isa *et al.*, "A structured and methodological review on vision-based hand gesture recognition system," *Journal of Imaging*, vol. 8, no. 153, pp. 1–19, 2022.
- [11] S. Chen, Y. Wang, C. T. Lin, W. Ding and Z. Cao, "Semi-supervised feature learning for improving writer identification," *Information Sciences*, vol. 482, no. 5, pp. 156–170, 2019.
- [12] S. Dargan, M. Kumar, A. Garg and K. Thakur, "Writer identification system for pre-segmented offline handwritten Devanagari characters using k-NN and SVM," *Soft Computing*, vol. 24, no. 13, pp. 10111–10122, 2020.
- [13] F. A. Farid, N. Hashim and J. Abdullah, "Vision-based hand gesture recognition from RGB video data using SVM," in *Proc. of Int. Workshop on Advanced Image Technology (IWAIT)*, Singapore, vol. 11049, pp. 265–268, 2019.
- [14] A. Semma, Y. Hannad, I. Siddiqi, C. Djeddi and M. E. Y. El Kettani, "Writer Identification using deep learning with FAST keypoints and harris corner detector," *Expert Systems with Applications*, vol. 184, pp. 115473, 2021.
- [15] H. T. Nguyen, C. T. Nguyen, T. Ino, B. Indurkha and M. Nakagawa, "Text-independent writer identification using convolutional neural network," *Pattern Recognition Letters*, vol. 121, no. 6, pp. 104–112, 2019.
- [16] L. Kang, M. Rusinol, A. Fornes, P. Riba and M. Villegas, "Unsupervised writer adaptation for synthetic-to-real handwritten word recognition," in *Proc. of the IEEE/CVF Winter Conf. on Applications of Computer Vision*, Snowmass Village, Colorado, pp. 3502–3511, 2020.
- [17] K. Ni, C. Patrick and H. Bradley, "Writer identification in noisy handwritten documents," in *Proc. of 2017 IEEE Winter Conf. on Applications of Computer Vision (WACV)*, Santa Rosa, CA, USA, pp. 1177–1186, 2017.
- [18] S. He and L. Schomaker, "Deep adaptive learning for writer identification based on single handwritten word images," *Pattern Recognition*, vol. 88, no. 4, pp. 64–74, 2019.
- [19] L. Xing and Y. Qiao, "Deepwriter: A multi-stream deep CNN for text-independent writer identification," in *Proc. of 2016 15th Int. Conf. on Frontiers in Handwriting Recognition (ICFHR)*, Shenzhen, China, pp. 584–589, 2016.
- [20] S. He and L. Schomaker, "Fragnet: Writer identification using deep fragment networks," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3013–3022, 2020.
- [21] M. Javidi and M. Jampour, "A deep learning framework for text-independent writer identification," *Engineering Applications of Artificial Intelligence*, vol. 95, no. 1, pp. 103912, 2020.
- [22] V. Christlein, M. Gropp, S. Fiel and A. Maier, "Unsupervised feature learning for writer identification and writer retrieval," in *Proc. of 2017 14th IAPR Int. Conf. on Document Analysis and Recognition (ICDAR)*, Kyoto, Japan, pp. 991–997, 2017.
- [23] M. Chammas, A. Makhoul and J. Demerjian, "Writer identification for historical handwritten documents using a single feature extraction method," in *Proc. of 2020 19th IEEE Int. Conf. on Machine Learning and Applications (ICMLA)*, Miami, FL, USA, pp. 1–6, 2020.
- [24] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez *et al.*, "Aggregating local image descriptors into compact codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1704–1716, 2012.
- [25] W. Sun, G. Dai, X. Zhang, X. He and X. Chen, "TBE-Net: A three-branch embedding network with part-aware ability and feature complementary learning for vehicle re-identification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14557–14569, 2022.

- [26] E. P. Arranz, “Unsupervised feature learning for writer identification,” Master’s Thesis, Universitat Politècnica de Catalunya, Barcelona, Catalonia, Spain, 2018.
- [27] A. Q. Ohi, M. F. Mridha, F. B. Safir, M. A. Hamid and M. M. Monowar, “Autoembedder: A semi-supervised DNN embedding system for clustering,” *Knowledge-Based Systems*, vol. 204, pp. 106190, 2020.
- [28] K. Pulli, A. Baksheev, K. Korniyakov and V. Eruhimov, “Real-time computer vision with OpenCV,” *Communications of the ACM*, vol. 55, no. 6, pp. 61–69, 2012.
- [29] M. F. Mridha, A. Q. Ohi, M. A. Ali, M. I. Emon and M. M. Kabir, “BanglaWriting: A multi-purpose offline Bangla handwriting dataset,” *Data in Brief*, vol. 34, pp. 106633, 2021.
- [30] A. Mikołajczyk and M. Grochowski, “Data augmentation for improving deep learning in image classification problem,” in *Proc. of 2018 Int. Interdisciplinary PhD Workshop (IIPhDW)*, Poland, pp. 117–122, 2018.
- [31] M. F. Mridha, A. Q. Ohi, M. M. Monowar, M. A. Hamid, R. M. Islam *et al.*, “U-vectors: Generating clusterable speaker embedding from unlabeled data,” *Applied Sciences*, vol. 11, no. 21, pp. 10079, 2021.
- [32] U. V. Marti and H. Bunke, “A full English sentence database for off-line handwriting recognition,” in *Proc. of the Fifth Int. Conf. on Document Analysis and Recognition. ICDAR’99*, Bangalore, India, pp. 705–708, 1999.
- [33] Y. Hannad, I. Siddiqi and M. E. Y. El Kettani, “Writer identification using texture descriptors of handwritten fragments,” *Expert Systems with Applications*, vol. 47, no. 5, pp. 14–22, 2016.