# EXPLORATION OF NOVEL APPROACHES FOR OFFLINE WRITER IDENTIFICATION USING HANDWRITTEN WORDS

*A*

*Thesis submitted*

*for the award of the degree of*

**Doctor of Philosophy**

By

**Vineet Kumar**


Department of Electronics and Electrical Engineering
Indian Institute of Technology Guwahati
Guwahati - 781039, Assam, India
May 2024

To

My **parents**

for their blessings, love and support

# Certificate

This is to certify that the thesis entitled "**Exploration of Novel approaches for offline writer identification using handwritten words**", submitted by **Vineet Kumar** (186102017), a research scholar in the *Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati*, for the award of the degree of **Doctor of Philosophy**, is a record of an original research work carried out by him under my supervision and guidance. The thesis has fulfilled all requirements as per the regulations of the institute and, in my opinion, has reached the standard needed for submission. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Date:

Place: Guwahati

Dr. Suresh Sundaram

Dept. of Electronics and Electrical Engg.,

Indian Institute of Technology Guwahati,

Guwahati - 781 039, Assam, India.

# Acknowledgements

First and foremost, I feel it as a great privilege in expressing my deepest and most sincere gratitude to my supervisor Dr. Suresh Sundaram for his excellent guidance throughout my study. His kindness, dedication, hard work and attention to detail have been a great inspiration to me. My heartfelt thanks to him for the unlimited support and patience shown to me. I sincerely thank him for the pain he undertook in scrutinizing every work I presented to them and offering critical comments for improvisations.

I am also very thankful to my doctoral committee members Prof. Prabin Kumar Bora, Dr. Prithwijit Guha, and Dr. Ashish Anand for sparing their precious time out of their busy schedule to evaluate my progress and enrich this work with their invaluable suggestions and feedbacks. I would also like to thank the Head of the Department and other faculty members for their kind help in carrying out this work. I am also grateful to all the members of the research and technical staff of the department, for their help in carrying out my research work. I am thankful to Brij Nandan Tripathi, Pallab Jyoti Dutta, Allen Pattnayak, Harshal Chaudhary, and Kaushik Mazumdar for their comradeship and for being there during both my highs and lows.

Last but not least, I would like to thank my parents and my siblings. Without their blessing and support this Ph.D. journey would not have been possible.

*(Vineet Kumar)*

# Abstract

In this thesis, we explore novel strategies for identifying the authorship of off-line handwritten word images. Handwriting falls under the category of behavioural biometric. Over the years owing to its widespread applicability in areas such as forensic analysis, historic document analysis and security, research in the field of writer identification has gained prominence.

In the first work of the thesis, we explore an identification framework that employs the feature maps of layers of a pre-trained CNN network to represent the features of the writer. To begin with, the SIFT algorithm is utilized to extract key-point regions (fragments) across different levels of abstraction, encompassing allographs, characters, or character combinations. These fragments are subsequently processed through a CNN network, yielding feature maps corresponding to convolution layers. The information in these maps is then transformed into a fixed-dimension representation using a modified version of the HOG feature descriptor.

The noteworthy contribution of the proposal lies in harnessing additional cues from the feature maps corresponding to the fragments for writer identification. A measure is proposed to gauge the importance or 'saliency' of feature maps within a CNN layer during training. This measure originates from applying Sparse Principal Component Analysis (SPCA) to Histogram Of Gradient (HOG) features. Once saliency values are obtained, they are combined with HOG representations to create descriptors customized for a CNN layer. These derived descriptors are then passed through a set of SVM classifiers that are scored at two levels. The identity of the handwritten word image is determined by the scores obtained from the fragments that constituent them.

In the second part of the thesis, we explore the notion of similarity learning by utilizing the Siamese neural network with a residual framework for writer identification from handwritten word images. One of the key aspects is in the utilization of a sparse-based model for representing the output feature vector of the Siamese network in a reduced dimensional space. Further to this, we also formulate a divergence-based approach for assigning a saliency score to each component in the sparse representation based on their discriminatory power. To the best of our knowledge, the proposed work is the first of its kind to employ the Siamese neural network for the problem of offline writer identification.

In the third work, we propose an end-to-end framework based on a multi-stream Convolutional Neural Network (CNN) for establishing the authorship of handwritten word images. The network is trained on image fragments and utilizes two parallel modules. One module adopts a writer-dependent training approach to extract writer-specific details, while the other considers a writer-independent strategy to capture global features across all writers. This dual network architecture enables the network to effectively capture the intricate characteristics of a fragment contributed by a writer.

Further to the above contribution, we investigate the integration of an attention mechanism into our proposed two stream network. This in a way enriches the representation power of the network by highlighting important regions within the fragments of the writers. By generating attention weights, we identify areas of importance in a fragment that further assists in refining the ability of the network to discern relevant features.

The efficacy of all the proposed work in the thesis is demonstrated on three publicly available database namely: IAM, CVL and CERUG-EN. The results obtained are found to be promising when compared to prior works.

# Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| SIFT | Scale Invariant Feature Transform |
| CNN | Convolutional Neural Network |
| HOG | Histogram of Oriented Gradients |
| PCA | Principal Component Analysis |
| SVM | Support Vector Machine |
| SVD | Singular Value Decomposition |
| CVL | Computer Vison Lab |
| IAM | Institut für Informatik und Angewandte Mathematik |
| CERUG-EN | Chinese-English database of the University of Groningen |
| EMNIST | Extended Modified National Institute of Standards and Technology |
| WI | Writer Independent |
| WD | Writer Dependent |

# List of Symbols

| | |
|---|---|
| $\boldsymbol{A}$ | Singular value matrix of SVD |
| $a_{ii}$ | $i^{th}$ diagonal entry of the singular value matrix |
| $\alpha^{i}_{j,k}$ | Sparse principal component associated with $k^{th}$ sparse direction for $j^{th}$ fragment of $i^{th}$ writer |
| $b^{l}_{j}$ | bias term associated with $j^{th}$ feature map of $l^{th}$ convolution layer |
| $B$ | number of bins for histogram based representation |
| $\boldsymbol{\beta}$ | Sparse approximation of the principal direction |
| $\beta_{j,i}$ | attention weights for a pair of input $i$ and output $j$ |
| $\lambda$ | Regularization parameter |
| $D^{i}$ | Combined classification score of $i^{th}$ writer word fragments |
| $d(P\|\|Q)$ | divergence score between two distributions $P$ and $Q$ |
| $\boldsymbol{D_k}$ | Divergence matrix corresponding to $k^{th}$ sparse component |
| $D^{i}$ | Overall classification score relative to $i^{th}$ writer word fragments |
| $\varepsilon_i$ | Slack variable |
| $\hat{e}_{j,k}$ | Modified sparse coefficient for $k^{th}$ component in $j^{th}$ fragment obtained from $\tilde{\boldsymbol{\alpha}}$ |
| $\boldsymbol{E}$ | Entropy measure |

| | |
|---|---|
| $\hat{\boldsymbol{E}}$ | Modified sparse matrix |
| $\mathbf{f}(\mathbf{x})$ | feature space transformation of input in query vector |
| $f_j^i$ | Feature representation corresponding to the *nth* fragment for the $j^{th}$ writer |
| $\gamma$ | Inverse of standard deviation of the RBF kernel for SVM |
| $\mathbf{g}(\mathbf{x})$ | feature space transformation of input in key vector |
| $\mathcal{H}_y$ | Histogram representation corresponding to $y^{th}$ sub-block of feature map |
| $H_{kb}^i$ | Distribution of $k^{th}$ sparse principal component of the $i^{th}$ writer in the $b^{th}$ histogram bin |
| $\boldsymbol{h}$ | Feature vector representation |
| $\mathbf{h}(\mathbf{x})$ | feature space transformation of input in value vector |
| $\lambda_i$ | Weight assigned to $i^{th}$ convolution layer classifier |
| $\mu_k$ | Mean of $k^{th}$ component |
| $M_j$ | $j^{th}$ Section of Convolution feature map |
| $m$ | Number of division across rows in an image |
| $n$ | Number of division across columns in an image |
| $\mathbf{o}$ | Attention layer output |
| $p$ | Probability distribution |
| $\boldsymbol{P}$ | Probability distribution matrix |
| $\sigma_k$ | standard deviation of $k^{th}$ component |
| $\boldsymbol{s}_{xy}$ | Histogram representation corresponding to $x^{th}$ sub-image in block $y$ |
| $S_j^i$ | Average pooled feature map representation obtained for $j^{th}$ fragment of $i^{th}$ writer |
| $\tilde{S}_j^i$ | Pre-saliency feature map representation obtained for $j^{th}$ fragment of $i^{th}$ writer |
| $\hat{S}_j^i$ | Post-saliency feature vector obtained for $j^{th}$ fragment of $i^{th}$ writer |

| | |
|---|---|
| $t$ | Number of sub-images in a block |
| $\boldsymbol{U}$ | Left singular vector of SVD |
| $\boldsymbol{V}$ | Right singular vector of SVD |
| $W$ | Number of writers in a database |
| $\mathbf{W_f}$ | Weight matrix for query |
| $\mathbf{W_g}$ | Weight matrix for key |
| $\mathbf{W_f}$ | Weight matrix for value |
| $z^i_{fj}$ | Feature map associated with $f^{th}$ filter of a given convolution layer obtained for $j^{th}$ fragment of $i^{th}$ writer |

# 1

# Introduction

## Contents

## 1.1 Introduction

The research area of biometric has been fuelled largely owing to the requirements of personal authentication for information security. The term originates from the combination of two Greek words 'bios' and 'metron' meaning life and measurement respectively. According to this basic definition, it can be considered as a measurement of the characteristics of a human body [2] using statistical methods. With the rapid advancement of technology, the field of bio-metrics underwent a significant evolution.

In the present context, bio-metric systems rely on techniques that analyze the behavioural and physiological traits aiming to identify or differentiate individuals from one another [3]. The physiological bio-metric uses information from specific measurements, dimensions and characteristics of the body such as fingerprint [4], face [5], iris [6] and hand-scan [7]. On the other hand, behavioural bio-metric takes into account the action performed by an individual over a period of time to identify measurable patterns in human activity such as voice [8], signature [9], handwriting [10] and keystroke dynamics [11].

Based on the preceding discussion concerning various bio-metric traits, a logical query arises: what biological measurements / characteristics satisfy the criteria for classification as a bio-metric trait? Jain *et.al* in [3] recommended seven factors on the basis of which the suitability of a bio-metric trait can be ascertained. These include features such as:

- Universality: Every individual should possess the characteristic.
- Distinctiveness: Two individuals must exhibit significant dissimilarity in terms of the characteristic.
- Permanence: The characteristic should display satisfactory in-variance (in relation to the matching criteria) over an extended period.
- Ease of collection: It should be feasible to quantitatively measure the characteristic.■
- Performance: The resources needed to achieve the desired recognition accuracy remain within specified constraints.
- Acceptability: The users of the bio-metric system must demonstrate acceptance of the

system and also feel comfortable in providing their bio-metric traits for its utilization.

- Circumvention: The collected bio-metric traits must be resistant to easy imitation or replication.

The aforementioned elements contribute to a robust biometric profile, although not every modality will meet each criterion. To be effective and practical, a biometric system needs to be tailored to the unique demands of its intended application.

Expanding on the aforementioned point, it is important to underscore that a bio-metric system can function in one of two distinct modes:

(i) **Verification / Authentication mode:** This mode refers to the process of validating a claimed identity of an individual. Such systems are designed to substantiate the identity asserted by a person. In particular, they conduct a comparison between the bio-metric data acquired from an individual and the corresponding template that is previously stored. The comparisons follow a one-to-one approach, whereby the system assesses the provided bio-metric data against a specific template of an individual, thereby confirming or rejecting the claimed identity.

(ii) **Identification mode:** In this mode, the system seeks to establish the identity of an individual by matching it against templates of all users stored in the database. This entails a one to many comparison to be made, which is typically accomplished by classifiers in a multi-category setting

The crux of the present dissertation is in the exploration of strategies for establishing the identity of the user with his / her handwriting.

## 1.2 Handwriting as a bio-metric trait

Handwriting serves as a behavioural bio-metric skill that individuals acquire and refine over time. This skill entails the harmonious coordination between the hands (motor functions under brain control) and the eyes (providing sensory input to the brain). This coordination in turn empower individuals to produce intricate ink patterns and sequences.

The individuality of handwriting is governed by two fundamental factors namely: *genetic*

(biological) and *memetic* (cultural). Genetic factors include features such as

- The structural composition of the hand from a bio-mechanical perspective
- Preference of hand (i.e. left or right-handedness) used for providing the data
- Strength of muscles
- Central nervous system properties

The second factor, namely *memetic* [12] relate to the forms of characters (allographs) that are cultivated through education or acquired by observing the handwriting of others. The interplay of genetic and memetic factors jointly influence the process of habitual writing in an individual. This results in the manifestation of distinctive elements of shape within the writing trace, that can be analyzed while designing a bio-metric system.

Owing to the aforementioned characteristic of handwriting as a bio-metric trait, the research in the field of writer identification has remained vibrant for several decades - with applications spanning diverse fields such as forensic analysis [13], historical document examination [14–16], and security [17] to name a few.

Writer identification encompasses the process of ascertaining the authorship of an unknown document through a comparison with a database of reference documents. It is achieved by utilizing obtained features or descriptors to establish a list of potential authors for the handwritten sample. These are then ranked based on confidence levels, following which a final decision is made by an expert. On the whole, such bio-metric systems assist forensic experts by relieving them of the need to manually examine each image within extensive databases. Instead, they can base their decisions on a concise list of writers that are predicted by the algorithm.

The tradition of identifying individuals through their handwriting extends as far back as the origins of writing itself. It may be worth noting that handwriting, in conjunction with additional methods such as DNA, fingerprints, and material analysis is recognized as a valid and permissible type of evidence in legal scenarios, especially within the domain of expertise of Questioned Document Examiners or Forensic Document Examiners [18, 19].

## 1.3 Overview of writer identification systems

Based on the manner of data capture, writer identification systems can be classified into one of the following two categories:

- Offline data acquisition in writer identification systems pertains to the collection and preparation of static handwritten samples for analysis. It involves obtaining images captured from pages using scanners or digital cameras [20–23]. The resulting images offer a passive representation of handwritten content, primarily conveying spatial information through image pixels that can be further analyzed by utilizing techniques from the realm of image processing.
- Online data acquisition for writer identification involves capturing and analyzing dynamic handwriting patterns in real time using digital devices. This approach differs from offline methods that work with static, handwritten samples. By utilizing specialized tools like digital pens or stylus-equipped tablets, we can record various dynamic aspects of writing, including stroke order, speed, pressure, and timing. As the writer interacts with the digital surface, the device captures and digitizes these intricate characteristics. The acquired data is subsequently processed to extract relevant features, such as pen trajectory and pressure variations [24, 25]. Online data acquisition is particularly valuable for generating a more comprehensive and authentic representation of the writing behaviour of an individual.

Another categorization of writer identification system is with respect to the textual content - namely text dependent and text independent approaches. In the former, a specific piece of text is used for the generation of handwriting samples of a writer and the identification process usually involves the use of a recognizer. Though the use of the knowledge pertaining to the content of the data increases the accuracy of text dependent systems, they fail in scenarios where text documents comprising different contents need to be contrasted. For such applications, text independent writer identification systems become more applicable as they capture the style information of handwriting. Such systems are designed to identify the writer irrespective of the textual content.

## 1.4 Literature review on off-line writer identification system

Over the past twenty years, significant progress has been achieved in the realm of writer identification. The subsequent sub-sections offer a concise overview primarily centred on methods for offline text-independent writer identification. Based on the literature, the works can be divided into one of the following categories: texture, shape, and deep learning-based approaches [26].

### 1.4.1 Identification techniques based on texture

The systems relying on texture analysis consider each handwritten input sample as a different texture. They extract features from the handwriting sample based either on the entire image [14, 27, 28], or around region of interest like blocks, grid cells, connected-components [21–23, 29–31] and writing fragments [32–34].

The initial work carried out in the field of writer identification using a texture-based approach employed frequency-based techniques to extract global traits from a handwritten image. These global features were then used to ascertain the author of a document by analyzing the overall visual and stylistic characteristics of the handwriting. One of the prominent works in this area was carried out by Said *et.al* [35]. In their proposal, the authors employ a multi-channel Gabor filter along with the Grey-Scale Co-occurrence Matrix to analyze a given writing sample at different frequencies and orientations. Likewise, in [36], a technique employing a hidden Markov tree model in the wavelet domain was suggested for identifying a writer from his / her handwriting.

In [37] the authors explored an approach for writer identification based on features extracted using hybrid spectral–statistical measures. In this method, the features are derived from a handwritten image by employing Fourier spectrum analysis. Thereafter, a set of statistical measures (such as mean, standard deviation, smoothness, Uniformity, entropy) is employed to represent the textual features. Last but not least, in [38, 39] the authors proposed pattern-based features from the input data by exploiting Gabor filters for identifying the handwriting.

Apart from the frequency-based approach for extracting textual features from a handwritten

image, spatial-based techniques have also been used to quantify textual features. These methods treat handwriting as a combination of edges and contours, with their statistical distributions being employed for writer description, as in [27]. In [21], the authors utilized a classification scheme based on dissimilarity-based representation obtained from texture descriptors (Local Binary Patterns (LBP) and Local Phase Quantization (LPQ)). Likewise, in [40], alongside Local Binary Patterns (LBP) and Local Phase Quantization (LPQ), Local Ternary Patterns (LTP) were used on fragments extracted from handwritten documents to characterize the given writer. The authors of [41] employed a set of run-length features extracted from the Gray Level Run Matrix to describe the handwritten characteristics of an author. The utility of the edge-hinge and run-length features are considered in a classifier combination scheme using the Dempster-Shafer-theory model in [34].

SIFT-based algorithms [1] owing to their ability to detect distinct and invariant features from an image have been used as robust feature extractors in writer identification algorithms ( [22,23,42,43]). These methodologies extract image key points at various levels of abstraction and use their description for analysis. In [22] the authors used image features segmented from a word image by considering the SIFT descriptor and its corresponding scale and orientation information to characterize the individuality of a writer. In [42], a Gaussian Mixture Model-Universal Background Model (GMM-UBM) is proposed for offline writer identification. In this method, the UBM is constructed by employing the Expectation Maximization Algorithm corresponding to the set of enrolled writers based on the Root-SIFT [44] features extracted from the handwritten image. Thereafter, the UBM is adapted to represent the characteristics of each individual writer. Finally, the identity of the writer is determined based on the output of the SVM classifier trained on the data of the enrolled writers.

In [23], the authors used a combination of SIFT and Root-SIFT to construct a set of Gaussian mixture models namely similarity GMM and Dissimilarity GMM. In essence, these models aimed at capturing the intra-class similarity and inter-class dissimilarity between the same and different writers of the enrolled database respectively. Last but not least, in [43], a technique to identify the handwriting in historical documents is used, wherein besides the SIFT feature, the

path-let feature is employed to capture the style of a writer. These are subsequently leveraged by employing an encoding method called bagged VLAD.

### 1.4.2 Identification technique using shape based features

Techniques based on this approach divide the image into a group of segmented shapes and incorporate statistical descriptions of their features to characterize the handwriting. More specifically, they utilize a code-book produced by a clustering algorithm to capture the different unique styles of a writer [20, 45–47]. A prominent work of the same is that of [45], where the contour of the connected component was employed to construct a code-book for capturing the common patterns from a handwritten document. Based on this generated code-book, a histogram is constructed for each writer, that acts as a feature descriptor to establish the identity using the distance-based approach. In another exploration [20, 46, 47] , the authors used graphemes to construct the code-book, which served as a primary feature in addition to other contour-based features for identifying a writer.

The proposal in [48] deals with detecting junctions from a handwritten image, that are obtained by analyzing the distribution of stroke lengths in all directions around a reference point within the textual content. Thereafter, these junctions are subsequently used to create a code-book based representation called junct-lets using which the identity of a writer is determined. This method is based on the proposition that junction shapes are not identical across writers but vary depending upon the writer in question.

In [23], the authors proposed a Discrete Cosine Transform based writer identification scheme. In this algorithm, the image is broken into small, overlapping blocks to construct a set of universal code-books based on which the final decision is made by applying the majority voting rule. In [49], the efficiency of an implicit shape code-book-based approach for identifying writers from handwritten images is investigated. This involves identifying key points within the handwriting and clustering them to construct a code-book. The characterization of a writer is then based on considering the probability distribution of generating the patterns from the code vectors in the code book.

### 1.4.3 Identification using deep learning based approaches

The advancements of machine learning in the recent decade have led to the prominence of deep learning-based methods in the area of computer vision-related applications. These methods, unlike conventional handcrafted features, learn data-dependent characteristics automatically from the training samples, thus resulting in higher performance.

The pioneering attempt using deep learning for writer identification was introduced by Fiel *et al.* [50]. In their work, Convolutional Neural Networks (CNNs) are employed to generate a feature vector for each writer. These feature vectors are subsequently compared with those stored in the database for writer identification. In order to generate this vector, the CNN is trained on the training database of enrolled writers. Following the training phase, the classification layer is removed, and the output of the penultimate fully connected layer is employed as the feature vector. Subsequently, the nearest neighbour classifier is used for identification purpose.

In [51], the authors proposed a clustering-based approach for training the weights of a convolution network in an unsupervised manner, thereby enabling it to learn robust local features. In this technique, surrogate classes are generated by clustering the training datasets, with each cluster index serving as a representation for a distinct surrogate class. Following this, as in [50] penultimate layer of the trained CNN layer is used for a classification task.

An empirical study of the effect of individual convolution layers on the writer identification rate was performed in the work [52]. In particular, the authors employed a deep transfer convolution neural network (CNN) to recognize writers based on handwritten text line images. The evaluation focused on various frozen layers of the CNN involving convolution and fully connected layers to assess their impact on the writer identification rate.

In the exploration [53] the authors employed a hybrid approach combining deep learning and hand-crafted descriptors to capture patterns from handwritten images. Local patches from handwritten images are extracted and processed in parallel using deep learning and hand-crafted descriptors to create local descriptions, which are combined to form a description matrix. Subsequently, the vector of locally aggregated descriptors (VLAD) encoding is applied to this ma-

trix to produce a 1-D feature vector that represents the description of the writer. Likewise, in [54] a segmentation-free Writer identification model is proposed by utilizing a convolution neural network (CNN) along with a weakly supervised region selection mechanism. This model processes unsegmented text documents to generate the identity of the writer along with a region probability map. The map contains probability vectors at each cell location, indicating the likelihood of each region belonging to a particular writer. The authorship of the document is established by employing a voting mechanism amongst the selected cell regions.

In [55], the authors proposed a writer identification system that operates by extracting key points from handwriting. The region around these keypoints is fed to a convolutional neural network (CNN) for the purposes of feature learning and classification. Lastly, in [56], a convolution-based network referred to as Deep Writer Identification Network (DeepWINet) is proposed for identifying a writer. The proposed model is assessed through two distinct evaluation scenarios. In the first scenario, the CNN activation features extracted by DeepWINet from connected components of the writing are used as input to a nearest neighbour classifier for writer identification. In the second scenario, DeepWINet is evaluated as an end-to-end CNN network, and the predicted results are combined using an efficient strategy called score averaging component-decision combiner.

All of the the aforementioned works use page or text-line level data to learn the descriptors of the writers. However, in applications such as forensic examination, often a situation arises where a decision needs to be made based on analyzing a very small amount of handwritten text, such as words. This task poses a challenge as the writer-related style information is restricted in contrast to page / text-line level input.

The pioneering exploration in the direction of word-level based off-line writer identification is that of [57]. Here, the authors employ a multi-task framework to enhance writer-related information by incorporating attributes learned in the auxiliary task along with features from the main task. In a subsequent work [58], the same authors proposed a deep neural network (*FragNet*, inspired from *Fraglets* and *BagNet*) to extract powerful features from the input word images. This network consists of two pathways: the feature pyramid pathway and fragment

pathway. The former is used for feature map extraction while the latter is trained to identify the writer using fragments extracted from the input image in combination with the feature maps produced by the feature pyramid.

In another contribution [59], the authors introduce an end-to-end neural network system known as global-context residual recurrent neural network (GR-RNN) for identifying writers from handwritten word images. The system combines global context information and a sequence of local fragment-based features. Further, in order to capture the spatial relationships between these fragments, a recurrent neural network is used to enhance the discriminative power of the local fragment features. The most recent work related to writer identification using word-level images is that of [60], where a Residual Swin Transformer classifier is designed to capture both local and global handwriting styles effectively from single-word images. The model employs transformer blocks to handle the local information by interacting with individual strokes and utilizes holistic encoding with the identity branch and global block to capture global handwriting characteristics.

## 1.5 Contribution of the thesis

In this thesis, we propose a set of novel techniques for offline text-independent writer identification utilizing word images. Our main focus in this work is to represent features in an effective way by utilizing a convolution network trained on a writer-independent framework. Our proposal has been laid out in three main contributing chapters and we elaborate them in the following sub-sections. All the approaches being developed are text-independent and adaptable to input word images of varying sizes.

It should be noted that in each of the three contributing chapters, the input being fed to the proposed models consists of fragments extracted from word images. These fragments are used to obtain feature representations, which are then quantified by a score in $(0, 1)$. The writer identity for an input test word image is established by accumulating the scores of the fragments that constitute it.

### 1.5.1 Chapter 2

In this Chapter, we explore information from the feature maps of a Convolution Neural Network (CNN) and investigate its applicability in identifying the authorship of handwritten word images. Our methodology begins by adapting the Scale-Invariant Feature Transform (SIFT) algorithm to extract multiple key-point regions referred to as "fragments" at varying levels of abstraction. These fragments can encompass elements such as allographs, individual characters, or even combinations of characters. These fragments are subsequently processed through a CNN, yielding feature maps corresponding to convolution layers. The information in these maps is then transformed into a fixed-dimension representation using a modified version of the HOG feature descriptor.

The noteworthy contribution of our proposal lies in harnessing additional cues from the feature maps of the fragments for writer identification. We propose a measure to gauge the importance or 'saliency' of feature maps within a CNN layer during training [1].

We estimate the saliency value by projecting the modified HOG features obtained from each feature map of a convolution layer onto a common subspace. The projection space is constructed using the concept of Sparse Principal Component Analysis (SPCA) [61], which employs the conventional Principal Component Analysis (PCA) in a regression-based framework. The resulting sparse principal components generated are used to construct a histogram, based on which the proposed saliency value is calculated.

The estimated saliency values are incorporated with the respective feature map outputs of the convolution layer to generate descriptors for the fragments. More specifically, two strategies for description of the fragments of the word are proposed in this work. These are related to the order in which the saliency values and HOG information are utilized for pooling the feature map information. For sake of convenience, these two approaches are termed as pre-saliency based pooling and post-saliency based pooling respectively. The proposed descriptors of the fragments of the word are passed through a set of writer specific SVM classifiers, that are

[1] Said in another way, for a particular convolution layer of the network with $F$ feature maps, we assign a saliency score to each of them, resulting in $F$ values

scored at two levels. Thereafter, the scores across all fragments are accumulated to decide on the identity of the writer.

The efficacy of the proposed work has been demonstrated on the segmented word images of the CVL, IAM and CERUG-EN datasets. The results obtained are shown to be promising when compared with previous works.

### 1.5.2 Chapter 3

In this study, we suggest a dissimilarity-based approach for representing the features of the handwritten data and utilize the same for writer identification exploring the utility of a variation of a neural network known as the Siamese network. To the best of our knowledge, the present work is the first of its kind to explore the application of the Siamese framework for offline text-independent writer identification.

The primary objective of our approach is to investigate the idea of similarity learning by examining the connections between different fragments generated by a writer. These fragments, when passed through the Siamese network generate a high-dimensional feature vector representation based on the penultimate layer output. To combat the issues related to the curse of dimensionality, we consider representing the output of the Siamese network in a reduced dimensional feature space. To be more specific, we utilize the concept of sparse PCA from Chapter 2, to project the learnt features in a lower dimension.

Further to the above contributions, we also explore to enhance the discriminating strength of a sparse principal component by introducing a saliency score that is obtained from its histogram-based representation during the training phase. In the identification phase, the sparse component is modified in accordance with this saliency score (using an adaptive power law transformation) to generate a descriptor for each of the fragments of the input word.

The derived descriptor is then passed through a set of writer-specific SVM classifiers, providing a score for each fragment. These scores are subsequently accumulated across all fragments of a given word image with regards to each enrolled writer-specific classifier. The identity of the word sample is determined based on the maximum value of scores represented across the

writer-specific SVM classifiers

Experiments conducted on the word images of the IAM, CVL, and CERUG-EN databases suggest that the sparse component representation of the penultimate layer of the Siamese network out-performs to the original high-dimensional feature description. Moreover, the proposed modified descriptors of the fragments of the word improve the overall identification accuracy of the system.

### 1.5.3 Chapter 4

In this Chapter, we propose an end-to-end framework based on Convolutional Neural Networks (CNNs) for establishing the authorship of handwritten word images. The network is trained on image fragments extracted from word images and utilizes two parallel CNN modules. One module adopts a writer-dependent training approach to extract writer-specific details, while the other considers a writer-independent strategy to capture global features across all writers. The writer-dependent part excels at capturing spatial hierarchies and local patterns, such as strokes and textures, which are unique to each writer's handwriting. Their ability to learn discriminative, tailored features enhances accuracy in scenarios where the writer is known. On the other hand, the writer-independent part focuses on capturing relationships between pairs of samples, learning invariant, distance-based embeddings that are independent of specific writers. This design is particularly effective in open-set scenarios, where the model must generalize to new or unseen handwriting. Together, these networks complement each other—CNNs capture detailed, writer-specific traits, while Siamese Networks provide robustness and adaptability, ensuring effective performance across both closed-set and open-set conditions.

The features obtained from these two networks are combined and fed to the classification block consisting of a set of fully connected layers which assign a probability score to these fragments using a soft-max activation function. Thereafter, the obtained scores are accumulated across all fragments of a given word image. The identity of the word sample is determined based on the maximum value of the accumulated scores obtained across the classification nodes.

Further to the above contribution, we investigate the integration of an attention mechanism

into our proposed two-stream network. We adopt a self-attention mechanism that leverages the relationships between pixels within the feature maps of a convolutional layer. This in a way enriches the representation power of the network by highlighting important regions within the fragments of the writers. By generating attention weights, we identify areas of importance in a fragment that further assists in refining the ability of the network to discern relevant features.

Experiments conducted on the word images of the IAM, CVL, and CERUG-EN databases suggest that the dual network architecture with the attention mechanism enables the network to effectively capture the intricate characteristics of a fragment contributed by a writer. The results obtained are promising when compared with prior end-to-end convolution-based frameworks for identifying writers using word images as input.

To summarize, in this thesis, we propose strategies for writer identification that solely rely on the utility of the fragments that make up a hand-written word.

## 1.6 Conclusion

In this Chapter, we initially provided a brief overview of biometric systems. Thereafter, we discussed in detail the overall process of offline writer identification along with an elaborate survey of the prior works on offline text-independent writer identification available in the literature. Finally, we highlight the contribution of all the working chapters, thus setting a roadmap to the preset thesis.

# 2

# Exploring Novel Pooling Strategies for CNN Feature Maps

## Contents

## 2.1 Introduction

In this Chapter, we propose a writer identification system that utilizes attributes of a given word image at the character and sub-character level (referred to as fragments) to exploit novel clues from feature maps of a CNN network. Typically each layer of a convolution network contains multiple filters capturing a diverse set of features from a given input fragment. An important aspect worth investigating is to explore whether such features can provide us *a-priori* information to improve the overall performance of the writer identification system for handwritten words as input.

As a step in this direction, the crux of our work is in quantifying the relative information available with respect to each feature map of a given convolution layer. Accordingly, we introduce the term 'saliency' in this work and estimate its value by projecting a modified version of HOG features obtained from each feature map of a convolution layer onto a common subspace. The projection space is constructed using the concept of Sparse Principal Component Analysis (SPCA) [61], which employs the conventional Principal Component Analysis (PCA) in a regression-based framework. The resulting principal components generated are used to construct a histogram, based on which the proposed saliency value is calculated.

The estimated saliency values are incorporated with the respective feature map outputs of the convolution layer to generate novel descriptors using the modified version of HOG representation. More specifically, two strategies for description of the fragments of the word are proposed in this work. These are related to the order in which the saliency values and HOG information are utilized for pooling the feature map information. For sake of convenience, these two approaches are termed as pre-saliency based pooling and post-saliency based pooling respectively. The proposed descriptors of the fragments of the word are passed through a set of writer specific SVM classifiers, that are scored at two levels. Thereafter, the scores across all fragments are accumulated to decide on the identity of the writer.

### 2.1.1 Block schematic of our proposal

Figure 2.1 summarizes the overview of our proposed algorithm. The input word image is divided into a number of fragments of different sizes. For each of these individual fragments, the following set of operations is performed:

- Generation of feature maps in each convolution layer of the CNN network.
- Combining feature maps of a convolution layer using the proposed saliency-based approach to generate descriptors using HOG representation
- Generation of scores from a trained writer-specific SVM classifier.

It is worth emphasizing that scoring of SVM is carried out at two levels (Level 1 and 2). In the first level, the input provided to the SVM involve the descriptor for each of the individual SIFT fragments of a word image. This results in the generation of a score in the range $(0, 1)$ for each of the fragments. These are then accumulated across all the fragments of the input word, leading to an average score with respect to the given convolution layer. Thereafter, in the subsequent level, the scores generated relative to the set of convolution layers from level 1 are fused together using a combination scheme. The authorship of the test word is assigned by considering the maximum of the combined score at the second level.

The main highlights of this work can be summarized as follows:

- Introducing the idea of saliency for each of the feature maps in a convolution layer.
- Proposing an entropy based approach to estimate saliency values for each feature map of a convolution layer using sparse PCA.
- Exploiting the saliency values of convolution layer feature maps to generate a modified writer descriptor.
- Suggesting a modified HOG based representation for the feature maps in a CNN.

The rest of the chapter is organized as follows. We begin by describing the process of fragment generation using the SIFT keypoint detector in Section 2.2. This is followed by a detailed discussion of the CNN block used for feature extraction in Section 2.3. In Section 2.4, we provide details of the HOG approach for obtaining the fixed-size representation of the feature maps. The proposed methodology for associating saliency values to each layer of a convolution

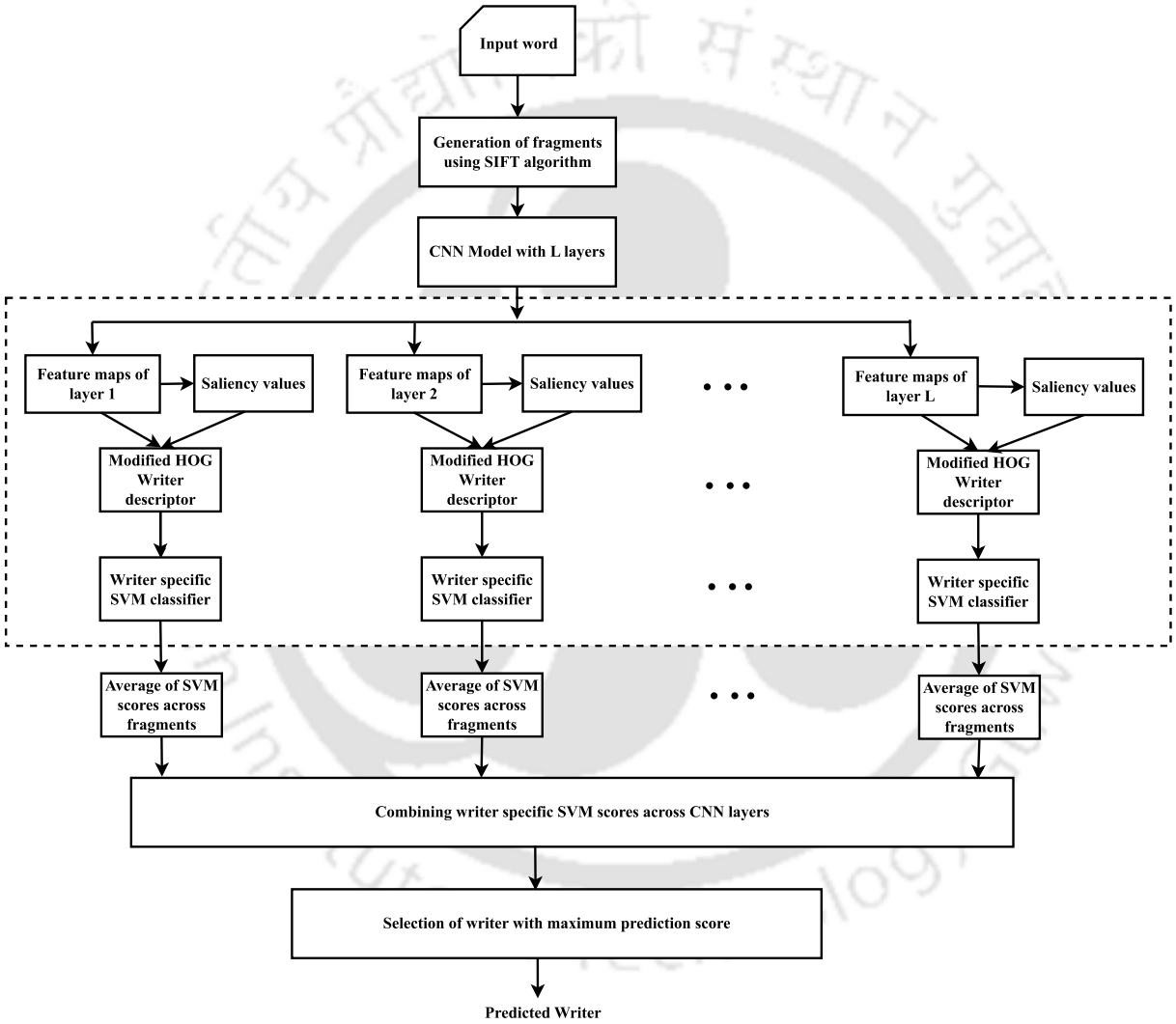



**Fig. 2.1:** Pictorial Overview of our proposed system. The blocks enclosed in dotted lines represent the operations that are performed on each writer fragments. For sake of clarity, L represents the number of convolution layers in the CNN model.

layer is discussed at length in Section 2.5. In Section 2.6 we focus on the saliency-based pooling strategies employed for generating a feature representation for each of the fragments of the word. In Section 2.7, the set of writer specific SVM classifiers together with their scoring mechanism is described. With regards to the experiments, the datasets used for evaluating our algorithm are presented in Section 2.8 together with the training and testing protocol. In Section 2.9 we carry out a set of experiments to evaluate the efficacy of the proposed writer identification algorithm and compare its performance with prior works. Finally, we summarize the findings of the study in Section 2.11.

## 2.2 Generation of fragments

Locating features across images is a common task prevalent in various tasks of computer vision such as object detection [62], face recognition [63], image retrieval [64] and image quality assessment systems [65–67] to name a few.

Many deep learning-based techniques in the field of writer identification use dense sampling for extracting multiple handwriting blocks from a given input sample. However, prior studies [68, 69] have suggested that all the characters in a handwritten image need not be equally informative in characterizing the writer. Secondly, dense sampling may result in important writer features getting split across several windows thus leading to a loss of information. As a circumvention to this, important attributes in an image (such as corners) are identified and the region around them is extracted as fragments using a pre-determined window size. As such, these fragments are more likely to discriminate against a writer effectively as compared to the image patches extracted using dense sampling.

A simple key-point detector such as Harris corner works effectively when the overall features of the images under consideration are similar (in terms of scale and orientation). However, in the case of a handwritten document, a writer may create a text that can differ in scale, whereby a fixed-sized window may not be always effective in capturing different-sized features. This observation motivated us to employ the popular scale invariant-based interest point detector (SIFT) [1] for extracting the fragments around each of the key points detected in the

image [22, 23, 42].

The SIFT keypoint detector algorithm is divided into three stages: In the first stage, an input image is broken into a Gaussian pyramid, with its size at each octave being half of the preceding one. In the second stage, the image at each octave is used to generate Difference of Gaussian (DoG) images, by convolving them with a series of Gaussian filers of different variances. In the third stage, the difference of Gaussian (DoG) images generated in the above step is used to detect the local extrema (either maxima or minima). This involves comparing a pixel with its 26 neighbours, within a 3×3 window at both the current and neighbouring. This list of identified extremums comprises candidate key points, which undergo additional filtering to finally yield a set of stable keypoints. The scale information of these stable key points is used to segment the area around the Region of Interest (ROI). The whole process of extracting keypoint using the SIFT algorithm is illustrated using a toy example shown in Figure 2.2.

Some of the key points extracted using the SIFT algorithm on an input word image from the CVL dataset are shown in Figure 2.3.

## 2.3 CNN model

Once the writing fragments are extracted, these are mapped to feature for subsequent identification. For feature learning, we use a convolution-based network as they are known to outperform conventional hand-crafted based statistical features.

Our network is trained using the EMNIST dataset [70]. This dataset contains scanned handwritten digits and characters each of size $(28 \times 28)$ divided into 62 classes comprising [0-9], [a-z] and [A-Z]. From this dataset, we select the handwritten English alphabets [a-z] containing 190,998 samples of which 178,998 is used for training and 12,000 for testing the network. Some of the sample images of the EMNIST dataset are shown in Figure 2.4.

A CNN architecture consists of a combination of several convolution layers each having a set of filters whose number increases progressively as we go deeper into the network. Each set of filters in a convolution layer extracts features from the preceding input layers using the convolution operation. These filter weights relative to each convolution layer are learned during

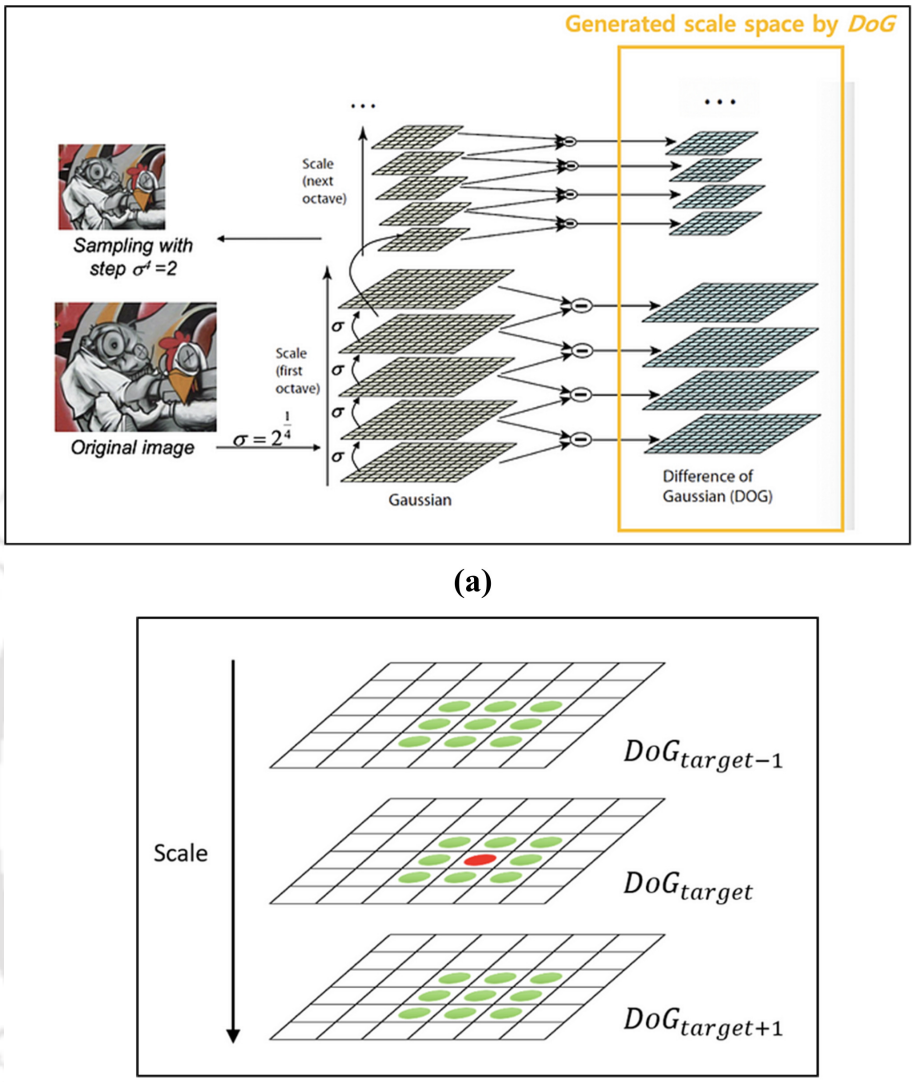


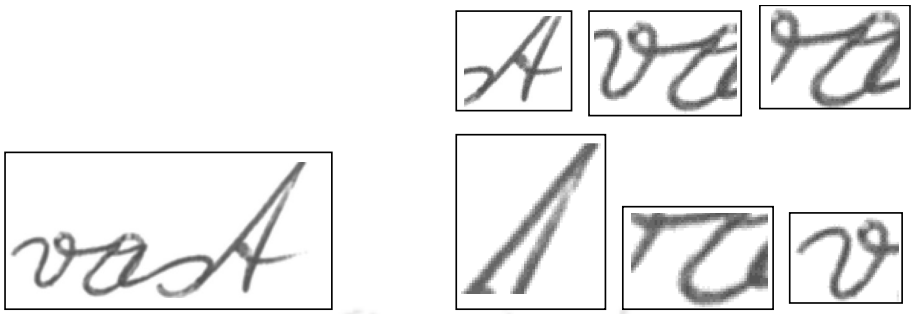

**Fig. 2.2:** Illustration of SIFT algorithm (adapted from [1]). Sub-figure (a) shows the approximated scale space obtained by Gaussian pyramid along with Difference of Gaussian (DoG) operation, and (b) Keypoint localization using local extrema detection.

the training stage by tuning them continuously to optimize an appropriate loss function. The convolution operation in a CNN model can be mathematically represented as:

$$x_j^l = f\left(\sum_{i \in M_j} x_i^{l-1} \times k_{ij}^l + b_j^l\right) \tag{2.1}$$

Here, $x_j^l$ represents the $j^{th}$ feature map of layer $l$ with $f$ being the non-linear activation function. The notation $M_j$ denotes a section of the input feature map, while $k$ and $b$ correspond to the convolution filter kernel and bias term respectively.

The block diagram of our network is shown in Figure 2.5. This network bears semblance to



(a) (b)

**Fig. 2.3:** Fragment generation from an input word using SIFT. (a) An input word image from CVL dataset, (b) Extracted fragments.

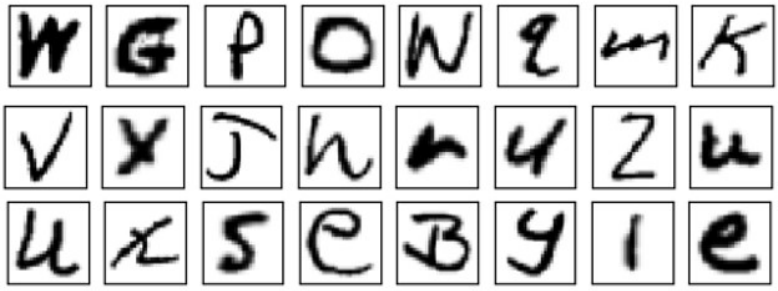


**Fig. 2.4:** Depiction of a few sample images from EMNIST database used for training the CNN network.

the Lenet 5 architecture and consists of six convolution layers. Each convolution layer has been assigned a value from 1 to 6 (conv 1 to conv 6), followed by an appropriate number of filters ( 32 or 64) and their size [$(3 \times 3)$ or $(5 \times 5)$) respectively. Each convolution layer in our network is succeeded by a Rectified Linear Unit (ReLU) and batch normalization layer. In our model, we have replaced the conventional max pooling operation with a strided convolution layer, as it emphasizes on learning the pooling values rather than merely fixing them. This modification enhances the performance of the network without significantly increasing the network parameters [71]. In addition to the convolution layers, the network also contains a fully connected layer and an output layer comprising 126 and 26 nodes respectively. A fully connected layer acts as a mapping function between the outputs of a convolution layer, thus helping in classifying the data into different categories.

The use of a writer-independent dataset such as EMNIST [70] ensures a good generalization of the network during the process of training. This is owing to the semblance of its alphabets to the fragment segmented from the handwritten word samples of the IAM, CVL and CERUG-EN

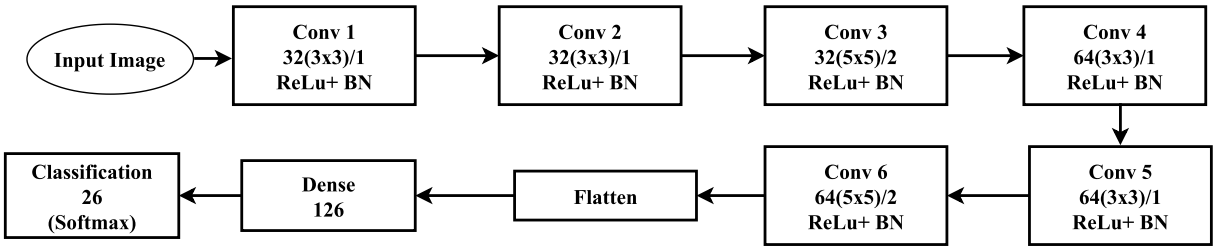



**Fig. 2.5:** CNN network used for training characters of the EMNIST dataset. Each convolution layer is succeeded by a Rectified Linear Unit (ReLU) and batch normalization (BN) layer. For convenience, each convolution block is represented as Conv $n$, $f(m \times m)/p$. The notations $n, f, m, p$ indicate the index of the convolution block, number and size of filter together with the value of the stride respectively.

datasets to be used later in this work.

It is worth emphasizing that once the Lenet-based convolution network is trained on the EMNIST dataset its weights are frozen. These weights are then used to obtain the feature map for each of the fragments of the word image.

## 2.4 Modified Histogram of Oriented Gradients

Given the variable-sized feature maps obtained from the fragments of the word, it is imperative to convert them to a fixed-sized representation is needed. In order to achieve this, we consider a modification of the Histogram of Oriented Gradients (HOG) [72] based descriptor, as discussed below:

(i) An input feature map of size $(M \times N)$ is first divided into $(m \times n)$ number of patches, with each patch consisting of a sub-image of size $(r_{cell} \times c_{cell})$.

$$r_{cell} = \left\lceil \frac{M}{m} \right\rceil$$
$$c_{cell} = \left\lceil \frac{N}{n} \right\rceil \tag{2.2}$$

(ii) The resulting patches of a feature map are grouped into $b$ number of blocks, with each block containing $t$ sub-images.

(iii) Magnitude and orientation of gradient vector is computed for each sub-image using the Sobel Kernel.

(iv) The magnitude and orientation information obtained from the above step is used to construct a histogram relative to each sub-image in a block. The horizontal axis of the histogram represents the orientation information discretized into a fixed set of bins in multiples of $\phi = \left\lceil \frac{360}{k} \right\rceil$ degrees, where $k$ refers to the number of bins in the histogram. The location ($l$) of the bin to be voted is selected based on the orientation ($l = \left\lceil \frac{Orientation}{\phi} \right\rceil$), and the value of the vote as such corresponds to the magnitude of the gradient vector of the pixel under consideration in a sub-image.

(v) Corresponding to each of the $x^{th}$ sub-images ($1 \leq x \leq t$) in the $y^{th}$ block ($1 \leq y \leq b$), a histogram $\boldsymbol{s}_{xy}$ is constructed as explained in step 4. Each of the histograms corresponding to a given block is concatenated and $L_2$ normalized to generate a $B \times t$ dimensional histogram representation $\mathcal{H}_y$ defined as:

$$\mathcal{H}_y = [\boldsymbol{s}_{1y}\ \boldsymbol{s}_{2y}\ ....\ \boldsymbol{s}_{xy}] \quad 1 \leq x \leq t, 1 \leq y \leq b \tag{2.3}$$

(vi) Finally, the histograms for each of the $b$ blocks are stacked together to form a $B \times t \times b$ dimensional column feature vector representation.

$$\boldsymbol{h} = [\mathcal{H}_1\ \mathcal{H}_2\ ...\ \mathcal{H}_b]^T \tag{2.4}$$

Each of the preceding steps involved in the process of histogram generation is illustrated with the help of Figure 2.6 for the case where the values of $m$,$n$, $t$ and $b$ each are 4.

As we shall see in the following Section, the HOG representation of the feature maps in a convolution network is utilized in determining their saliency values during the training phase.

## 2.5 Proposal of saliency values and their estimation

Storing the information corresponding to all the feature maps of a convolution layer is a memory-intensive and computationally expensive task. A solution to this problem involves combining/pooling the output feature maps of the convolution layer with a notion to summarize the extracted information. The simplest way of achieving this objective is by assigning equal

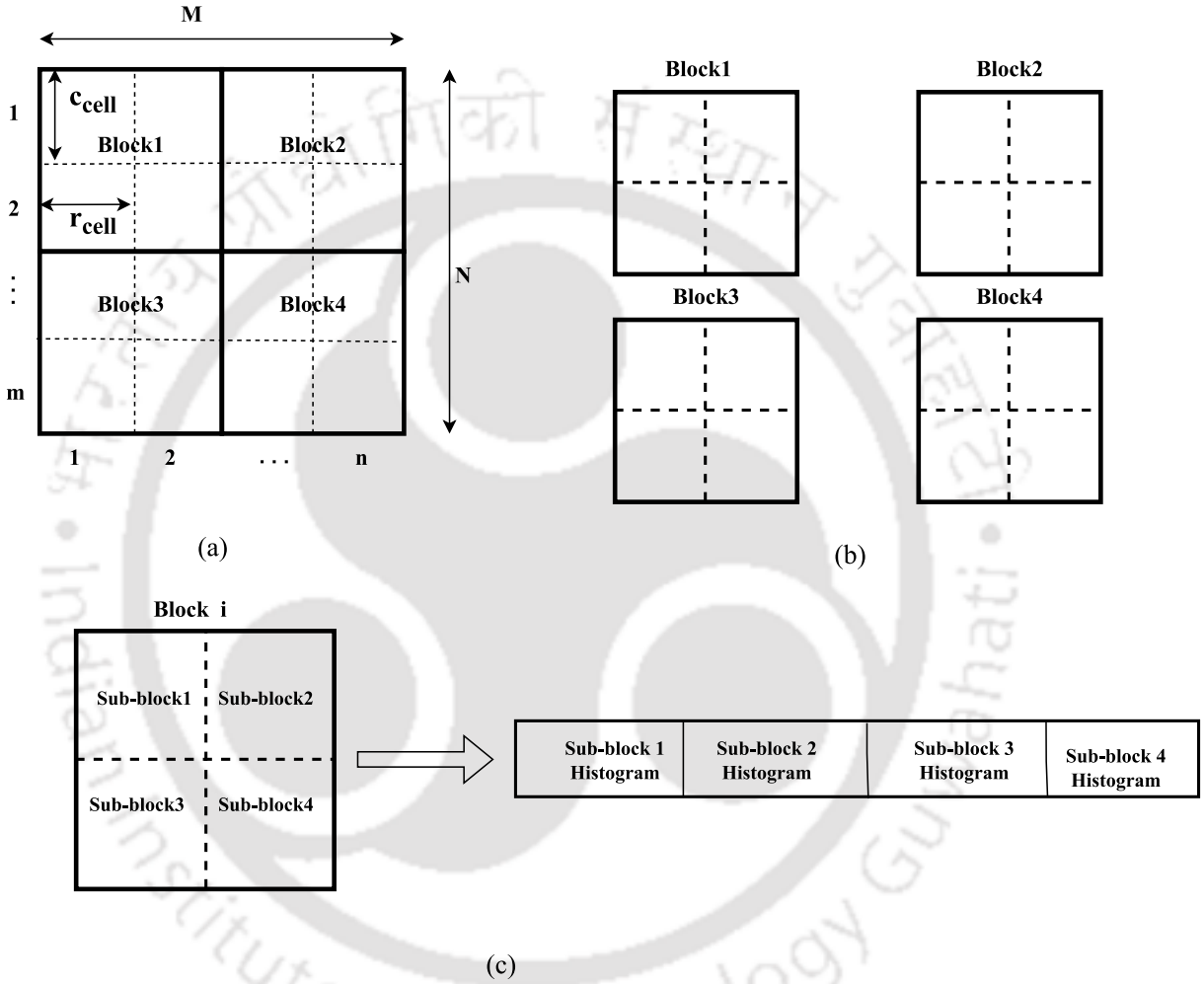



**Fig. 2.6:** Pictorial illustration of the steps involved in generating a HOG feature representation obtained from CNN feature map output of the writer fragments. Here we have selected the values of $m$, $n$, $t$ and $b$ as 4. The sub-figure (a) represents the input image of size $M \times N$ being subdivided into $m \times n$ grids. These grids are grouped to form $b$ number of blocks as shown in sub-figure (b). Each of the $t$ sub-blocks in a given block corresponds to an image patch of size ($r_{cell} \times c_{cell}$). The features corresponding to each of the individual image sub-blocks within a block are represented using a $k$ bin histogram. These histograms are concatenated to generate a $(k \times t)$ dimensional feature vector. In sub-figure (c), we present one such histogram representation of the $i^{th}$ block ($1 \leq i \leq b$). Overall, across all the $b$ blocks, the feature representation will be of dimension $k \times t \times b$.

weights to the output feature map of a convolution layer (average pooling). However, considering the fact that all the feature maps of a convolution layer are not equally informative, we introduce a so called saliency or importance value to each of them based on the amount of information they contain. Our idea has been primarily inspired by works on visual attention that have found promising success in various computer vision applications [73, 74].

We now elaborate in detail on our proposed approach based on entropy for quantifying the degree of importance of each feature map in a CNN layer. For the sake of simplicity, we select a set of $W$ writers, with the assumption that each writer contributes $N$ number of fragments. Corresponding to each of the fragments of the writers, the HOG feature representation is extracted from the feature map of a convolution layer for which saliency value needs to be determined. In the analysis hereinafter, our focus is on obtaining the saliency values of a feature map in a particular layer of CNN. Nevertheless, it may be reiterated that the same procedure is repeated with regard to the saliency values of all feature maps in a CNN layer.

Let $\boldsymbol{X}$ represent a matrix containing the HOG representation of the feature map corresponding to the filter of a particular convolution layer. This matrix can be represented as:

$$\boldsymbol{X} = \begin{bmatrix} \cdots & \boldsymbol{X}_1 & \cdots \\ \cdots & \boldsymbol{X}_2 & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \boldsymbol{X}_j & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \boldsymbol{X}_W & \cdots \end{bmatrix} \tag{2.5}$$

The entries in each row in submatrix $\boldsymbol{X}_j = \begin{bmatrix} \boldsymbol{h}_j^1 & \boldsymbol{h}_j^2 & \cdots & \boldsymbol{h}_j^n & \cdots & \boldsymbol{h}_j^N \end{bmatrix}$ represents the HOG feature representation corresponding to the $n^{th}$ fragment ($1 \leq n \leq N$) for the $j^{th}$ writer each having a dimension of $N \times d$.

With regards to the matrix $\boldsymbol{X}$, following operations are performed:

(i) Extraction of sparse principal components (SPCA) from the HOG features and thereafter constructing a histogram from it.

(ii) Computation of entropy by utilizing the histogram generated in the above step.

(iii) Incorporation of the entropy values to assign a saliency value to each feature map in a convolution layer.

In the following sub-section, we explain in detail each of the above-mentioned steps.

### 2.5.1 Histogram generation using sparse principal component analysis

The first step in our objective of assigning a saliency value to a feature map involves projecting the HOG features of the writer fragments onto a common subspace.

For effectively analyzing multidimensional data, the principal component is one of the most commonly used techniques. It is a statistical procedure that aims to transform the original data into a new coordinate system where the axis are the new principal components (PC). These principal components are orthogonal and capture the maximum variance in the data.

Let $\boldsymbol{X}$ be a real-value data matrix of $NW \times d$ size where $N$ is the number of fragments per writer, $W$ is the number of enrolled writers, and $d$ is the dimension of HOG feature vector. Then $d \times d$ covariance matrix $\boldsymbol{C}$ is given by $\boldsymbol{C} = \frac{\boldsymbol{X}^T \boldsymbol{X}}{n-1}$. Being a symmetric matrix it can be diagonalized as:

$$\boldsymbol{C} = \boldsymbol{V}\boldsymbol{L}\boldsymbol{V}^T \tag{2.6}$$

Where $\boldsymbol{V}$ is a matrix of eigenvectors each column of which represents one eigenvector and $\boldsymbol{L}$ is a diagonal matrix with eigenvalues $\lambda_i$ arranged in decreasing order along the diagonal. These eigenvectors are called the principal axes or principal directions of data. The projection of the data on this principal axis is called the principal components, also known as principal scores, these can be seen as new transformed variables. The $j^{th}$ principal component is given by $j^{th}$ column of $\boldsymbol{XV}$.

If singular value decomposition is performed on $\boldsymbol{X}$, we obtain the following decomposition matrix.

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{A}\boldsymbol{V}^T \tag{2.7}$$

where $\boldsymbol{U}$ is the unity matrix (with columns called the left singular vector), $\boldsymbol{A}$ is the diagonal matrix having singular values $\boldsymbol{a}_{ii}$ along the diagonals and columns of $\boldsymbol{V}$ denote the principal

directions for projection. Using the SVD expression the covariance matrix $\boldsymbol{C}$ can be written as:

$$\boldsymbol{C} = \frac{\boldsymbol{V}\boldsymbol{S}^2\boldsymbol{V}^T}{n-1} \tag{2.8}$$

The expression indicates that the right singular vectors correspond to the principal directions (eigenvectors), while the singular values are connected to the eigenvalues of the covariance matrix through the relation $\boldsymbol{\lambda}_i = \frac{\boldsymbol{a}_i^2}{n-1}$. Furthermore, $\boldsymbol{Z} = \boldsymbol{UA}$ is a matrix representing the principal components.

From the above discussion, it is evident that each principal component in PCA represents a linear combination of all input variables. This makes it difficult to interpret which features are most influential in the reduced dimensions. Moreover, PCA lacks an inherent mechanism for feature selection, as it incorporates all features into the components, even those contributing minimal variance. This can lead to the inclusion of noise or irrelevant information, reducing the robustness and interpretability of the model, especially in applications where feature selection is crucial.

Sparse PCA addresses these limitations by introducing sparsity into the principal components, ensuring that only a subset of the original features is retained. This improves interpretability, as it becomes easier to identify which features are contributing most to the components. Additionally, Sparse PCA naturally performs feature selection by zeroing out the coefficients of less important features, reducing the impact of noise and irrelevant information. This is done by formulating PCA as a linear regression-type optimization problem and imposing sparsity constraints. In the formulation being employed in our work sparsity is enforced on $\mathbf{V}$, using an elastic-net penalty, leading to the following optimization problem:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{arg\,min} ||\boldsymbol{z}_i - \boldsymbol{X}\boldsymbol{\beta}||^2 + \lambda||\boldsymbol{\beta}||^2 + \lambda_1||\boldsymbol{\beta}||_1 \tag{2.9}$$

Here, $\hat{\boldsymbol{v}}_i = \frac{\hat{\boldsymbol{\beta}}}{||\hat{\boldsymbol{\beta}}||}$ is a sparse approximation of the principal direction $\boldsymbol{v}_i$ and $\boldsymbol{\alpha_i} = \boldsymbol{X}\hat{\boldsymbol{v}}_i$ is a vector of size $NW \times 1$, whose elements correspond to the values of the $i^{th}$ sparse principal component.

In our implementation, we select $L$ principal directions $\{\boldsymbol{v}_1, \boldsymbol{v}_2 ....., \boldsymbol{v}_L\}$ for projection based

on the variance criterion. The value of $L$ is determined by setting the penalty coefficients $\lambda_1$ and $\lambda$ to 1 and 0.01 respectively in Equation 2.9. Accordingly, we can write the sparse principal components in the matrix form as:

$$\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1\ \boldsymbol{\alpha}_2\ ...\ \boldsymbol{\alpha}_l\ ...\ \boldsymbol{\alpha}_L] \tag{2.10}$$

Expanding the Equation 2.10, we can write,

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha^1_{1,1} & \alpha^1_{1,2} & \cdots & \alpha^1_{1,k} & \cdots & \alpha^1_{1,L} \\ \vdots & \vdots & \vdots & \vdots & & \\ \alpha^1_{j,1} & \alpha^1_{j,2} & \cdots & \alpha^1_{j,k} & \cdots & \alpha^1_{j,L} \\ \vdots & \vdots & \vdots & \vdots & & \\ \alpha^1_{N,1} & \alpha^1_{N,2} & \cdots & \alpha^1_{N,k} & \cdots & \alpha^1_{N,L} \\ \alpha^2_{1,1} & \alpha^2_{1,2} & \cdots & \alpha^2_{1,k} & \cdots & \alpha^2_{1,L} \\ \vdots & \vdots & \vdots & \vdots & & \\ \alpha^2_{j,1} & \alpha^2_{j,2} & \cdots & \alpha^2_{j,k} & \cdots & \alpha^2_{j,L} \\ \vdots & \vdots & \vdots & \vdots & & \\ \alpha^2_{N,1} & \alpha^2_{N,2} & \cdots & \alpha^2_{N,k} & \cdots & \alpha^2_{N,L} \\ \alpha^W_{1,1} & \alpha^W_{1,2} & \cdots & \alpha^W_{1,k} & \cdots & \alpha^W_{1,L} \\ \vdots & \vdots & \vdots & \vdots & & \\ \alpha^W_{j,1} & \alpha^W_{j,2} & \cdots & \alpha^W_{j,k} & \cdots & \alpha^W_{j,L} \\ \vdots & \vdots & \vdots & \vdots & & \\ \alpha^W_{N,1} & \alpha^W_{N,2} & \cdots & \alpha^W_{N,k} & \cdots & \alpha^W_{N,L} \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \\ 1 \le i \le W \\ 1 \le j \le N \\ 1 \le k \le L \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \tag{2.11}$$

Here, $\alpha^i_{j,k}$ denotes the value of the $k^{th}$ sparse component with respect to the $j^{th}$ fragment of $i^{th}$ writer. Note that the size of $\boldsymbol{\alpha}$ will be $(WN \times L)$.

The sparse principal components generated from $N \times W$ number of fragments is used to construct a histogram of $B$ bins for each of the $L$ individual components of the sparse representation

using Equation (2.12).

$$H_{kb}^{i} = \begin{cases} H_{kb}^{i} + 1, & \text{if } h_b \leq \alpha_{j,k}^{i} < h_{b+1} \\ H_{kb}^{i}, & \text{otherwise} \qquad 1 \leq b \leq B \end{cases} \tag{2.12}$$

$H_{kb}^{i}$ denotes the number of nonzero entries for the $k^{th}$ sparse component of the $i^{th}$ writer in the bin $b$ of the histogram.

Following the process of histogram generation, each of the values in the $B$ bins is normalized in the range between 0 and 1 as follows:

$$p_{kb}^{i} = \frac{H_{kb}^{i}}{\sum_{b=1}^{B} H_{kb}^{i}} \tag{2.13}$$

### 2.5.2 Computation of saliency based on entropy

The normalized histogram $p_{kb}^{i}$ is used to compute the entropy of the principal components for each enrolled writer as:

$$E_{k}^{i} = -\sum_{b=1}^{B} p_{kb}^{i} \log_2 (p_{kb}^{i}) \quad 1 \leq k \leq L \tag{2.14}$$

The notation, $E_{k}^{i}$ is the entropy value of the $k^{th}$ principal component corresponding to the $N$ fragments of the $i^{th}$ enrolled writer in the system. This entropy can be represented by a matrix:

$$\boldsymbol{E} = \begin{bmatrix} E_1^1 & E_2^1 & \cdots & E_L^1 \\ E_1^2 & E_2^2 & \cdots & E_L^2 \\ \vdots & \vdots & \ddots & \vdots \\ E_1^W & E_2^W & \cdots & E_L^W \end{bmatrix} \tag{2.15}$$

Note that in $\boldsymbol{E}$, the entries along the row specify the value of the entropy relative to each sparse principal component of the enrolled writer. Likewise, the entries along the column specify the entropy for a particular sparse component relative to each of the enrolled writers. These entropy values give us information about the probability distribution of the sparse components $\alpha_{j,k}$ corresponding to each of the $W$ writers. These individual entropy values are summed up across all writers to obtain an overall entropy value for a particular convolution feature map as

follows:

$$\hat{\theta} = \frac{\sum_{i=1}^{W} \sum_{k=1}^{L} E_k^i}{W \times L} \tag{2.16}$$

This entropy value $\hat{\theta}$ signifies the amount of information contained in a particular feature map of a convolution layer and is referred to as saliency.

Though the preceding analysis pertains to the estimation of saliency / entropy value for a particular feature map of a convolution layer , it is nonetheless worth reminding that the same approach is applicable to determining the values for all the feature maps. Without loss of generality, for a convolution layer comprising $F$ feature maps, we obtain a set of $F$ saliency values, $\hat{\theta}_f$ for $f \in \{1, 2, .., F\}$ . In order to ensure their range between $(0, 1)$ they are normalized as follows:

$$\theta_f = \frac{\hat{\theta}_f}{\sum_{f=1}^{F} \hat{\theta}_f} \tag{2.17}$$

As a visual interpretation of the above explanation, we generate four histograms corresponding to the feature maps of the first 2 layers (conv 1 and conv 2), that are assigned the top two maximum and minimum saliency values. These histograms are presented in Figure 2.7 and 2.8 respectively. For constructing each of the histograms, we have used 200 fragments selected randomly from each of the first 50 writers of the IAM database [75]. Each bin in the histograms corresponds to the entropy value of each writer that is arrived at by summing up the entries along the rows of $\boldsymbol{E}$ (defined in Equation 2.15).

## 2.6 Proposed fragment description

Given a handwritten word sample, we obtain its fragments by employing the SIFT algorithm. Each of the fragments is then passed through a convolution network to generate a set of feature maps corresponding to the number of filters included in that layer. The feature map outputs are then pooled together to generate an overall HOG representation for each fragment of the word.

The different types of pooling strategies considered in our work are as follows:

- **Average pooling**: In this methodology, equal weightage is assigned to all the feature

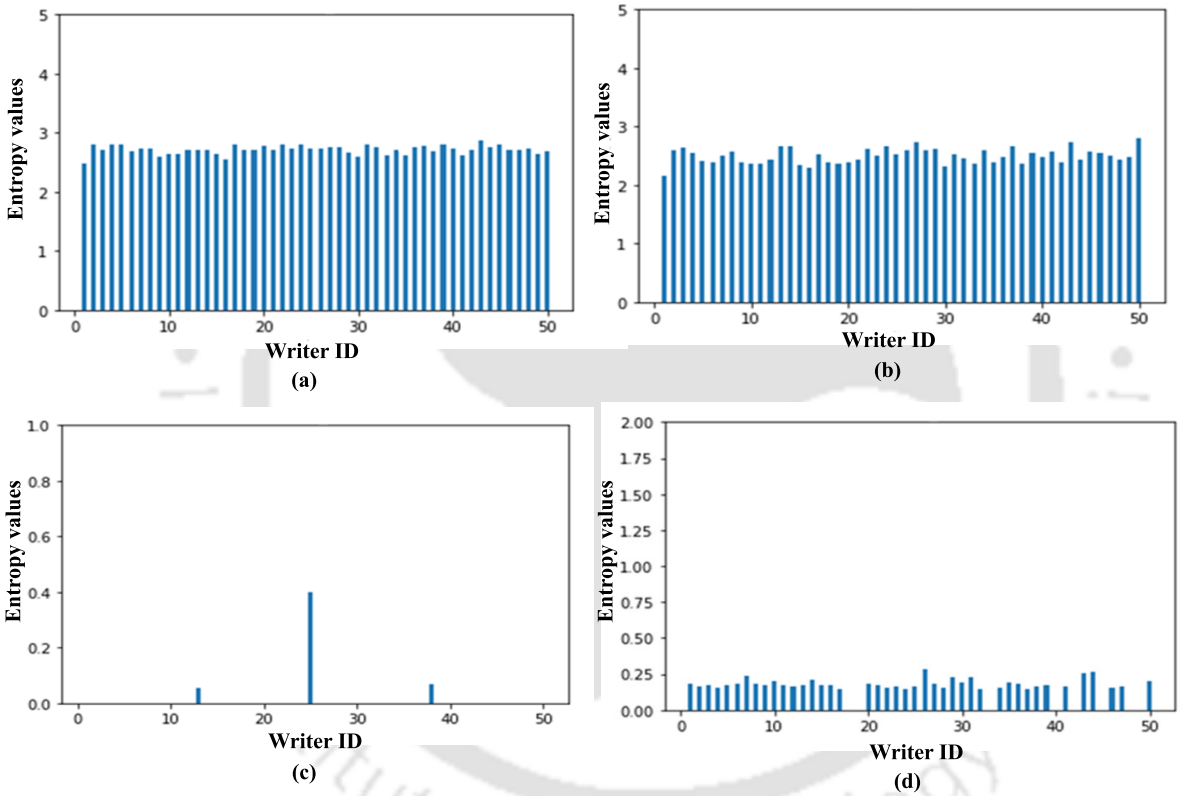



**Fig. 2.7:** (a) and (b) are the histograms corresponding to the two feature maps of the first covolution layer (conv 1) having the highest and second highest saliency values, (c) and (d) are the histogram corresponding to two feature maps of conv 1 with minimum and second minimum saliency values. These histograms are constructed using 10000 fragments corresponding to 50 writers of the IAM database. The *x*-axis represents the identity of the writers while the *y*-axis denotes their corresponding entropy values.

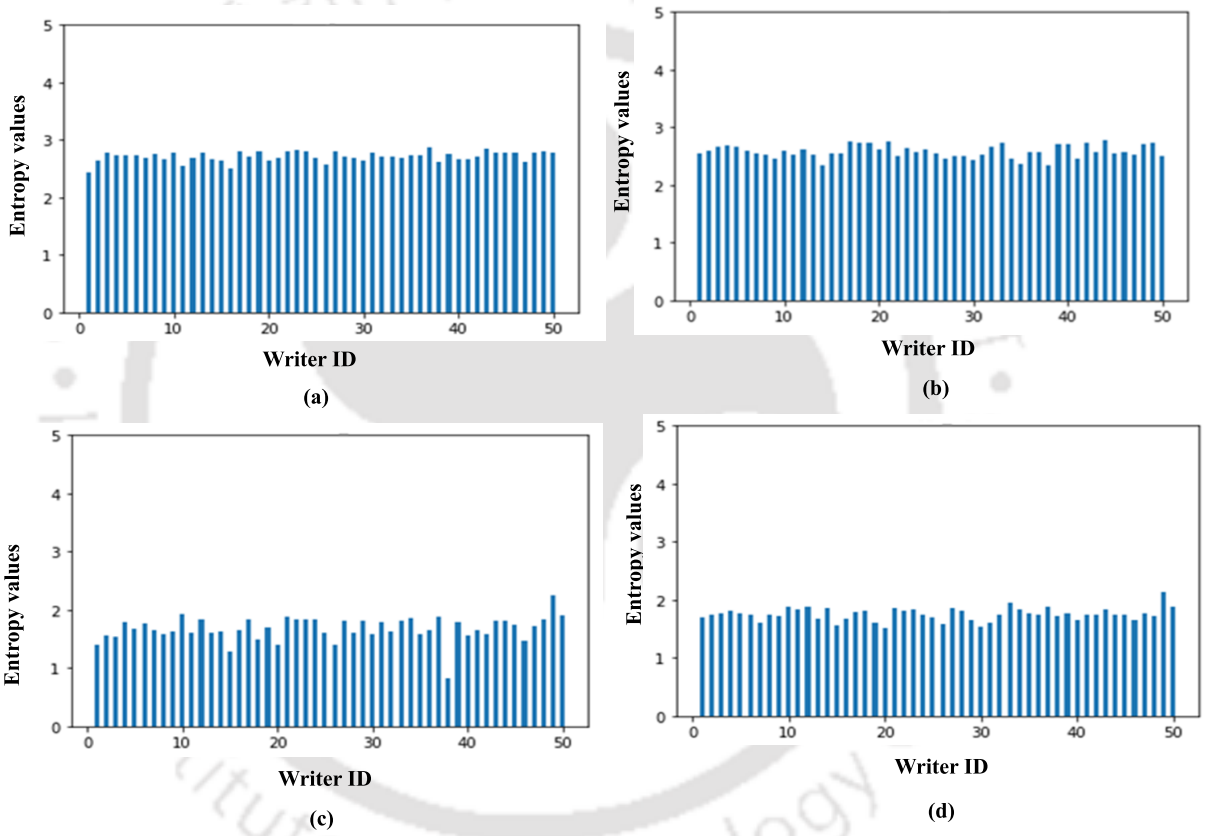



**Fig. 2.8:** (a) and (b) are the histograms corresponding to two feature maps of the second convolution layer (conv 2) having highest and second highest saliency values, (c) and (d) are the histograms corresponding to two feature maps of conv 2 with minimum and second minimum saliency values. These histograms are constructed using 10000 fragments corresponding to 50 writers of the IAM database. The *x*-axis represents the identity of the writers while the *y*-axis denotes their corresponding entropy values.

maps of a convolution layer. These are then combined together to obtain an overall single feature map. Given the $j^{th}$ fragment of writer $i$, let the feature map corresponding to the $f^{th}$ filter be represented as $z^i_{fj}$. Accordingly, the feature map ${S^i}_j$ after combination can be written as

$$ {S^i}_j = \frac{1}{F} \sum_{f=1}^{F} z^i_{fj} \tag{2.18} $$

Subsequent to obtaining the combined feature map using the average pooling strategy, they are then transformed into a one-dimensional feature vector by employing the modified HOG representation approach discussed in Section 2.4.

- **Pre-saliency pooling**: In this pooling strategy, a saliency value is assigned to the feature map of a convolution layer based on its entropy value.

$$ {\tilde{S}^i}_j = \sum_{f=1}^{F} (\theta_f \times z^i_{fj}) \tag{2.19} $$

Here, $\theta_f$ is the entropy-based saliency value assigned to each feature map $f$ of a convolution layer and ${\tilde{S}^i}_j$ represents the combined feature map output for the $j^{th}$ fragment of $i^{th}$ writer.

Subsequent to obtaining the combined feature map using the pre-saliency pooling strategy, they are then transformed into a one-dimensional feature vector by employing the modified HOG representation approach discussed in Section 2.4.

- **Post-saliency pooling**: In this pooling technique, the modified HOG feature representation is extracted individually for each feature map of a convolution layer and then subsequently combined using the corresponding saliency values. As such, in this technique, the pooling operation is performed post the incorporation of saliency values.

Mathematically, the post-pooling based saliency technique is represented as:

$$ {\hat{S}^i}_j = \frac{\sum_{f=1}^{F} (\theta_f \times \boldsymbol{h}^i_{fj})}{\| \sum_{f=1}^{F} (\theta_f \times \boldsymbol{h}^i_{fj}) \|_2} \tag{2.20} $$

Here, $\boldsymbol{h}^i_{fj}$ represents the extracted HOG feature corresponding to the $f^{th}$ feature map output from the $j^{th}$ fragment of $i^{th}$ writer, and $\theta_f$ represents the saliency value assigned to

the $f^{th}$ feature map of the convolution layer.

To summarize, each of the fragments of the word are represented by a descriptor which is derived by applying one of the three pooling techniques discussed in this Section. The resulting descriptions of the fragments are then scored using a SVM classifier to establish the identity of the writer. The effect on the identification accuracy by incorporating each of the above pooling strategies is discussed in Section 2.8.

## 2.7 Proposed two level scoring

Prior to outlining the two level scoring strategy being proposed for establishing the identity of the word image, we provide a brief overview of the classifier that will be used for the same namely, Support Vector Machine.

The SVM is a supervised binary classifier. The objective of SVM is to obtain a decision boundary that maximizes the separation between positive and negative class. Consider a training set consisting of a pair $(\mathbf{x}_i, l_i)$, $1 \leq i \leq N_{T_r}$, where $l_i$ refers to the binary label $\{-1, +1\}$ corresponding to each feature vector $x_i$. For a linearly separable case, SVM minimizes the cost function defined by:

$$J(w) = \frac{1}{2}\mathbf{w}^T\mathbf{w} \tag{2.21}$$

subjected to the following constraints:

$$l_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \qquad 1 \leq i \leq N_{T_r} \tag{2.22}$$

Here, $\mathbf{w}$ is the weight vector, and $b$ is the bias. In a more general setting a slack variable $\xi_i \geq 0$ for $i = 1, 2, ..., N_T$ is also introduced in SVM to cater to the need for testing samples, which may slightly differ from the training samples $\mathbf{x}_i$. The modified SVM cost function, including the slack variable is defined as:

$$J(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N_{T_r}} \varepsilon_i \tag{2.23}$$

subjected to:

$$l_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \varepsilon_i \tag{2.24}$$

The constant $C$ is the regularization parameter. Sometimes, the data is not linearly separable in the original space. In such scenarios, we apply a transformation technique on the input data to map it from the original space to a higher-dimensional space. This transformation is achieved using a mapping function, often referred to as a kernel function.

SVM in its original form is used primarily as a binary classifier, in order to classify multi-category data, many approaches have been proposed in literature among them two most popular approaches are: one vs. one and one vs. all classifiers. In the former, for classify, $C_1$ categories, a hyper-plane is constructed for each pair, thus resulting in $\frac{C_1(C_1-1)}{2}$ binary classifiers. During the testing phase, the testing data is voted by each classifier and the classifier with the most accumulated votes is considered the winner. Contrary to this in the one vs. rest classifier, a classifier is constructed by treating the training data belonging to a particular category as positive and remaining as negative, thus transforming the data into a binary classification problem. For $C_1$ number of classes, $C_1$ number of one vs. rest classifier needs to be constructed. During the testing phase the test data is categorized based on the category that yields the highest discriminant function value.

In this work, a set of one vs. all multi-class SVM classifiers using radial basis function (RBF) [76] is employed for training the writer descriptor. The feature vectors of a given writer are treated as a positive class, while the negative class comprises the feature vectors extracted from the rest of the writers. This ensures that the samples of the positive and negative classes are disjoint. The optimal values of RBF parameters $C$ and $\gamma$ are obtained by grid search. Each of the SVM classifiers assigns a positive or negative value to the input fragment sample based on the proximity of its feature representation to the hyperplane. We then bound these classification scores in the range between $[0 - 1]$ by passing them through a sigmoid function.

The scoring is performed at two levels, the details of which are presented in the following sub-sections.

### 2.7.1 Level 1 scoring

Given a word image (s) comprising $\hat{N}$ fragments, the description generated for each of them from the pooled feature map output of a convolution layer is passed through a set of $W$ writer-specific SVM classifiers. The resulting $\hat{N}$ scores are then accumulated with regards to a particular SVM classifier, following which a decision is made.

For sake of understanding, consider the $l^{th}$ fragment of a word (s), where $1 \leq l \leq \hat{N}$. For its HOG representation corresponding to the $j^{th}$ layer in a CNN, let $p_{il}^{j}(s)$ represent the score obtained from the $i^{th}$ writer. Accordingly, then the scoring in the level 1 scheme can be mathematically represented as

$$P_i^j(s) = \frac{1}{\hat{N}} \sum_{l=1}^{\hat{N}} p_{il}^j(s) \qquad 1 \leq i \leq W \tag{2.25}$$

Here $P_i^j(s)$ denotes the overall classification scores generated by accumulating the individual classification score for each of the word fragments obtained from the input word (s).

The final prediction is made based on the label of the writer for which the highest score is obtained.

$$\hat{y} = \underset{i \in \{1,\ldots,W\}}{arg\ max}\ P_i^j(s) \tag{2.26}$$

### 2.7.2 Level 2 scoring

Let the scores generated at the first level for the $i^{th}$ writer with respect to convolution layers 1 to $L$ be denoted by $P_i^1(s), P_i^2(s), ..., P_i^L(s)$ respectively. These are subsequently combined together to get a final score $D^i(s)$ as follows:

$$D^i(s) = \sum_{j=1}^{L} \lambda_j P_i^j(s) \tag{2.27}$$

Here, $0 \leq \lambda_j \leq 1$ is a positive weight factor such that $\sum_{j=1}^{L} \lambda_j = 1$. Its values are determined by performing cross-validating on the training data. The writer identity of the word image (s) is now predicted as follows

$$i^* = \underset{i}{arg\ max}\ D^i(s) \tag{2.28}$$

## 2.8 Dataset description and pre-processing

The proposed method is evaluated on three datasets, namely IAM [75], CVL [77] and CERUG-EN [78].

The IAM [75] dataset is the most widely used English language dataset for the purpose of writer identification. It contains 1593 images of English handwriting documents collected from a set of 657 writers. Each writer has contributed a variable number of handwritten documents. Out of 657 writers, 301 have provided two or more handwritten documents, while the rest have contributed only a single handwritten document.

We modified the IAM dataset as described in [22]. For writers contributing more than one page of a handwritten document, we randomly select two pages, of which one page is used for training and the other for testing. For writers with only one page of a handwritten document, we split it roughly into two halves. One half is used for training and the other half for testing. We utilize the bounding box information for the word images provided in the dataset to generate the training and test samples.

The CVL [77] dataset contains handwritten text documents from 310 writers of which 27 writers have contributed 7 text documents, while the rest have written 5 text documents. Each writer contributed one text document in German and the rest in the English language. In our experimentation, we have only utilized the text document written in English. We follow the same methodology as is done in [58], by selecting three documents per writer for training and the fourth for testing. Similar to the IAM dataset, segmented word images are made available in this dataset.

The CERUG-EN [78] contains handwritten documents collected from 105 subjects with each subject contributing two paragraphs in English. In our experiment, we utilize the first paragraph contributed by each subject for training and the second paragraph for testing. Similar to the IAM and CVL databases the segmented word images are made available in the database by the authors.

Table 2.1 gives a detailed overview of all the datasets used for our experiments.

**Table 2.1:** Overview of the datasets used for the experiments

| Dataset | Number of Writers | Language | Words | | Fragments | |
|---|---|---|---|---|---|---|
| | | | Training | Testing | Training | Testing |
| CVL | 310 | English | 59724 | 33090 | 1521804 | 805238 |
| IAM | 657 | English | 32449 | 28653 | 2130995 | 1848155 |
| CERUG-EN | 105 | English | 5702 | 5127 | 360552 | 305257 |


**Fig. 2.9:** Illustration of pre-processing operation. (a) Raw input word image from the IAM data base and (b) preprocessed image output.

### 2.8.1 Pre-processing

With regards to the datasets, the details regarding the steps associated with obtaining the segmented word image from the handwritten document are provided in [75, 77]. These include a set of operations such as skew-correction and text-line segmentation, removal of ink blots and extraneous marks. Nonetheless, in order to address the variation in the background due to lighting conditions in the segmented word images of the IAM and CERUG-EN database, we perform a two-stage operation on the grayscale images. The image is first filtered using $(5 \times 5)$ Gaussian kernel following which Otsu thresholding is applied. The resulting filtered word image thus generated is shown in Figure 2.9. Finally, as a last pre-processing operation all the segmented images of the three databases are normalized in the range between [0-1] using min-max normalization operation.

## 2.9 Results and discussion

In this Section, we enumerate a set of experiments to demonstrate the efficacy of our proposed algorithm on the aforementioned three datasets. Each set of experiments has been conducted for ten trials and the obtained average writer identification rate has been reported. In the

context of this work, each trial refers to the following set of steps namely, the selection of word images for training and testing, fragment generation, HOG feature description, determination of Sparse PCA components, computation of saliency values of feature maps of a convolution layer and final writer classification based on the descriptors of its constituent fragments, as obtained from a pooling strategy.

### 2.9.1 Training and implementation details

The proposed convolution network is built using the TensorFlow framework. The Adam optimizer is used for optimizing the network, with a weight decay factor of 0.1 after every ten epochs. The model is trained for 50 epochs. The pixel intensities of the fragments are normalized (between $[0 - 1]$) before feeding to the convolution network for feature extraction.

For extracting image fragments using SIFT algorithm, the input word image is divided into three octaves, with the image in each octave being half of the previous one. The base image is gradually blurred over the course of an octave using a series of Gaussian kernels, each with a mean of 0 and variance of 1.6, to produce a set of five images of various scales. This set of Gaussian-filtered images is used to generate a set of 4 Difference of Gaussian (DoG) images per octave. Each pixel of this DoG image is compared with its 8 neighbours in the current image as well as 9 neighbours in the level above and below (26 total) to check for the candidate key points (by analyzing if the pixel is local extremum). In order to remove low-contrast key points the value of the Hessian determinant corresponding to all candidate keypoint is compared against the threshold value of 0.04. The scale information of the remaining stable key points is used to generate a set of word fragments corresponding to each octave.

In order to extract the HOG representation [1]. corresponding to the features maps of convolution layers 1 and 2, we select the value of $(m, n)$ to be $(4, 4)$. As a result, each of the conv 1 and conv 2 layer feature maps are divided into 16 image patches. These 16 patches are grouped into $b = 4$ blocks, with each block containing $t = 4$ sub-images.

In a convolution network, the size of the feature map decreases with increasing depth. In

[1]For the discussion of HOG, refer Section 2.4

**Table 2.2:** Table showing the effect of bin size ($B$) on the overall dimension of the HOG feature vector for a given value of $m, n, t$ and $b$.

| Convolution layer | m,n,t,b | Bin Size (B) | HOG feature vector dimension (B*t*b) |
|---|---|---|---|
| **Conv1** | 4,4,4,4 | 6 | 96 |
| | | 8 | 128 |
| | | 10 | 160 |
| | | 12 | 192 |
| **Conv2** | 4,4,4,4 | 6 | 96 |
| | | 8 | 128 |
| | | 10 | 160 |
| | | 12 | 192 |
| **Conv3** | 2,2,2,2 | 6 | 24 |
| | | 8 | 32 |
| | | 10 | 40 |
| | | 12 | 48 |

order to maintain the spatial consistency between the various filtered outputs, the value of $m$ and $n$ is adjusted based on the size of the feature maps of a convolution layer. Keeping this in consideration, we select the value of $(m, n)$ to $(2, 2)$ in conv 3 feature map (refer Figure 2.5). This is owing to the fact that the size of the conv 3 layer is half the size of the preceding conv 2 layer. Likewise, the value of $t$ and $b$ is also readjusted accordingly to $(2, 2)$. The computation of the HOG depends on the size of the feature map in each convolution layer as well as the number of histogram bins used for representation. In Table 2.2 we tabulate the dimension of the resulting feature vector for the selected parameters of $m, n, t$ and $b$ with different bin sizes $B$ corresponding to convolution layers 1,2 and 3.

While deciding the writer identity for a particular word image, the fragments being fed to the pre-trained CNN are not fixed in number - rather they depend on the number and aspect ratio of their individual characters. As an illustration, we show in Figure 2.10 samples of word images with varying number of characters selected from IAM and CVL writer databases respectively with the number of generated fragments in Table 2.3. Clearly, the number of fragments is not the same across the words.

(a)

(b)

**Fig. 2.10:** Samples of word images taken from different image databases. Sub-figure (a) corresponds to the word samples taken from IAM database, and sub-figure (b) corresponds to word samples taken from CVL database respectively.

**Table 2.3:** Number of fragments generated for the word image samples in Figure 2.10.

| Database | Word | Number of word Fragments |
|---|---|---|
| IAM | for | 32 |
| | would | 99 |
| | adequate | 154 |
| CVL | was | 63 |
| | differ | 122 |
| | shouting | 136 |

### 2.9.2 Performance of average pooling strategies with different HOG feature representation

In this section, we examine the impact of extracting HOG features from convolutional feature maps and compare its performance with the feature vector obtained by flattening the convolutional feature map output. Since SIFT keypoints vary in size, the convolutional output cannot be flattened into a consistent, fixed-dimensional feature representation. As, such, the SIFT keypoints are resized to $50 \times 50$, based on the average dimension of the extracted keypoints, before being fed into the convolutional network. It is important to note that this resizing is performed solely to facilitate uniform dimension of feature representation obtained through flattened con-

volutional feature maps and does not apply to HOG-based representations. Furthermore, both the HOG-based and flattened feature map representations is evaluated for each convolutional layer without incorporating saliency values. In other words, we employ the average pooling formulation defined in Equation 2.18 and use the SVM classifier scores at level 1 for scoring. For this experiment alone, we consider a subset of writers corresponding to one-third of the database for CVL and IAM (100 for CVL, 200 for IAM and 35 for CERUG-EN database). To obtain the HOG-based feature representation, we vary the value of bin size from 6 to 12 in steps of two. Table 2.4 presents the result of our experiment. We observe that the highest identification rates for the Conv1 layer using HOG feature representation are 91.15%, 82.97%, and 77.71% for the IAM, CVL, and CERUG-EN databases, respectively, with an optimal bin size of 10. In contrast, the identification rates using flattened convolutional feature maps are 84.25%, 79.67%, and 62.13% for the same databases. For the Conv2 layer, the best identification rates with HOG features across the IAM, CVL, and CERUG-EN databases are 90.25%, 80.95%, and 74.42%, again with a bin size of 10. Meanwhile, the flattened feature map representation yields identification rates of 83.65%, 78.87%, and 60.58% for these datasets.

When we consider the feature maps of the conv 3 layer, the identification rate decreases to a value of 81.88%, 67.15%, and 61% respectively for the three databases. The same trend is also observed with the higher convolution layers conv4, conv5 and conv6.

Based on the above-tabulated results, it can be inferred that the initial layers of the convolution neural network (namely conv 1 and conv 2) capture most of the information that is helpful in differentiating one set of writers from another. As we move deeper into the convolution network, the resolution of the image decreases, with the underlying information becoming more abstract and less visually interpretative as shown in Figure 2.11. This results in the loss of discriminability among writers (as depicted by the result of conv 3).

The results in Table 2.4 further demonstrate that the HOG-based representation outperforms the flattened feature map representation. This may be because the flattened representation relies on resizing the original SIFT keypoints, which can reduce their quality and diminish their discriminative power. Additionally, HOG (Histogram of Oriented Gradients) captures local

**Table 2.4:** Comparison of average identification rates (in % ) for the word level data with different bin sizes for HOG representation. The average pooling strategy is used for this experiment. The best identification rate is marked in bold. – in the bin size indicates the result obtained by flattening the feature map output of the convolution map.

| Convolution layer | Bin size | IAM | CVL | CERUG-EN |
|---|---|---|---|---|
| conv 1 | - | 84.25 | 79.67 | 62.13 |
| | 6 | 89.8 | 81 | 71.14 |
| | 8 | 90.15 | 82 | 74.57 |
| | 10 | **91.15** | **82.97** | **77.71** |
| | 12 | 89.85 | 81.21 | 76.71 |
| conv 2 | - | 83.65 | 78.87 | 60.58 |
| | 6 | 89.4 | 79 | 70.42 |
| | 8 | 89.8 | 79.95 | 71.85 |
| | 10 | **90.25** | **80.95** | **74.42** |
| | 12 | 88.55 | 80.80 | 72.85 |
| conv 3 | 6 | 70.00 | 47 | 53.14 |
| | 8 | 73.35 | 57.8 | 55.85 |
| | 10 | 79.97 | 64.45 | 59.85 |
| | 12 | 81.88 | 67.15 | 61.00 |

patterns by encoding gradient directions and magnitudes, preserving the spatial arrangement of edges and textures. In contrast, flattening convolutional feature maps eliminates spatial relationships, treating all pixel values as independent features.

Based on the observed trend, the subsequent experiments adopt the HOG feature representation for writer features, with the bin size set to 10. Moreover, we consider the incorporation of saliency values of the feature maps in a convolution layer for obtaining the writer descriptor. The evaluation is performed on the entire writer database by considering the conv 1 and conv 2 layers.

### 2.9.3 Influence of saliency based pooling strategy for writer descriptor

In this experiment, we consider the effect of incorporating the different pooling strategies employed for obtaining the feature descriptor of the writer. These methods namely: average pooling, pre-saliency pooling and post-saliency pooling (refer Equations 2.18 - 2.20) have been discussed in Section 2.6.

The performance of our methodology on the individual conv 1 and conv 2 layers (using the

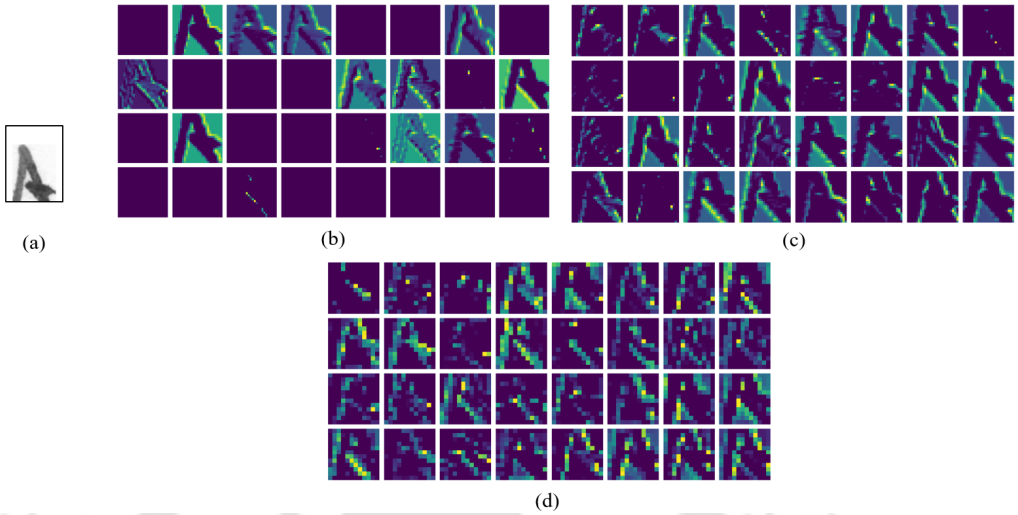



**Fig. 2.11:** Illustration of feature map outputs of an image fragment (sub-figure (a)) as obtained from the convolution layers 1, 2 and 3 (in sub-figures (b), (c) and (d)).

SVM scores at level 1) are tabulated in Table 2.5 for the three databases. Based on the entries, it can be inferred that the performance of the average pooling methodology is lower in contrast to the other two pooling strategies across both layers. This is owing to the fact that the former assigns equal importance to all the feature maps of a convolution layer during pooling. This in turn suggests that the resulting output takes into consideration feature maps having little or no information as well. Contrary to this strategy, by incorporating the idea of saliency scores in the pooling process, the contribution of the feature maps is adjusted based on their values, thereby increasing the performance of the system.

Further, amongst the two saliency-based pooling strategies, the post-saliency pooling gives a slightly improved result. This observation can be attributed to the fact that the pre-saliency-based pooling approach performs a combination of the feature maps prior to HOG feature representation. This, at times, may cause the feature maps having a lower saliency score to get masked by those with relatively high values. Contrary to this, in the post-saliency-based pooling approach, the HOG feature representation corresponding to each of the feature maps is weighted with the saliency values and this in a way, may aid in preserving the contributions made by the individual feature map in the overall representation of writers.

To demonstrate the effectiveness of sparse PCA-based feature representation over conventional PCA, we compute the saliency score of the convolutional map using both approaches.

**Table 2.5:** Average accuracy achieved (in %) on IAM, CVL and CERUG-EN dataset using various pooling strategies employed for generating the writer descriptor. The results are presented for the conv 1 and conv 2 layers respectively by scoring the SVMs at the first level.

| Database | layer | Average pooling | Pre pooling | Post pooling |
|---|---|---|---|---|
| IAM | conv 1 | 84.31 | 85.02 | 86.07 |
| | conv 2 | 83.42 | 84.62 | 85.81 |
| CVL | conv 1 | 77.53 | 78.05 | 79.21 |
| | conv 2 | 75.29 | 76.54 | 77.79 |
| CERUG-EN | conv 1 | 72.19 | 72.57 | 73.33 |
| | conv 2 | 69.90 | 70.19 | 72.19 |

**Table 2.6:** Average accuracy achieved (in %) on IAM, CVL and CERUG-EN dataset using pre-pooling strategies employed for generating the writer descriptor. The results are presented for the conv 1 and conv 2 layers respectively, by scoring the SVMs at the first level. The saliency weights are generated using PCA and Sparse PCA-based methods respectively.

| Database | layer | PCA | Sparse PCA |
|---|---|---|---|
| IAM | conv 1 | 84.87 | 85.02 |
| | conv 2 | 83.85 | 84.62 |
| CVL | conv 1 | 77.87 | 78.05 |
| | conv 2 | 76.15 | 76.54 |
| CERUG-EN | conv 1 | 72.28 | 72.57 |
| | conv 2 | 70.08 | 70.19 |

The resulting saliency scores are combined using a pre-saliency pooling strategy to generate the writer's feature descriptor. The result of this analysis is presented in Table 2.6, which shows that saliency scores obtained through Sparse PCA outperform those from conventional PCA. This improvement is likely due to Sparse PCA's ability to zero out the coefficients of less significant features, retaining only the most informative ones. As a result, it reduces the influence of noise and irrelevant information, enhancing the quality of the principal components.

### 2.9.4 Statistical Significance

In this sub-section, we demonstrate that the results of the saliency-based pooling strategy is statistically significant when compared to the conventional average-based pooling strategy. The analysis has been performed using Student's $t-$test for a significance level of 0.05. Table 2.7 outlines the $p-$values obtained for the feature vectors extracted from convolution layers 1 and 2 across the three databases. A trend of low value of $p$ is observed for each entry in the Table

**Table 2.7:** Statistical significance test on the performance of the proposed modified HOG based feature descriptor by incorporating saliency values at different levels over the average pooling method via the Student's $t-$test

| Database | Conv 1 layer | | Conv 2 layer | |
|---|---|---|---|---|
| | Pre-saliency pooling | Post-saliency pooling | Pre-saliency pooling | Post-saliency pooling |
| CVL | $4.3\times10^{-2}$ | $7.9\times10^{-3}$ | $2.8\times10^{-3}$ | $9.1\times10^{-7}$ |
| IAM | $4.5\times10^{-2}$ | $2.8\times10^{-2}$ | $2.4\times10^{-3}$ | $9.1\times10^{-4}$ |
| CERUG-EN | $4.8\times10^{-2}$ | $3.3\times10^{-2}$ | $1.8\times10^{-3}$ | $8.6\times10^{-5}$ |

**Table 2.8:** Comparison of average accuracy achieved (in %) using individual convolution layer and the proposed fusion approach.

| Pooling strategy | IAM | | | CVL | | | CERUG-EN | | |
|---|---|---|---|---|---|---|---|---|---|
| | conv1 | conv2 | combined | conv1 | conv2 | combined | conv1 | conv2 | combined |
| Average pooling | 84.31 | 83.42 | 86.07 | 77.53 | 75.29 | 79.70 | 72.19 | 69.90 | 73.66 |
| Pre-pooling | 85.02 | 84.62 | 86.76 | 78.05 | 76.54 | 80.16 | 72.57 | 70.19 | 74.47 |
| Post-pooling | 86.07 | 85.81 | 87.68 | 79.21 | 77.79 | 82.10 | 73.33 | 72.19 | 75.80 |

across each dataset thereby indicating the statistical significance of our proposal.

### 2.9.5 Evaluation of scoring in Level 2

In this Section, the effectiveness of incorporating the overall proposed SVM score fusion at level 2 (refer Section 2.7.2) is demonstrated. The result of this experiment is tabulated in Table 2.8. From the entries, it can be observed that the performance with level 2 scoring is better in comparison to the scores provided by level 1 for the convolution layers 1 and 2. Moreover, the improvement is evident across all three pooling strategies. This shows the proposed two-level fusion step exposes the complementary power of the convolution layers in identifying a writer. That being said, it is worth mentioning here that we could have as well considered combining other convolution layers. In a way, our choice of restricting to convolution layers 1 and 2 is due to their higher individual writer identification rates (using the scores from level 1) as compared to the deeper layers.

### 2.9.6 Performance of proposed system with word images of different lengths

Figure 2.12 shows the performance of the overall proposed system when tested on word images of different lengths taken from the IAM and CVL datasets respectively. It can be inferred

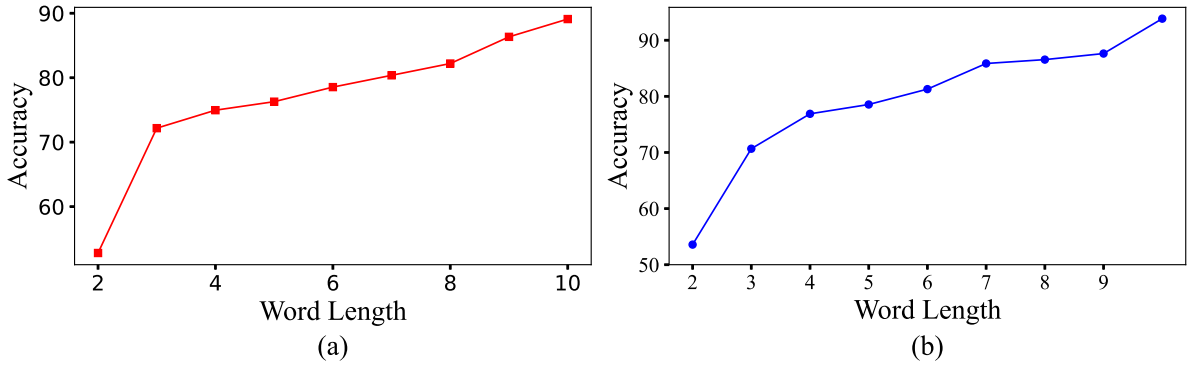



**Fig. 2.12:** Average performance of writer identification with different word lengths on the IAM and CVL data set.

**Table 2.9:** Comparison of our proposal to prior works

| Method | IAM | CVL | CERUG-EN |
|---|---|---|---|
| Hinge [14] | 13.8 | 13.6 | 14.4 |
| Quill [79] | 23.8 | 23.8 | 24.5 |
| CoHinge [80] | 19.4 | 18.2 | 17.7 |
| QuadHinge [80] | 20.9 | 17.8 | 17.0 |
| COLD [81] | 12.3 | 12.4 | 17.3 |
| Chain Code Pairs [20] | 12.4 | 13.5 | 14.5 |
| Chain Code Triplets [20] | 16.9 | 17.2 | 17.8 |
| WordImgNet [58] | 52.4 | 62.5 | 74.3 |
| FragNet-64 [58] | 72.2 | 79.2 | 75.9 |
| Vertical GR-RNN(FGRR) [59] | 83.3 | 83.5 | 70.2 |
| Horizontal GR-RNN(FGRR) [59] | 82.4 | 82.9 | 68.9 |
| Proposed Methodology | 87.68 | 82.10 | 75.80 |

that for word lengths less than 4, accuracy is quite less, hovering around 75%. Nevertheless, with an increase in the number of characters in the input word image, an improvement to as high as 20% is perceived. This trend suggests that with more number of characters in the word image, the overall number of fragments also goes up in proportion, thereby capturing the writers' style information more effectively.

### 2.9.7 Comparison to prior works

In this sub-section, we evaluate the performance of our proposed approach with traditional handcrafted as well as recent deep-learning-based systems built for identifying the authorship of handwritten word images.

From the entries in Table 2.9, it can be observed that the identification accuracy of traditional

handcrafted features are low. This is because these algorithms make predictions based on the statistical data collected from the input image, for which a certain number of text samples are required. Nonetheless, this information is insufficient to generate a stable representation of the input text sample.

Contrast to hand crafted features, the presence of multiple feature maps in each layer of a convolution network enables it to capture a diverse quantity of information from the word-level training data, thus contributing to improved performance. As a matter of fact, with regards to the recent deep learning methods, our proposed algorithm provides an identification rate that is at par with the performance of the features learned in [58, 59].

It may be noted that the systems being enumerated in Table 2.9 can differ with regards to the feature set as well as the classification, and enrollment strategies. Therefore, it should be borne in mind that a direct one-to-one comparison with these approaches may not be fair.

## 2.10 Computational complexity

For the sake of completeness, we also provide the average execution time for the proposed modified sparse-based representation framework based on conducting multiple trials on the word sample of the IAM database. The analysis was done on an HP desktop with 16GB RAM and an Intel i7 processor. Table 2.10 tabulates the average execution time corresponding to the set of operations carried out at each stage:

- Pre-processing: The operation carried out in this step corresponds to the removal of background variation from the input word image and fragment generation using the SIFT algorithm.
- Feature extraction: In this stage, we measure the average time required to obtain the feature map representations in the convolution layers. These are derived from the input fragments of the word image.
- Saliency score generation: The average time spent in this stage corresponds to the set of operations mentioned in Section 2.5 namely (a) sparse PCA generation, and (b) Entropy-based saliency estimation.

**Table 2.10:** Average time complexity (in seconds) of the proposed framework corresponding to the IAM database.

| Description | Execution time in second |
|---|---|
| Pre-processing (per word) | 3.06 |
| Feature extraction (per word/filter) | 3.29 |
| codebook generation(per filter) | 79.18 |
| Saliency score estimation(per filter) | 89.09 |
| SVM training (per writer) | 55.45 |
| SVM testing (per word) | 0.047 |

- Modified fragment representation: This step consists of obtaining the modified representation for each fragment using a post-saliency-based pooling strategy.
- SVM training: It represents the average time lapse that occurs in training the SVM classifiers at the fragment level in one vs all framework.
- SVM testing: It denotes the average time involved in acquiring the final score that will be used to decide on the writer identity of the handwritten word. This is achieved by aggregating the individual scores for each of the fragments associated with a particular word. This process entails the use of an individual SVM classifier for representing a specific writer.

## 2.11 Conclusion

In this Chapter, we have proposed an offline test-independent writer identification system based on word images using a trained CNN network. The main focus of this study is to exploit the importance of feature maps in a convolution layer by assigning saliency values. The proposed approach leverages convolutional layer feature maps to generate a feature vector for writer identification. An entropy-based metric assigns saliency scores to measure each channel's informativeness, and these scores are then used to compute a weighted sum, prioritizing relevant features. The method refines decision boundaries by selecting informative features, enhances feature interactions, and improves overall model performance.

# 3

# Exploration of Siamese network representation in a reduced subspace

## Contents

## 3.1 Introduction

In the previous chapter, we explored a classification-based framework for feature representation in the context of writer identification. Although convolutional classification models have achieved success in various applications, such as image classification [82], object detection [83], and text categorization [84], they are highly dependent on large, labelled datasets. However, in real-world scenarios, such datasets are often limited due to technical, ethical, or privacy constraints, which can impair the generalization ability of these models and result in inconsistent performance across different classes. This reliance on extensive labelled data per class presents a significant challenge for traditional classification approaches, as it is essential for learning accurate decision boundaries.

To address this limitation, in this chapter, we introduce a dissimilarity-based method for predicting the author identity of handwritten word images using a variant of deep neural networks known as the Siamese architecture. Unlike traditional classification methods that depend on predefined classes, this approach focuses on comparing pairs of handwriting samples to determine their similarity or dissimilarity. It learns a metric space in which fragments from the same writer are mapped closer together, while those from different writers are placed further apart, enabling strong generalization to unseen writer samples. This makes the method particularly effective for open-set scenarios, where new or unknown writers frequently appear. While classification-based models perform well in closed-set settings, dissimilarity-based frameworks offer greater adaptability and robustness for real-world writer identification tasks. Although Siamese networks have been widely applied in multi-class classification [85], signature verification [86], and handwritten character recognition [87]. This makes them a promising approach for addressing the limitations of traditional classification models.

As a first contribution, the present work explores the idea of similarity learning by analysing the relationship between various patterns (referred to as fragments) created by a writer in a text-independent framework. The fragments are generated by applying the SIFT algorithm [1], that uses a multi-scale analysis on the word image to locate key-point regions ranging from

characters to allographs and graphemes. The generated fragments are then subsequently passed through a pre-trained Siamese network. The pre-training is done by using the hand-written samples of the Omniglot dataset [88]. This dataset contains characters spread across a variety of alphabets in different languages, thereby ensuring better generalization of the features of a writer.

Corresponding to each set of fragments extracted from a word image, a fixed-sized feature representation is generated based on the penultimate layer output of the trained network. However, the dimensionality of the obtained feature representation is quite high and a need thus arises to reduce the same with the view of alleviating the effects of the curse of dimensionality. To this, we propose to utilize the sparse PCA from the previous Chapter as a tool to obtain the lower dimensional representation of the Siamese network features.

Most sparse representation-based approaches assume that all sparse components (such as sparse principal components in our case) are equally informative for classification, relying on the distribution of these coefficients to make decisions. In contrast, this study aims to enhance the discriminative power of each sparse component by introducing a divergence-based saliency score derived from their histogram-based representation during training to extract latent information within the data. A divergence-based metric measures the disparity between probability distributions, offering a way to assess the saliency of each component.

In the identification phase, each of the sparse components are modified in accordance with their estimated saliency score (using an adaptive power law transformation) to generate a modified writer descriptor for each fragment of the word image. This descriptor is then passed through a set of writer-specific SVM classifiers, providing a score for each fragment. These scores are subsequently accumulated across all fragments of a given word image with regard to each enrolled writer-specific classifier. The identity of the word sample is determined based on the maximum value of scores represented across the writer-specific SVM classifiers.

The main contribution of this work can be summarized as follows:

(i) Applying the concept of the Siamese network for feature representation in a writer-independent framework.

(ii) Exploring a sparse-based model for representing the output feature vector of the Siamese output in reduced dimensional space.

(iii) Proposing an approach for assigning saliency score to each component in the sparse PCA representation, thereby enhancing their discriminative ability.

## 3.2 Block schematic of our proposal

Figure 3.1 presents the overview of the proposed algorithm. An input word image is first fed to a SIFT key-point detector for generating the fragments. These image fragments correspond to parts of characters that comprise a word. These are then individually fed to a trained Siamese network to obtain a set of feature vectors from the penultimate layer (Figure 3.1 (a)). The resulting feature vectors are then encoded by projecting them over a set of principal directions constructed using the sparse PCA framework. Each resulting sparse PCA coefficient is assigned a value (referred to as 'saliency') signifying its effectiveness in discriminating individual writers, using which a modified writer descriptor is constructed (Figure 3.1 (b)) . Finally, the modified descriptor generated for each image fragment is fed to a trained SVM classifier to generate a classification score relative to each enrolled writer, which is then averaged across the fragments to establish the identity of the writer.

The rest of the Chapter is organized as follows. We begin by describing the process of fragment generation using SIFT algorithm along with a detailed discussion about the Siamese-based CNN architecture, whose penultimate layer is used for feature extraction in Section 3.3. In Section 3.4, we provide the details of the Sparse PCA technique for encoding the extracted features. The methodology employed for assigning a saliency score to each sparse principal component is discussed in Section 3.5. In Section 3.6 the modified writer descriptor generated by incorporating the saliency score is elaborated. In Section 3.7 we describe the implementation details along with the experimental results to evaluate the efficacy of our proposed system. Finally, we summarize our work in Section 3.8.

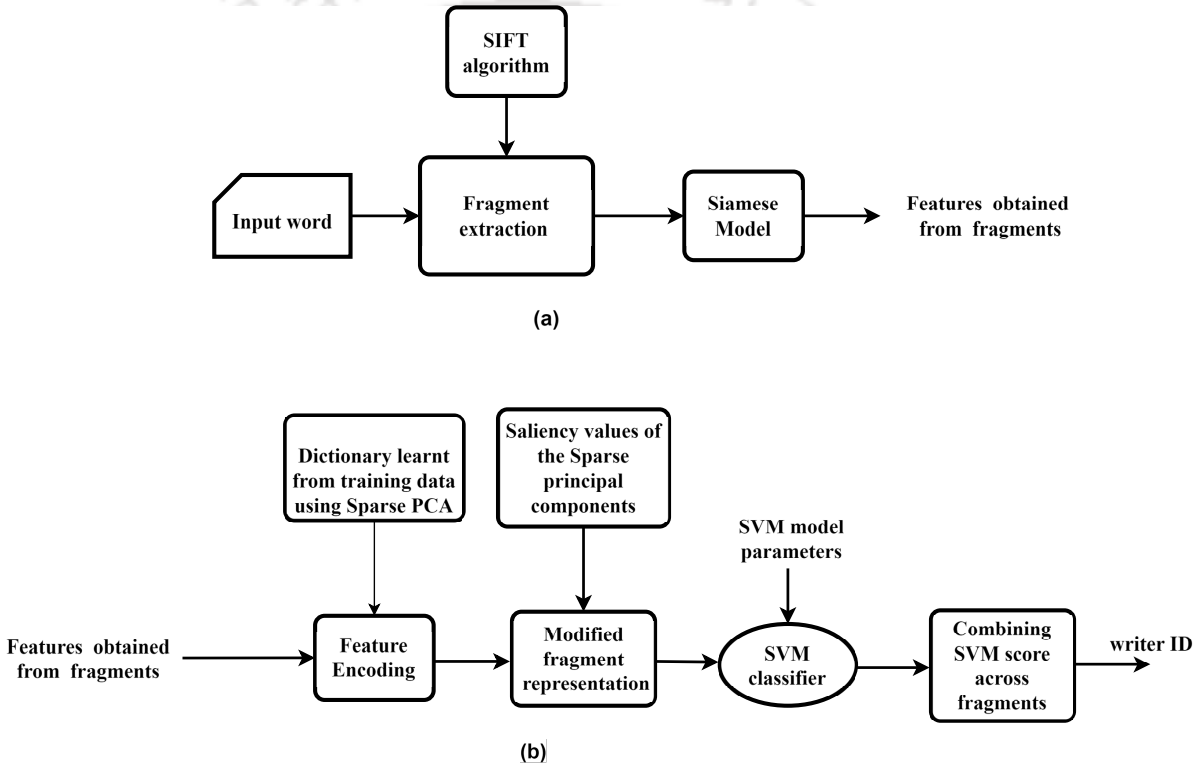



**Fig. 3.1:** Pictorial overview of our proposed system. In sub-figure (a), we depict the feature extraction step obtained from the trained Siamese network with respect to the fragments generated from the input word. Likewise, sub-figure (b) shows the steps involved in obtaining the writer ID by utilizing the representation of the Siamese network outputs in the reduced sub-space, that is constructed using Sparse PCA.

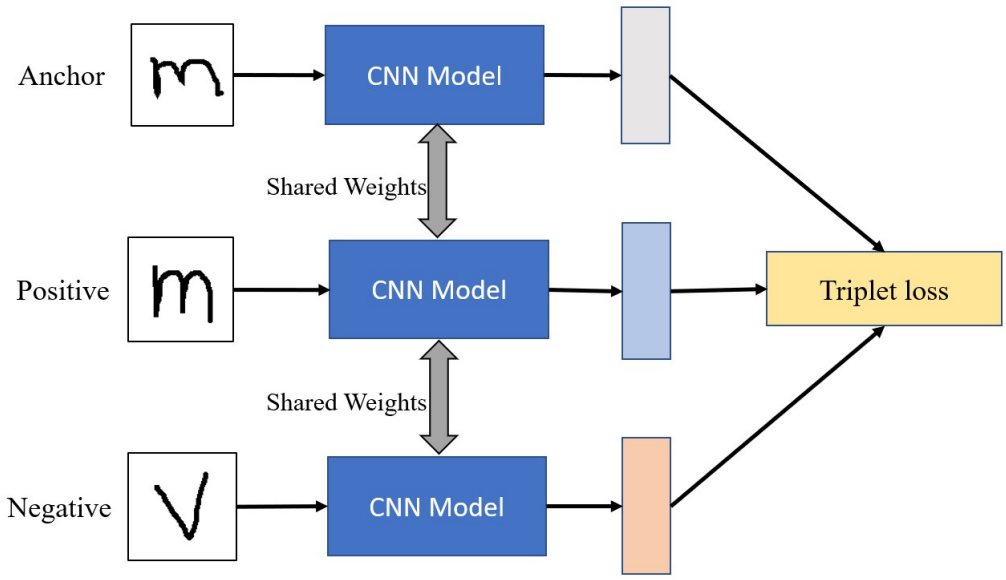



**Fig. 3.2:** Siamese network with triplet loss trained on samples of the Omniglot dataset.

## 3.3 Siamese network architecture

We begin by performing a sequence of operation on input word image for obtaining fragments using SIFT algorithm [1] as described in Section 2.2. Once these are extracted, they need to be mapped into features for subsequent analysis by considering a class of neural networks called the Siamese network. The Siamese network consists of a combination of two or more identical sub-networks each having a separate input that is interconnected by a loss function based on which the weights of the network is updated. These sub-network weights are shared, which aids in ensuring that fragments with comparable content are mapped to the same region in a feature space.

In our application we have used triplet loss for adjusting the weights of the network (as shown in Figure 3.2). Triplet loss is a metric learning method that enables a model to distinguish between similar and dissimilar samples by learning embedding for each input. The goal of this loss function is to ensure that embedding of similar inputs are closer together, while those of dissimilar inputs are pushed farther apart. This is accomplished by forming triplets, each containing an anchor input, a positive input (which is similar to the anchor), and a negative input (which is dissimilar to the anchor). The triplet computes the squared Euclidean distance between the anchor and positive embedding, then subtracts it from the squared Euclidean distance

between the anchor and negative embedding. This difference is compared against a predefined margin. If the difference is smaller than the margin, the loss is set to zero; otherwise, the loss is equal to the computed difference. The margin hyper-parameter dictates how far apart the embedding of the anchor and negative inputs should be, ensuring a distinct separation between them. The triplet loss can be mathematically expressed as:

$$\mathcal{L}(A, P, N) = \max\left(\|f(A) - f(P)\|_2^2 - \|f(A) - f(N)\|_2^2 + \alpha,\ 0\right) \tag{3.1}$$

Here $f(A)$ is the embedding (feature vector) for the anchor, $f(P)$ is the embedding for the positive sample, and $f(N)$ is the embedding for the negative sample. $\alpha$ is a hyper-parameter that specifies the minimum margin to be maintained between the distance of the anchor-positive pair and the anchor-negative pair.

The block diagram of the CNN module used in the proposed Siamese network is shown in Figure 3.3. The module consists of ten convolution layers with skip connections being added to some of these convolution blocks transforming them into the residual block as shown in Figure 3.3.(b). Each convolution layer is assigned a set of filters in the range from 64 to 512 having a kernel size of $(n \times n)$, where $n \in (1, 3, 7)$. A batch normalization layer and a Rectified Linear Unit (ReLU) follow each convolution layer. The convolution block is finally terminated using a fully connected layer. Our model uses a configuration in which the fully connected layer is preceded by a Global average pooling layer instead of a flattened version of the convolution layer. This is done to reduce the overall number of trainable parameters and preventing over-fitting. The weight of the proposed Siamese network is updated using the triplet loss [89].

Our network is pre-trained using the writer-independent Omniglot dataset [88] containing 1623 different handwritten characters each of size $(105 \times 105)$ collected from 50 different alphabets. The characters correspond to alphabets collected from a diverse set of languages, thus ensuring the generalization of writer features. A set of 40 random alphabets are used for training and the rest for evaluation. Some of the sample images of this dataset are shown in Figure 3.4.

It is worth emphasizing that once the Siamese architecture network is trained on the Om-

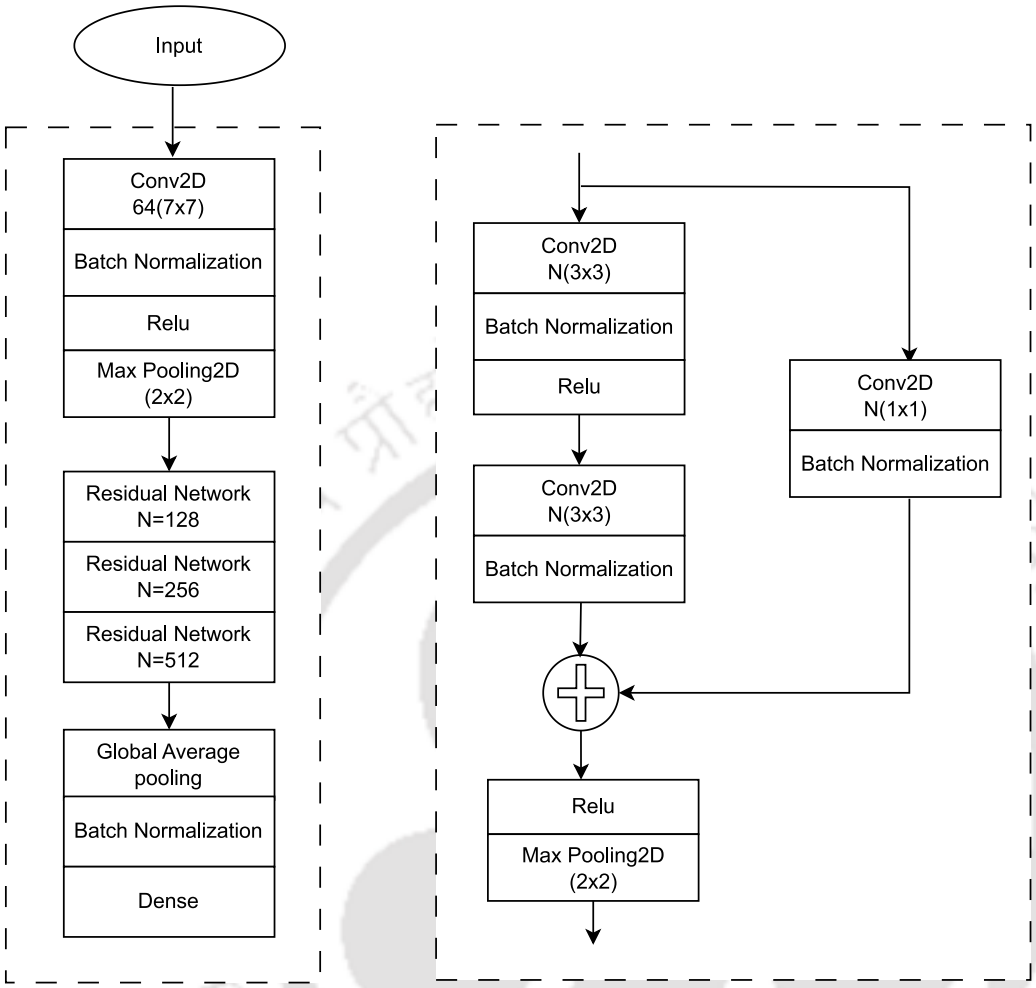



**Fig. 3.3:** Schematic of residual block-based convolution network used in the Siamese architecture of Figure 3.2. Figure (a) represents the overall structure of the convolution network, and (b) depicts the architecture of the residual block used in the convolution network (Here, N represents the number of filters in the residual block).

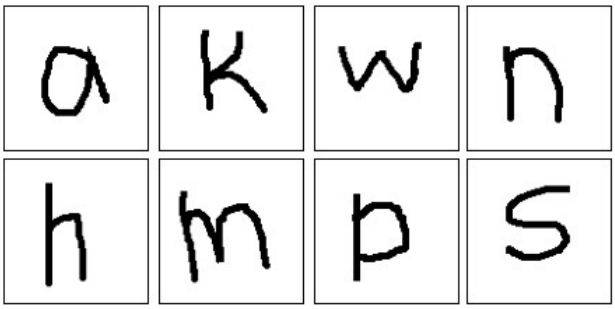


**Fig. 3.4:** Depiction of images of English letters selected from Omniglot database

niglot dataset its weights are frozen. These weights are then used to obtain the feature representation. The fragments extracted at different scales from the word image using SIFT algorithm are resized to a 105×105 while maintaining the aspect ratio and padding (if necessary) with white pixels. These modified image patches are then passed through one of the branches of the trained Siamese network, whose penultimate layer output is used for the purpose of feature representation.

## 3.4 Feature encoding using Sparse PCA

The feature representation of the word fragments from the penultimate layer of the Siamese network are transformed in a reduced-dimensional space by the Sparse PCA method discussed in Section 2.5.1.

In order to obtain the principal directions for projection, let $\boldsymbol{X}$ denote the Siamese feature representation corresponding to a set of fragments of $W$ writers during the training phase. This matrix can be represented as:

$$\boldsymbol{X} = \begin{bmatrix} \cdots & \boldsymbol{X}_1 & \cdots \\ \cdots & \boldsymbol{X}_2 & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \boldsymbol{X}_j & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \boldsymbol{X}_W & \cdots \end{bmatrix} \tag{3.2}$$

where each entry of the sub-matrix $\boldsymbol{X}_j = \begin{bmatrix} \boldsymbol{f}_j^1 & \boldsymbol{f}_j^2 & \cdots & \boldsymbol{f}_j^n & \cdots & \boldsymbol{f}_j^N \end{bmatrix}$ is the Siamese feature representation corresponding to the $n^{th}$ fragment ($1 \leq n \leq N$) for the $j^{th}$ writer. Using the concept of Sparse PCA discussed in Section 2.5.1, the sparse PCA matrix can be derived by projecting the features on to the set of

$$\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1 \ \boldsymbol{\alpha}_2 \ \ldots \ \boldsymbol{\alpha}_l \ \ldots \ \boldsymbol{\alpha}_L] \tag{3.3}$$

The value of $L$ is chosen to be less than the dimension of the feature vector $\boldsymbol{f}_j^n$.

Expanding Equation 3.3, we have

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha^1_{1,1} & \alpha^1_{1,2} & \cdots & \alpha^1_{1,k} & \cdots & \alpha^1_{1,L} \\ \vdots & \vdots & \vdots & \vdots & & \\ \alpha^1_{j,1} & \alpha^1_{j,2} & \cdots & \alpha^1_{j,k} & \cdots & \alpha^1_{j,L} \\ \vdots & \vdots & \vdots & \vdots & & \\ \alpha^1_{N,1} & \alpha^1_{N,2} & \cdots & \alpha^1_{N,k} & \cdots & \alpha^1_{N,L} \\ \vdots & \vdots & \vdots & \vdots & & \\ \alpha^W_{1,1} & \alpha^W_{1,2} & \cdots & \alpha^W_{1,k} & \cdots & \alpha^W_{1,L} \\ \vdots & \vdots & \vdots & \vdots & & \\ \alpha^W_{j,1} & \alpha^W_{j,2} & \cdots & \alpha^W_{j,k} & \cdots & \alpha^W_{j,L} \\ \vdots & \vdots & \vdots & \vdots & & \\ \alpha^W_{N,1} & \alpha^W_{N,2} & \cdots & \alpha^W_{N,k} & \cdots & \alpha^W_{N,L} \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ 1 \le i \le W \\ 1 \le j \le N \\ 1 \le k \le L \\ \\ \\ \\ \\ \end{matrix} \tag{3.4}$$

Here, $\alpha^i_{j,k}$ denotes the $k^{th}$ sparse component for the $j^{th}$ fragment of $i^{th}$ writer. Note that the size of $\boldsymbol{\alpha}$ will be $(WN \times L)$.

## 3.5 Determination of saliency scores

In this Section, we discuss in detail the methodology employed to assign a saliency score to each component of Sparse PCA. This score is obtained by performing a series of operations on the coefficient matrix $\boldsymbol{\alpha}$ obtained in Equation 3.4. These involve:

(i) Generating a histogram for each component of approximated sparse PCA corresponding to a given set of writers.

(ii) Computing relative entropy between the group of writers using the histogram generated in the above step and utilizing the same to obtain the saliency score.

The coefficients generated from $N \times W$ number of fragments is used to construct a histogram of $B$ bins for each of the $L$ individual components of the sparse representation using Equation

(3.5).

$$H^i_{kb} = \begin{cases} H^i_{kb} + 1, & \text{if } h_b \leq \alpha^i_{j,k} < h_{b+1} \\ H^i_{kb}, & \text{otherwise} \qquad 1 \leq b \leq B \end{cases} \tag{3.5}$$

$H^i_{kb}$ denotes the number of nonzero entries for the $k^{th}$ sparse component of the $i^{th}$ writer in the bin $b$ of the histogram.

Following the process of histogram generation, each of the values in the $B$ bins is normalized in the range (0,1)

$$p^i_{kb} = \frac{H^i_{kb}}{\sum_{b=1}^{B} H^i_{kb}} \tag{3.6}$$

For ease of convenience, we represent Equation (3.6) in the matrix form

$$\boldsymbol{P}_k = \begin{bmatrix} \cdots & \boldsymbol{P}^1_k & \cdots \\ \cdots & \boldsymbol{P}^2_k & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \boldsymbol{P}^i_k & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \boldsymbol{P}^W_k & \cdots \end{bmatrix} \quad 1 \leq i \leq W \tag{3.7}$$

where $\boldsymbol{P}^i_k = \begin{bmatrix} p^i_{k,1} & p^i_{k,2} & \cdots & p^i_{k,b} & \cdots & p^i_{k,B} \end{bmatrix}$ represent the normalized probability distribution of the $k^{th}$ sparse component with regards to the $i^{th}$ writer.

Having obtained a set of $W$ distributions in Equation 3.7 for the $k^{th}$ sparse component, we now compute the relative entropy between them by using the Jensen–Shannon divergence (JS) score [90]. The JS divergence by definition measures the similarity between the two distributions based on which scores in the range between 0 (identical) and 1 (maximally different) are assigned. The JS divergence between two distributions is calculated as follows:

$$\begin{aligned} D^i_{j,k} &= \frac{1}{2} d\left(\boldsymbol{P}^i_k \,\|\, \frac{\boldsymbol{P}^i_k + \boldsymbol{P}^j_k}{2}\right) + \frac{1}{2} d\left(\boldsymbol{P}^j_k \,\|\, \frac{\boldsymbol{P}^i_k + \boldsymbol{P}^j_k}{2}\right) \\ d(P \,\|\, Q) &= \sum_{b=1}^{B} p_{k,b} \log\left(\frac{p_{k,b}}{q_{k,b}}\right) \end{aligned} \tag{3.8}$$

The notation, $D^i_{j,k}$ represents the divergence score between the probability distributions of writer $i$ with respect to $j$ for $k^{th}$ sparse component, represented in matrix form as:

$$\boldsymbol{D_k} = \begin{bmatrix} D^1_{1,k} & D^1_{2,k} & \cdots & D^1_{W,k} \\ \vdots & \vdots & \ddots & \vdots \\ D^i_{1,k} & D^i_{2,k} & \cdots & D^i_{W,k} \\ \vdots & \vdots & \ddots & \vdots \\ D^W_{1,k} & D^W_{2,k} & \cdots & D^W_{W,k} \end{bmatrix} \quad \begin{matrix} 1 \leq i, j \leq W \\ 1 \leq k \leq L \end{matrix} \tag{3.9}$$

The divergence matrix $\boldsymbol{D}_k$ is a symmetric matrix of size $(W \times W)$ for each $k \in [1, L]$ with its diagonal entries being zero. The ratio of the mean $\mu_k$ to standard deviation $\sigma_k$ of this matrix $\boldsymbol{D}_k$ gives us an estimate of the spread of the divergence score with regards to the $k^{th}$ writer. Considering the dynamic range of the ratio across writers, it is appropriately scaled using a natural log transformation and utilized as the saliency score for the $k^{th}$ sparse component.

$$\begin{aligned} \tilde{\theta}_k &= ln\left(1 + \frac{\mu_k}{\sigma_k}\right) \\ \theta_k &= \frac{\tilde{\theta}_k}{max([\tilde{\theta}_1\ \tilde{\theta}_2\ \cdots \tilde{\theta}_k \cdots \tilde{\theta}_L])} \end{aligned} \qquad 1 \leq k \leq L \tag{3.10}$$

A high saliency value $\theta_k$ signifies that the divergence score is almost uniform across $W$ writers with the sparse coefficients having a relatively high degree of randomness associated with them. Contrary to this, a low saliency score signifies that considerable variation exists in the divergence score across the $W$ writers with the sparse coefficients being confined within a certain range.

A visual interpretation of the above explanation is provided using histograms presented in Figures 3.5 and 3.6. To begin with, the histograms in Figure 3.5 correspond to the sparse components having the top two maximum and minimum saliency values. Each bin of the histogram provides the value for a writer arrived at by summing up the entries along the rows of the divergence matrix [1] and subsequently normalizing them. The histograms in Figure 3.5 are

[1] It may be clarified here that for each of the sparse components being considered in the present discussion, their corresponding histograms are generated from their respective divergence matrix

obtained by analyzing the fragments corresponding to the first 50 writers selected from the IAM database [75], with each writer contributing 200 fragments. Accordingly, the number of bins in each of them is 50.

For sparse components with the top two saliency values, it can be inferred that the divergence scores are uniformly distributed in sub-figures (a) and (b). Contrary to it, the histogram plots in sub-figures (c) and (d) corresponds to those having low saliency value. Here, we see a noticeable variation amongst the divergence scores.

Further more, we also illustrate the histograms (refer Figure 3.6) depicting the distribution of the sparse coefficients of two writers having the highest and lowest normalized saliency value. The two writers being considered are from the IAM database with ID 000 and 001 respectively. The histograms in Figure 3.6 reveal that the sparse coefficients corresponding to the principal directions with the highest saliency values (sub-figures (a) and (b)) exhibit a high degree of randomness, resulting in dissimilar distributions. In contrast, the sparse coefficients for the principal directions with the lowest saliency values (sub-figures (c) and (d)) display similar distributions, having their coefficient values confined within a specific range.

The spread of the divergence scores, as specified using the saliency value $\theta_k$ can be explored to generate the descriptor as explained in the following section.

## 3.6 Generation of fragment descriptor

Given a handwritten word image $s$, we generate $n_T$ number of word fragments by employing the SIFT algorithm. The encoded feature vector is represented using a matrix $\tilde{\alpha}$ of size $(n_T \times L)$. Recall that the feature matrix is obtained by projecting the fixed Siamese network penultimate layer output onto the principal directions obtained from the Sparse PCA algorithm.

Mathematically, each $(j,k)^{th}$ element of the matrix $\tilde{\alpha}$ represents the sparse component obtained for the $j^{th}$ fragment upon being projected onto the $k^{th}$ principal direction. This elements of $\tilde{\alpha}$ are first $l_2$ normalized along each row. Thereafter they are modified by the saliency value

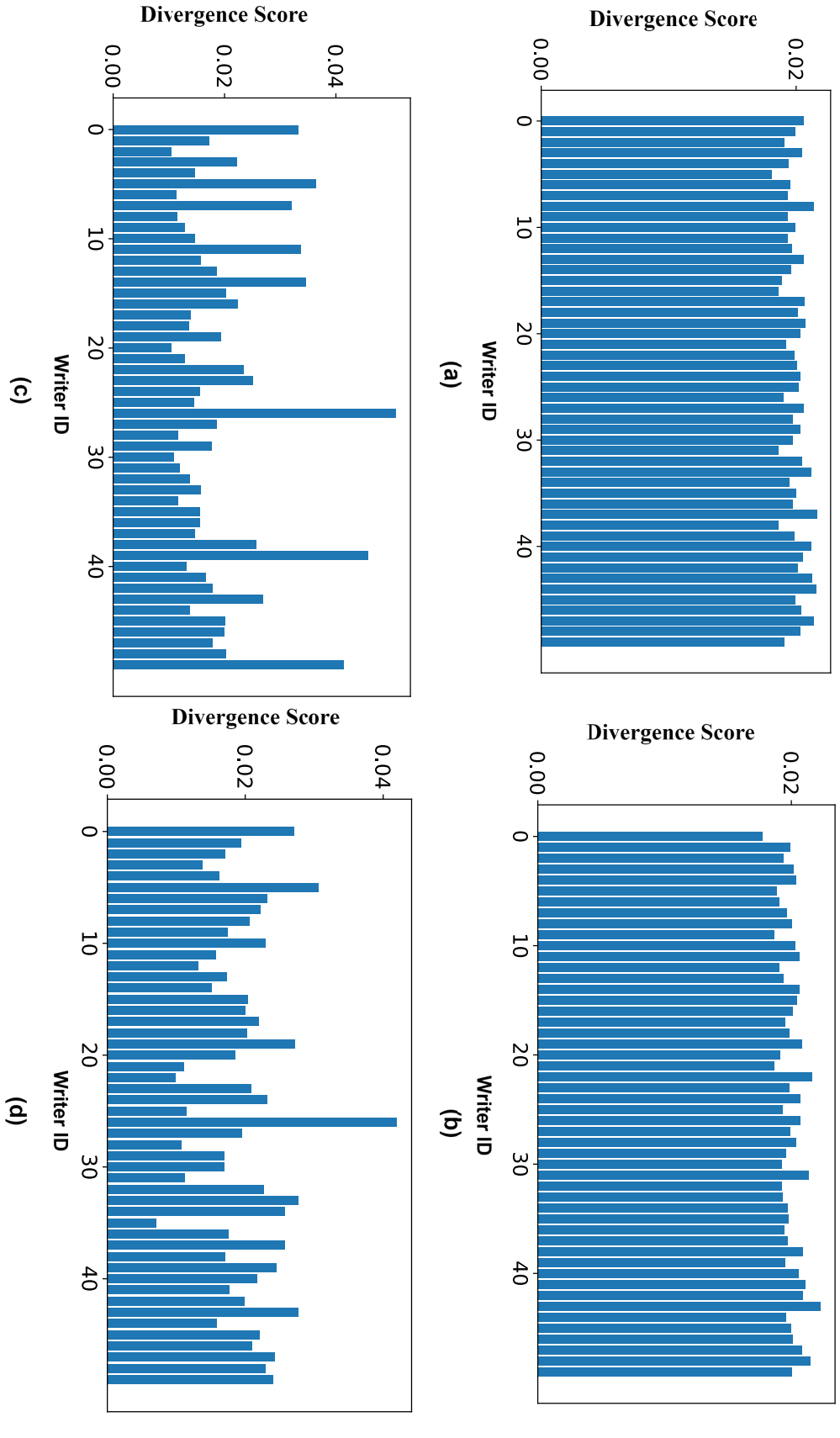



**Fig. 3.5:** Sub-figures (a) and (c) represents histogram of the divergence scores with regards to the sparse component having maximum and minimum saliency value. Likewise, sub-figures (b) and (d) represents the histogram of the divergence scores with regards to the sparse component second maximum and minimum saliency value.

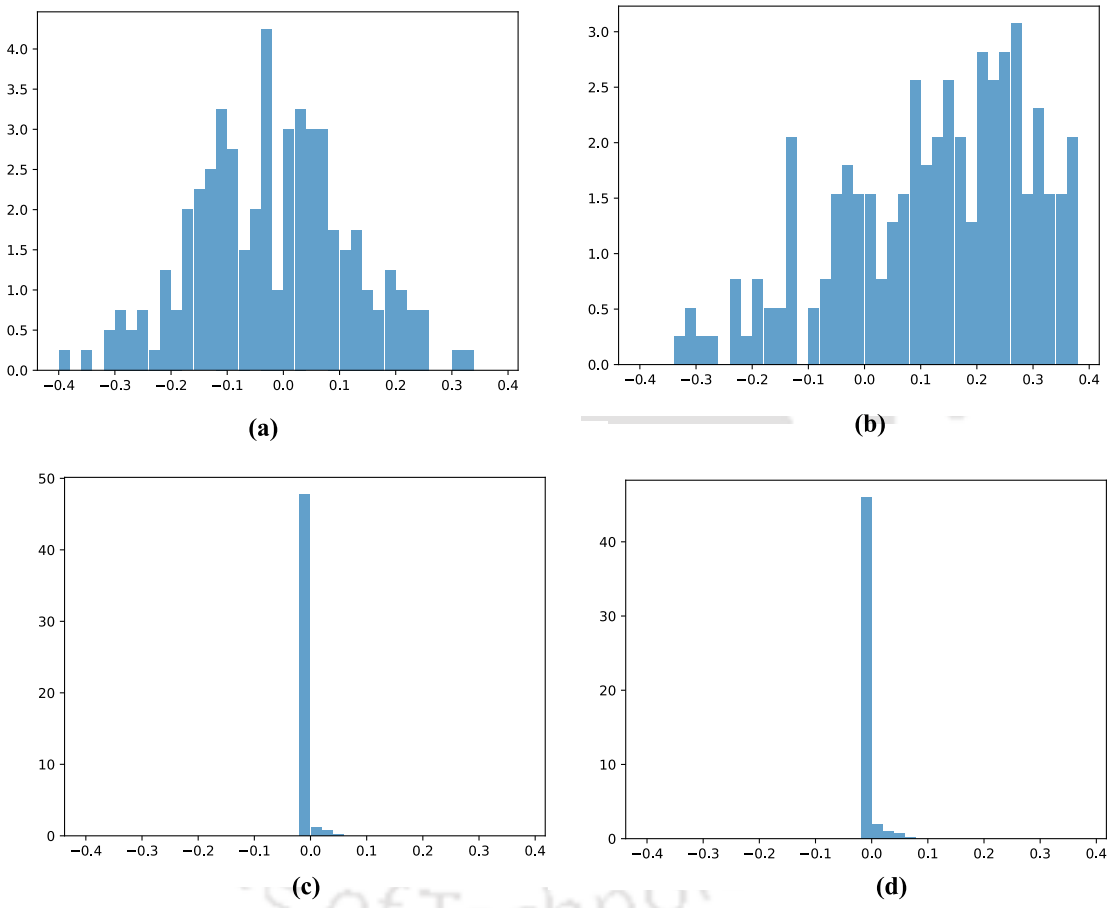



**Fig. 3.6:** The sub-figures (a) and (b) represents the frequency distribution of sparse coefficients corresponding to the component having the highest saliency value for writers selected from the IAM database having ID 000 and 001, Similarly, sub-figures (c) and (d) represent the frequency distribution of the sparse coefficients corresponding to the sparse component with the lowest saliency value for the same set of writers. The x-axis represents the sparse coefficient values and y-axis, their frequency of occurrence.

associated to each sparse component with a power law transformation.

$$\begin{aligned} \hat{e}_{j,k} &= sign(\tilde{\alpha}_{j,k}) * |\tilde{\alpha}_{j,k}|^{\theta_k} \quad 1 \leq k \leq L \\ \hat{\boldsymbol{e}}_j &= [\hat{e}_{j,1}, \hat{e}_{j,2}, \ldots, \hat{e}_{j,L}] \\ \hat{\boldsymbol{e}}_j &= \frac{\hat{\boldsymbol{e}}_j}{\|\hat{\boldsymbol{e}}_j\|_2} \end{aligned} \tag{3.11}$$

Here, $\tilde{\alpha}_{j,k}$ represents entries along the $k^{th}$ component corresponding to the $j^{th}$ fragment. $\hat{e}_{j,k}$ represents the modified sparse representation obtained by exponentially multiplying the entry at location $(j,k)$ of the input matrix $\tilde{\boldsymbol{\alpha}}$ with the corresponding saliency value $\theta_k$. In a sense, the power law transformation being proposed by us is adaptive in nature.

At this point, it may be mentioned that prior work related to writer recognition [51,91] used a fixed value of power $\phi_k$ (in the range between $0 \leq \theta_k \leq 1$) to modify the feature vector input. Contrary to this, we experimentally observed that having an adaptive power-law normalization during the process of writer descriptor generation improves the overall performance of the writer identification system. This will be elaborated further in Section 3.7

The modified descriptors of the fragments of the word $s$ are passed through a set of writer-specific SVM classifiers [76] trained in a one-vs-all framework. Each SVM classifier assigns a value to each fragment descriptor depending on its proximity to the separating hyperplane. These are then bounded between $[0-1]$ by passing them through a sigmoid function. Following this, the scores are accumulated and averaged with respect to each of the writer specific SVM classifiers.

$$P_i(s) = \frac{1}{n_T} \sum_{j=1}^{n_T} p_j^i(s) \quad 1 \leq i \leq W \tag{3.12}$$

Here, $p_j^i(s)$ represents the score obtained for the $j^{th}$ fragment from the SVM classifier trained on samples of the $i^{th}$ writer. Accordingly, $P_i(s)$ denotes the average score obtained for an input word image containing $n_T$ number of fragments.

The final prediction is made based on the label of the writer for which the highest score is obtained.

$$\hat{y} = \underset{i \in \{1,\ldots,W\}}{arg\ max} \quad P_i(s) \tag{3.13}$$

## 3.7 Experimental results and discussion

In this Section, we discuss in detail the set of experiments carried out on the word images from three databases namely IAM [75], CVL [77] and CERUG-EN [78]. Each experiment is conducted through ten trials, and the average writer identification rate is reported. Each trial in the context of this study relates to the following collection of steps, namely the selection of word images for training and testing, fragment generation, Siamese network feature description, determination of sparse principal components, computation of saliency values and final writer classification based on the modified descriptors of its constituent fragments.

### 3.7.1 Implementation details

Our proposed Siamese network is built using the TensorFlow framework with the Adam optimizer [92] used for optimization of the weight parameters. The batch size for training is set to 16. The learning rate is initialized to 0.001 and is progressively reduced by half per 20 epochs. The whole network is trained for 100 epochs by using the triplet loss function.

To establish the writer's identity based on the extracted features, we use a one-vs-all SVM classifier with a radial basis function (RBF) kernel. The optimal values for the RBF parameters,$C$ and $\gamma$, are determined through cross-validation.

### 3.7.2 Performance evaluation of baseline Siamese architecture

The aim of this experiment is to investigate the effect of training the Siamese network and assess its impact on the performance of the writer identification system. We separately train the parameters of the architecture from scratch on two datasets, namely the EMNIST and Omniglot. Subsequent to training, we freeze the weights and apply the network on the individual fragments constituting the word images. The writer fragments are then passed through one of the branch of the trained Siamese network, with the output of the penultimate layer serving as the feature representation. These feature representations are subsequently evaluated using a set of writer-specific SVMs. The individual scores from the extracted word fragments are then aggregated to determine the writer's identity.

**Table 3.1:** Average writer identification rate (in %) based on the word image fragment representations obtained from the penultimate layer of the Siamese architecture. The pre-training of the network is done on the samples of the Omniglot dataset with varying sizes of the penultimate layer. The best identification rate is marked in **bold**

| Database | Size of penultimate layer | | | |
|---|---|---|---|---|
| | 256 | 512 | 1024 | 2048 |
| IAM | 80.64 | 83.43 | **85.22** | 81.8 |
| CVL | 72.58 | **76.83** | 75.11 | 74.38 |
| CERUG-EN | 65.23 | **69.14** | 66.57 | 63.58 |

**Table 3.2:** Average writer identification rate (in %) based on the word image fragment representations obtained from the penultimate layer of the EMNIST architecture. The pre-training of the network is done on the samples of the Omniglot dataset with varying sizes of the penultimate layer. The best identification rate is marked in **bold**.

| Database | Size of penultimate layer | | | |
|---|---|---|---|---|
| | 256 | 512 | 1024 | 2048 |
| IAM | 75.89 | **77.37** | 77.20 | 76.88 |
| CVL | 60.87 | **65.27** | 65.14 | 64.38 |
| CERUG-EN | 56.19 | 62.28 | 65.80 | **68.28** |

Tables 3.1 and 3.2 present the result of this experiment on the word images of the three datasets. In particular, the size of the penultimate layer being used for feature extraction is varied from 256 to 2048. From the entries, it can be observed that the network trained on the Omniglot dataset achieves better performance as against the one on the EMNIST dataset. This observation can be attributed to the fact that the Omniglot dataset contains handwriting samples collected from diverse set of languages,ensuring the generalization of writer features and minimizing the likelihood of the network to rely on writer-database-specific features. Based on this, we use the network trained on the Omniglot dataset in the proposed identification system.

Moving further, the optimum size of the penultimate layer to be used for feature representation is decided by examining its overall accuracy in identifying the writers. From Table 3.1 we observe that the best identification rate achieved for IAM database corresponds to the size of 1024, while for CVL and CERUG-EN databases it is achieved at 512.

Accordingly, for the remainder of the experiments in this Section, the penultimate layer size in the Siamese network is fixed to 512 (for CVL and CERUG-EN databases) and 1024 (IAM database) respectively.

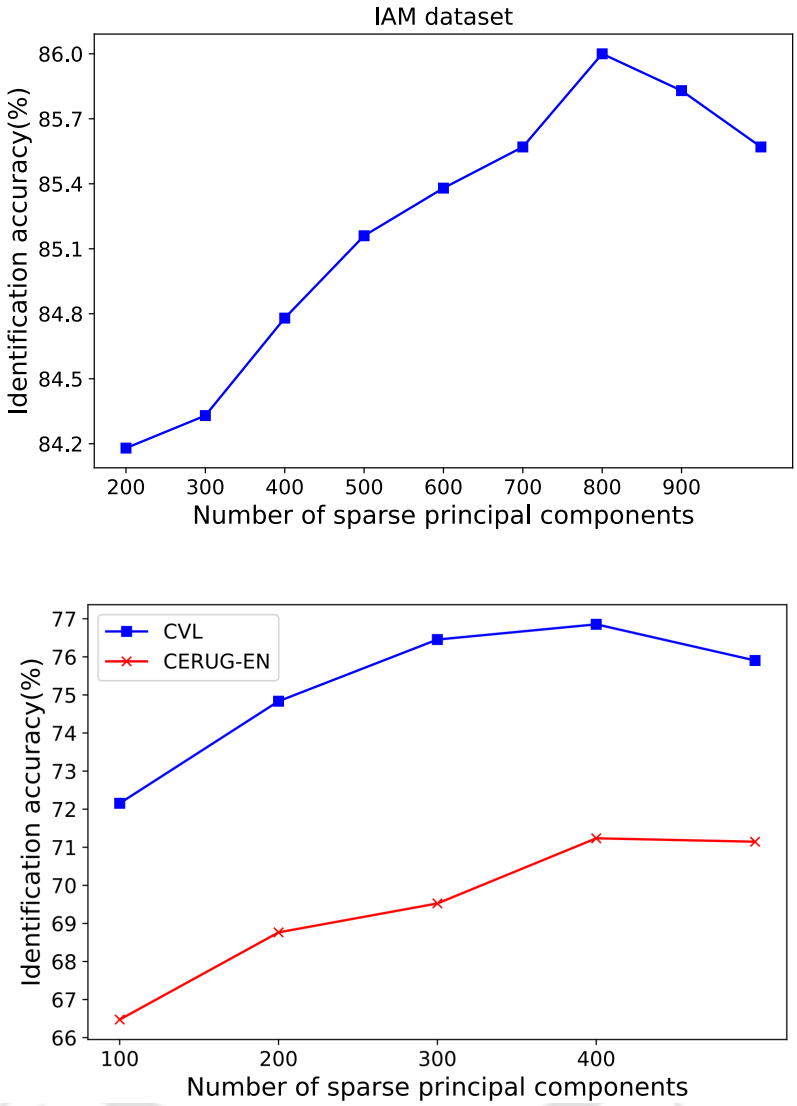



**Fig. 3.7:** Writer identification accuracy with varying number of principal components for the IAM, CVL and CERUG-EN databases

### 3.7.3 Performance evaluation using varying number of sparse components

In this experiment, we assess the performance of the system for different number of sparse coefficients $L$ [2]. We vary the number of principal directions for projection from 200 to 1000 for the IAM database and 100 to 500 for the CVL and CERUG-EN databases in increments of 100. Figure 3.7 presents the result of this analysis on the three databases with the accuracy reported at the word level.

[2]Note that here the sparse coefficients are not modified by the saliency values using the adaptive power normalization.

The accuracy of the system increases with the number of principal directions with a maximum value at 850 for the IAM and 450 for the CVL and CERUG-EN databases respectively. Thereafter, we observe a decrease in the identification rate. The drop can be explained by the fact that, at times, when the fragments of the word are being transformed to a large number of sparse principal components, a perturbation with respect to one or more entries may lead to a modification in the overall feature representation, thus leading to misclassification.

Additionally, we aim to demonstrate the advantages of sparse representation compared to traditional PCA. To achieve this, we will evaluate the performance of Sparse PCA with an optimal dictionary size (as obtained above) and compare it with that of standard PCA representation. The result of this analysis is presented in Table 3.3.

**Table 3.3:** Comparison of average Top-1 writer identification rate (in %) on word images for the PCA and Sparse PCA representations employed in this work

| Database | Siamese + PCA | Siamese + Sparse PCA |
|---|---|---|
| IAM | 86.05 | 86.55 |
| CVL | 77.88 | 79.67 |
| CERUG-EN | 70.19 | 71.41 |

### 3.7.4 Incorporation of saliency values on the sparse principal components

In this sub-section, we study the effect of incorporating the saliency score into the sparse-based representation and its effect on the overall accuracy of the system. More specifically, we compare the performance of the sparse and modified sparse-based framework (using an adaptive power law transformation) over the traditional method of training an SVM classifier on the output representation of the Siamese network. The Top-1 and Top-5 results obtained at the word level for IAM, CVL and CERUG-EN databases are shown in Table 3.4.

From the Table, we observe that as a result of using the sparse-based representation, the system attains a Top-1 accuracy of 86.55 %,79.67% and 71.41 % respectively for IAM, CVL and CERUG-EN databases. Contrast to it, the baseline Siamese feature representation achieves a performance of 85.22%, 76.83% and 69.14% respectively. The identification rate increases further to 87.24%, 81.85% and 74.56% respectively by incorporating adaptive power normal-

ization in the sparse components.

To further corroborate the effectiveness of the adaptive power normalization method, we consider evaluating our proposal using a fixed power normalization coefficient $\theta$ over the input matrix as follows:

$$\begin{aligned} \hat{e}_{j,k} &= sign(\tilde{\alpha}_{j,k}) * |\tilde{\alpha}_{j,k}|^{\theta} \quad 1 \leq k \leq L \\ \hat{\boldsymbol{e}}_j &= [\hat{e}_{j,1}, \hat{e}_{j,2}, \ldots, \hat{e}_{j,L}] \\ \hat{\boldsymbol{e}}_j &= \frac{\hat{\boldsymbol{e}}_j}{||\hat{\boldsymbol{e}}_j||_2} \end{aligned} \tag{3.14}$$

We vary $\theta$ within the range of 0 to 1 and compare its performance with the adaptive power normalization derived from Equation 3.11. The results are presented in Table 3.5 for the databases. From the entries, it may be inferred that for the value of $\theta$ in the range between 0.1 to 0.5, the overall accuracy of the system decreases compared to the sparse-based representation (corresponding to $\theta$=1). This is primarily due to the effect of the power-law transformation function that increases the spread of the sparse coefficients leading to a high degree of randomness amongst its values. On the other hand, an adaptive power normalization method caters to this issue by varying each coefficient based on its normalized saliency value in a way such that the relative spread of the distribution for the components having higher saliency values is less to those with lower values. The effect of the transformation on the distribution is demonstrated with the help of histograms in Figure 3.8.

The histograms shown in Figure 3.8(a) and 3.8(d) corresponds to the distribution of the original sparse components having the highest and lowest saliency values. These are constructed by utilizing 200 fragments from a writer of the IAM database having ID 003. Multiplying the sparse components with a fixed power normalization factor of 0.5 leads to their values being redistributed to the surrounding bins thus increasing the overall spread of the distribution (see Figure 3.8(b) and 3.8(e)). By using the adaptive power normalization, we strive to adjust the spread of the sparse components in a way that those with low saliency values are spread more as against to their counterparts having high saliency values (refer Figure 3.8(c) and 3.8(f)).

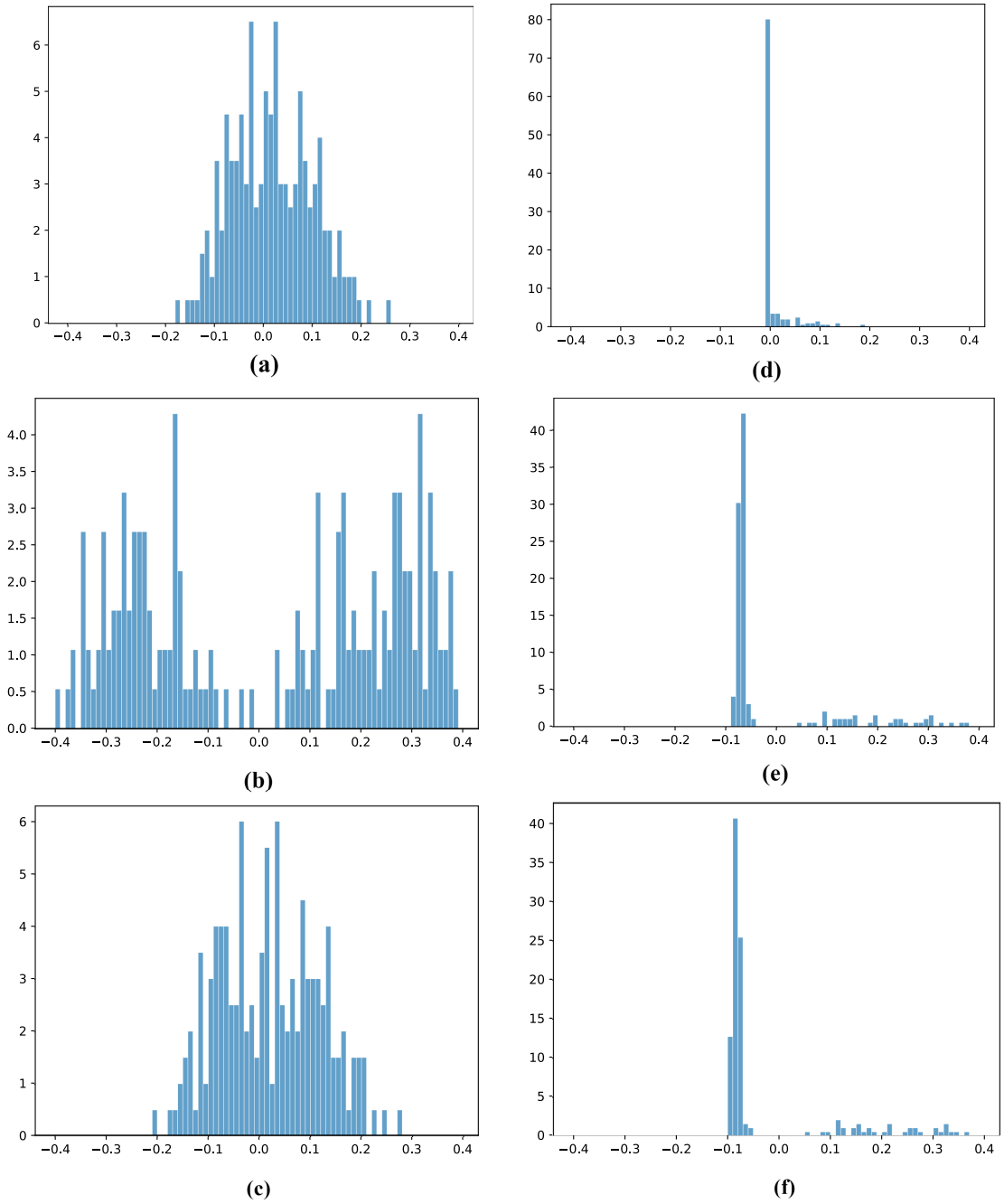



**Fig. 3.8:** Histograms shown in sub-figures (a) and (d) corresponds to the distribution of the original sparse components having the highest and lowest saliency values. These are constructed by utilizing 200 fragments from a writer of the IAM database having ID 003. Sub-figures(b) and (e) display the effect of using a fixed power normalization factor of 0.5 on the overall sparse principal component distribution. Finally, sub-figures (c) and (f) are the histograms after the transformation of distribution with adaptive power normalization.

**Table 3.4:** Comparison of average Top-1 and Top-5 writer identification rate (in %) on word images for the different representations employed in this work

| Representation | IAM | | CVL | | CERUG-EN | |
|---|---|---|---|---|---|---|
| | **Top 1** | **Top 5** | **Top 1** | **Top 5** | **Top 1** | **Top 5** |
| Siamese network | 85.22 | 92.39 | 76.83 | 91.11 | 69.14 | 90.28 |
| Siamese + Sparse PCA | 86.55 | 93.10 | 79.67 | 92.57 | 71.41 | 91.57 |
| Siamese + Sparse PCA (with saliency) | 87.24 | 94.15 | 81.85 | 93.68 | 74.56 | 93.82 |

**Table 3.5:** Top 1 average identification rates for different values of power factor $\theta$ in Equation 3.14. The best accuracy is achieved when it is adaptive to the sparse components (Equation 3.11).

| $\theta$ | **IAM** | **CVL** | **CERUG-EN** |
|---|---|---|---|
| 0.1 | 85.5 | 76.51 | 66.38 |
| 0.3 | 86.20 | 77.60 | 69.23 |
| 0.5 | 86.45 | 78.52 | 71.14 |
| 0.7 | 86.50 | 79.23 | 71.61 |
| 0.9 | 86.61 | 79.86 | 71.90 |
| 1 | 86.55 | 79.67 | 71.42 |
| Adaptive $\theta$ | **87.24** | **81.85** | **74.56** |

### 3.7.5 Statistical significance and time complexity

In this Section, we establish that the results of the proposed writer descriptor (using sparse representation) is statistically significant when compared to the baseline Siamese framework. The analysis was conducted with a significance level of 0.05 using the Student's t-test approach. Table 3.6 outlines the p-values obtained with regard to the algorithms across the three data sets. A trend of low values for p is empirically observed across each of the entries in the table — thus indicating the statistical significance of our proposal.

For the sake of completeness, we also provide the average execution time for the proposed modified sparse-based representation framework based on conducting multiple trials on the word samples of the IAM database. The analysis was done on an HP desktop with 16GB RAM and an Intel i7 processor. Table 3.7 tabulates the average execution time corresponding to the set of operations carried out at each stage:

- Pre-processing: The operations carried out in this step correspond to the removal of back-

**Table 3.6:** Statistical significance of the proposed sparse descriptor (with and without saliency value incorporation) over the baseline Siamese network representation. We employ the Student's $t$-test to obtain the values.

| Database | Siamese + Sparse PCA | Siamese + Sparse PCA (with saliency) |
|---|---|---|
| IAM | $2.5 \times 10^{-6}$ | $2.84 \times 10^{-7}$ |
| CVL | $9.62 \times 10^{-7}$ | $2.19 \times 10^{-7}$ |
| CERUG-EN | $2.37 \times 10^{-3}$ | $1.01 \times 10^{-6}$ |

ground variation from the input word image and fragment generation using the SIFT algorithm.

- Feature extraction: This stage represents the average time elapsed in obtaining the feature representation using the penultimate layer of the Siamese Network on the image fragments.
- Sparse PCA : The average time indicated involves obtaining the principal directions in the Sparse PCA framework, For this, we consider the fetures from $(W \times N)$ fragments selected from a set of $(W = 200)$ writers chosen randomly with each writer contributing $(N = 250)$ fragments.
- Saliency score generation: These correspond to the set of operations mentioned in section 3.5.
- Fragment representation: This step consists of obtaining the feature representation for each fragment by projecting them using sparse PCA and incorporating the saliency values associated with each sparse component.
- SVM training: It represents the average time lapse that occurs in training the writer specific SVM classifiers at the fragment level.
- SVM testing: It denotes the average time involved in acquiring the final classification score for the word image. This is achieved by aggregating the individual scores for each of the fragments associated with a particular word.

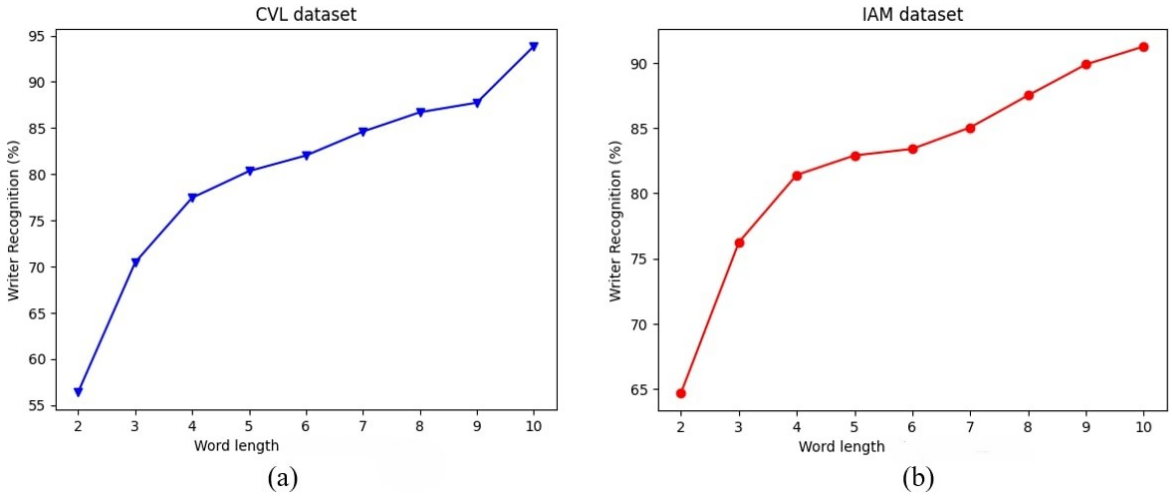
Based on Table 3.7 it can be inferred that the process of Sparse PCA and saliency score generation are computationally expensive tasks. However, these are non-recurrent and are performed only once during the training phase.

**Table 3.7:** Average time complexity (in seconds) of the proposed algorithm.

| Description | Execution time (sec) |
|---|---|
| Pre-processing (per word) | 3.06 |
| Feature extraction(per word) | 4.27 |
| Sparse PCA | 4064 |
| Saliency score generation | 9820 |
| Writer Descriptor(per word) | 0.17 |
| SVM training (per writer) | 80.98 |
| SVM testing(per word) | 0.25 |

### 3.7.6 Performance of writer identification with word images of different lengths

Figure 3.9 shows the performance of our algorithm when tested against word images of different lengths taken from CVL and IAM datasets respectively. The accuracy of the word images containing around four characters hovers around 80% for the two databases. However, as the number of characters in a word image increases, an improvement of around 18% can be observed. This is owing to the fact that with an increase in the number of characters, the number of fragments also increases proportionately, thereby capturing writer-style information more effectively.




(a) (b)

**Fig. 3.9:** Average writer identification rates with different word lengths (in characters) on the CVL and IAM data base.

**Table 3.8:** Comparison of our proposal with previous works.

| Method | IAM | | CVL | | CERUG-EN | |
|---|---|---|---|---|---|---|
| | Top 1 | Top 5 | Top 1 | Top 5 | Top 1 | Top 5 |
| Hinge [14] | 13.8 | 28.3 | 13.6 | 29.7 | 14.4 | 32.8 |
| Quill [79] | 23.8 | 44.0 | 23.8 | 46.7 | 24.5 | 51.9 |
| CoHinge [80] | 19.4 | 34.1 | 18.2 | 34.2 | 17.7 | 34.0 |
| QuadHinge [80] | 20.9 | 37.4 | 17.8 | 35.5 | 17.0 | 36.0 |
| COLD [81] | 12.3 | 28.3 | 12.4 | 29.0 | 17.3 | 42.2 |
| Chain Code Pairs [20] | 12.4 | 27.1 | 13.5 | 30.3 | 14.5 | 33.0 |
| Chain Code Triplets [20] | 16.9 | 33.0 | 17.2 | 35.4 | 17.8 | 38.0 |
| WordImgNet [58] | 52.4 | 62.5 | 62.5 | 82.0 | 74.3 | 94.6 |
| FragNet-64 [58] | 72.2 | 88.0 | 79.2 | 93.3 | 75.9 | 94.7 |
| Vertical GR-RNN(FGRR) [59] | 83.3 | 94.0 | 83.5 | 94.6 | 70.2 | 91.6 |
| Horizontal GR-RNN(FGRR) [59] | 82.4 | 93.8 | 82.9 | 94.6 | 68.9 | 90.9 |
| Proposed Methodology (Modified sparse Representation) | 87.24 | 94.15 | 81.85 | 93.68 | 74.56 | 93.82 |

### 3.7.7 Discussion of prior works

In this sub-section, we evaluate the performance of our proposed approach with traditional handcrafted as well as recent deep-learning-based systems built for identifying the authorship of handwritten word images.

From the entries in Table 3.8, it can be observed that the identification accuracy of traditional handcrafted features are low. This is because these algorithms make predictions based on the statistical data collected from the input image, for which a certain number of text samples are required. Nonetheless, this information is insufficient to generate a stable representation of the input text sample.

Contrast to hand crafted features, the presence of multiple feature maps in each layer of a convolution network enables it to capture a diverse quantity of information from the word-level training data, thus contributing to improved performance. As a matter of fact, with regards to the recent deep learning methods, our proposed algorithm provides an identification rate that is at par with the performance of the features learned in [58, 59].

It may be noted that the systems being enumerated in Table 2.9 can differ with regards to the feature set as well as the classification, and enrollment strategies. Therefore, it should be borne in mind that a direct one-to-one comparison with these approaches may not be fair.

## 3.8 Conclusion

In this study, we offer a novel method for determining the identity of a writer based on offline handwritten word images in a text-independent framework. The main highlights of our approach are enumerated as follows:

(i) Exploring the Siamese framework for feature representation of the fragments of a word.

(ii) Exploring a sparse-based PCA model for the representation in (i).

(iii) Proposing an approach for assigning saliency score to each component in the sparse PCA representation, thereby enhancing their discriminative ability.

The efficacy of our proposed system is demonstrated on three databases namely CVL, IAM and CERUG-EN, and the result thus obtained is comparable to the state of art methods.

# 4

# Exploration of an attention based multi-stream CNN network

## Contents

## 4.1 Introduction

In this Chapter, we propose an end-to-end framework based on a multi-stream CNN for establishing the authorship of handwritten word images. The network is trained on image fragments and utilizes two parallel CNN modules. One module adopts a writer-dependent training approach to extract writer-specific details, while the other considers a writer-independent strategy to capture global features across all writers. This dual network architecture enables the network to effectively capture the intricate characteristics of a fragment contributed by a writer.

Further to the above contribution, we investigate the integration of an attention mechanism into our proposed two stream network. This in a way enriches the representation power of the network by highlighting important regions within the fragments of the writers. By generating attention weights, we identify areas of importance in a fragment that further assists in refining the ability of the network to discern relevant features.

Experiments performed with the proposed CNN framework show an improved performance in identification rates over the current state-of-the-art techniques.

In essence, the key contributions of this Chapter include:

- Exploring writer characterization using fragments extracted from the word image.
- Proposing a multi-stream end-to-end Convolution network trained on fragments corresponding to writer-dependent and independent datasets.
- Investigating the effect of attention mechanism on the multi-stream Convolution network.

The organization of this Chapter is as follows - We commence by presenting a block schematic of our proposed framework in Section 4.2. Following this, Section 4.3 offers a comprehensive explanation of the proposed multi-stream Convolution network for training the fragments of the handwritten word. The architectures corresponding to both writer dependent and writer independent modules are discussed in sufficient detail. Section 4.4 presents a detailed analysis of the attention block with its incorporation in the multi-stream CNN network, while Section 4.5 delves into the training associated with the proposed CNN model as well as the classification strategy being adopted to establish the identity of the writer from the fragments of the

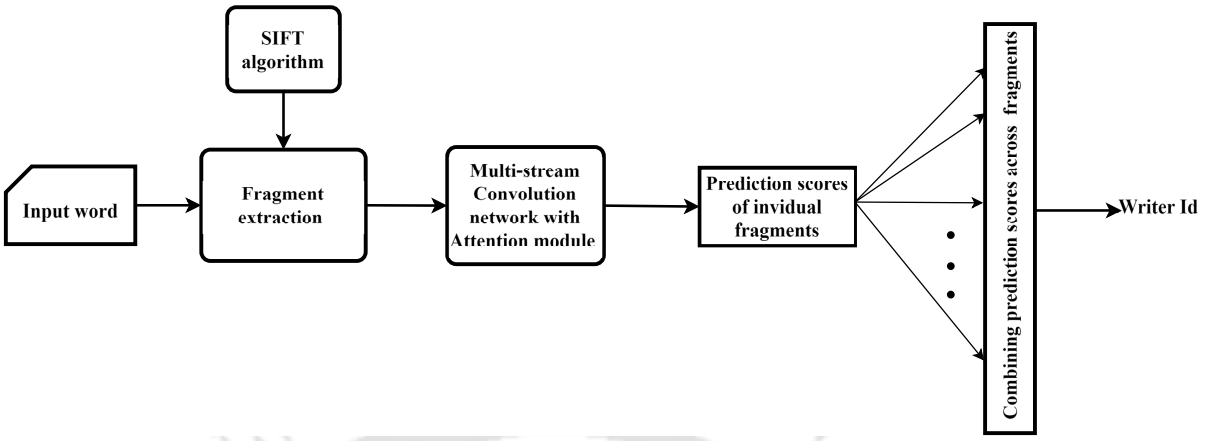



**Fig. 4.1:** Block diagram of the proposed system

handwritten word. The implementation details, alongside with experimental results on the three datasets namely IAM, CERUG-EN and CVL are discussed in Section 4.6. Finally, our work is summarized in Section 4.7.

## 4.2 Block schematic of the proposed framework

Figure 4.1 provides a pictorial description of our approach. To begin with, the preprocessed input word image is passed through a fragment extraction block. This block uses SIFT algorithm [1] to extract image fragments at various levels of abstraction employing a scale space-based approach. The fragments are fed individually to the multi-stream CNN with attention network, which generates a score in the range $(0, 1)$ with regards to each of the enrolled writers at the output layer of the network. Thereafter, the scores obtained from the set of fragments of the word image are averaged across to establish the identity of the writer.

Before moving ahead, we would like to reiterate the following in context to the present work:

- The pre-processing step associated with respect to obtaining the word image involve operations such as skew-correction, text-line segmentation, removal of ink blots and extraneous marks along with background noise. This is elaborated in Section 2.8.1.
- The process of generation of fragments from word image involving SIFT algorithm is discussed in Section 2.2.

## 4.3 Multi-stream CNN Model

Subsequent to the fragment extraction, it needs to be mapped to a feature vector for subsequent analysis. For this, we propose a multi stream Convolution Neural Network (CNN) of the form shown in Figure 4.2. The specifics of the writer-dependent and writer-independent modules are outlined below.

### 4.3.1 Writer dependent (WD) module

Every writer possesses a unique handwriting style, characterized by distinct segments or patterns that may exhibit certain similarities among them. The process of identifying these shared traits within a handwritten sample serves a crucial role in authenticating the authorship of the document. The objective of the writer-dependent Convolutional Neural Network (CNN) module is to extract pertinent details from the input fragments with regards to a writer.

The architecture of the writer-dependent module comprises a series of convolution layers, each followed by Batch Normalization and a Rectified Linear Unit (ReLU) layer sequentially. These convolution layers are equipped with a range of filters spanning from 64 to 512 having size of $(n \times n)$, where $n$ takes values from $\{1, 3, 7\}$. The kernel size of the initial convolution filter is selected to be of size $(7 \times 7)$ to capture the global pattern and structure in a fragment image. Notably, certain convolution blocks incorporate skip connections, thereby transforming them into residual blocks, as depicted in Figure 4.4. The utilization of skip connections enhances the ability of the network to capture intricate features while mitigating the vanishing gradient problem.

Most of the convolution operations within the writer-dependent block is executed via a separable convolution layer. Unlike standard convolution, which applies a kernel across the entire input volume simultaneously, separable convolution operates in two stages. Initially, a convolution operation is conducted along each channel of the input independently. Subsequently, a $(1 \times 1)$ point-wise convolution is performed across the channels [93] as depicted in Figure 4.3. This strategy is adopted to reduce the overall number of trainable parameters as compared to standard convolution, which in turn improves computational efficiency.

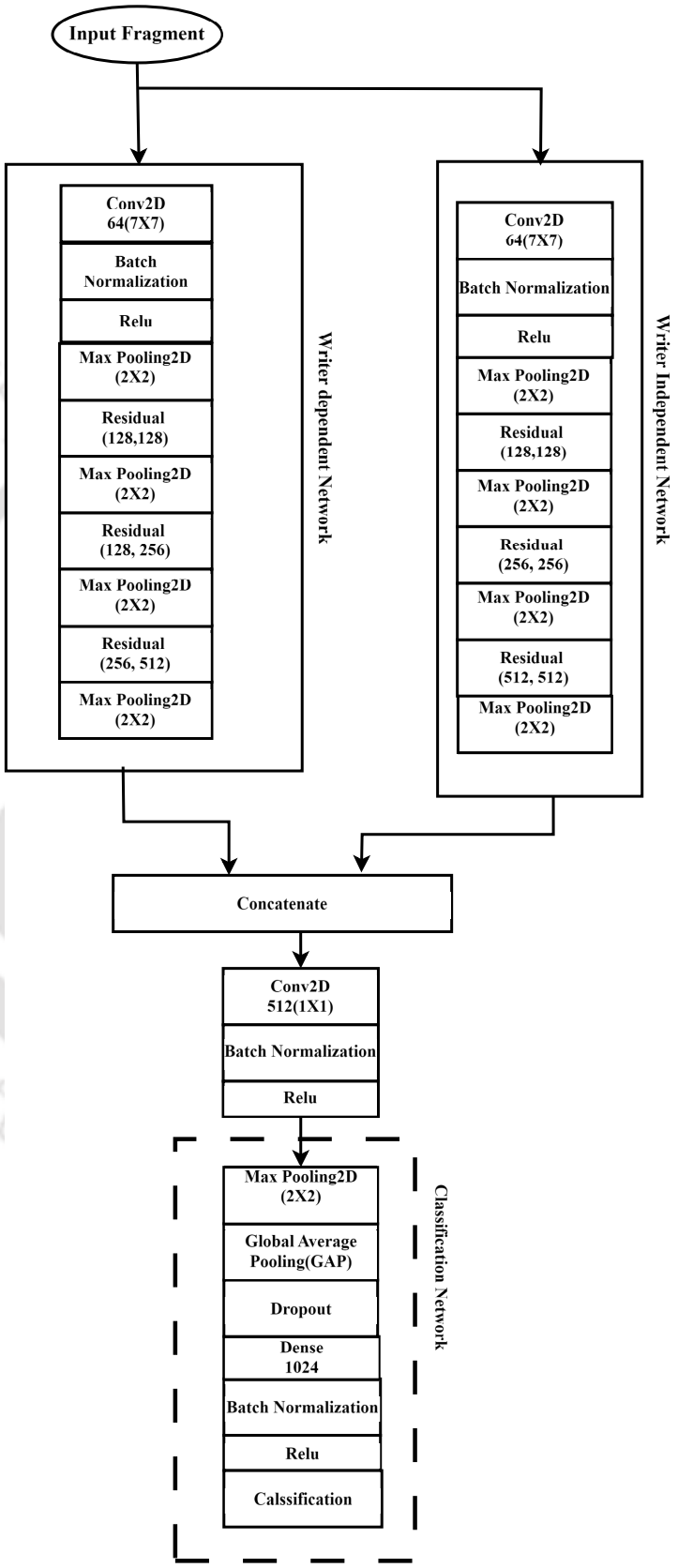



**Fig. 4.2:** The architecture of multi stream CNN model used for training the fragments of the words. Each convolution block and its variant (represented as Conv2D/Separable Conv2D) is followed by entries signifying the number of filters and their kernel size respectively.

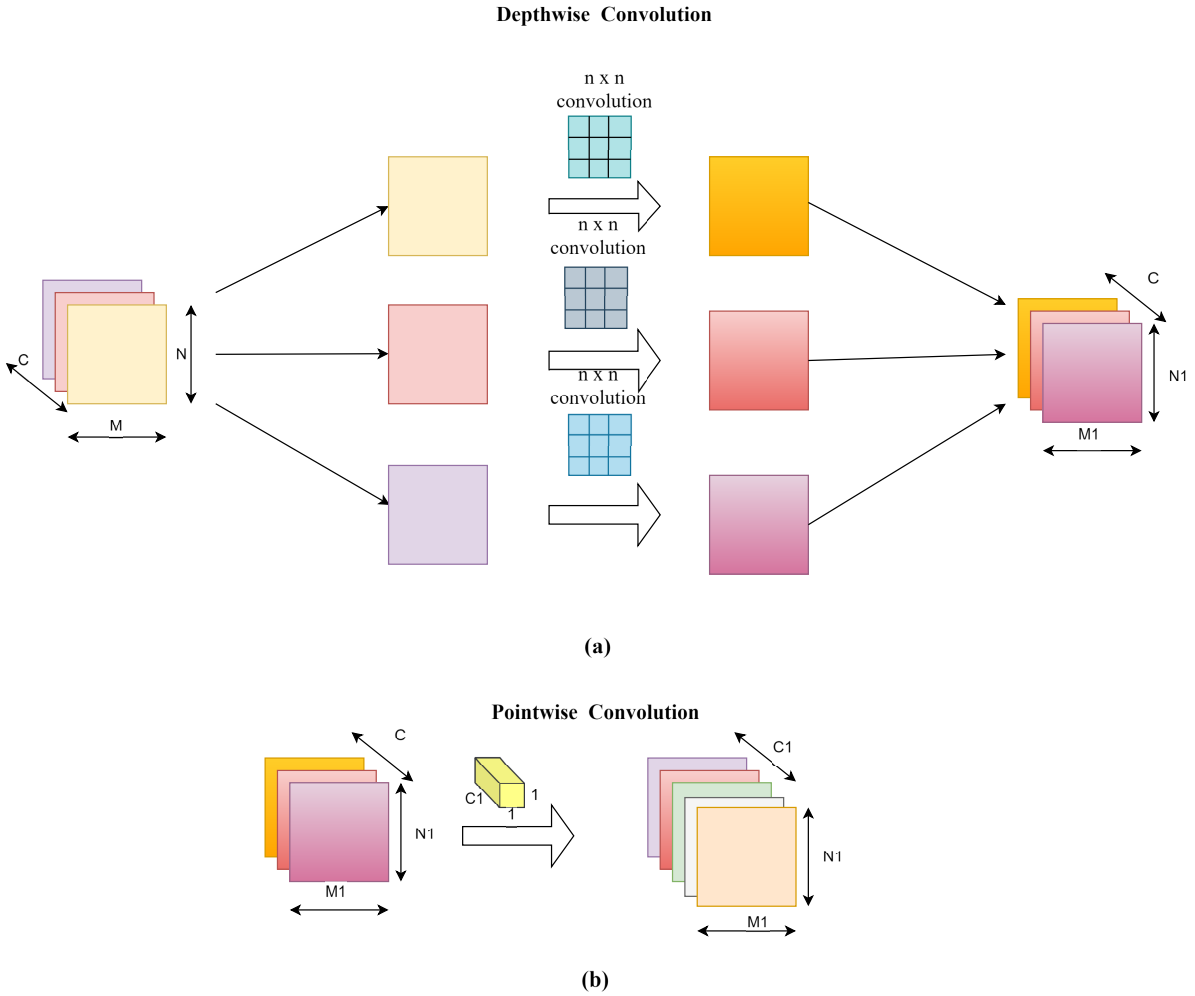



**Fig. 4.3:** Visual illustration of a depth-wise separable convolution. In the first stage standard convolution is applied along each channel of the feature map. This is followed by applying a point-wise convolution operation across the channels. In this figure $M$,$N$,$C$ and $M1$,$N1$,$C1$ represents the width, height and number of channels in the input and output feature map respectively.

### 4.3.2 Writer independent (WI) module

The main function of this module is to extract generalized features from the input fragment that are writer-independent. We seek to combine it with the writer dependent features with the hope of obtaining a more robust representation of the fragments as well as better generalization. With regards to the architecture for this module, we employ a dissimilarity-based approach that focuses on recognizing distinctions or dissimilarities between the fragments irrespective of the identity of the writer.

The proposed dissimilarity measure is framed as a distance metric within an embedding space, wherein fragments sharing commonalities between writers are brought closer together.

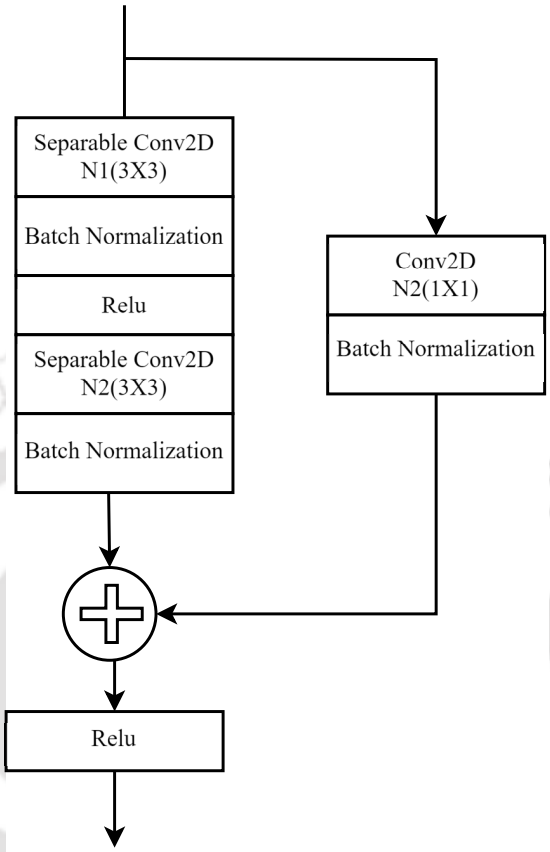



**Fig. 4.4:** The architecture of the residual block $(N1, N2)$ used in the architecture of Figure 4.2. Each residual block in the network is followed by entries N1 and N2 specifying the number of filters in the residual block.

For this, we utilize a Siamese architecture that is pre-trained on the samples of the Omnniglot data set using the triplet loss function.

The block diagram of the CNN used in the Siamese architecture is shown in Fig. 4.5(a). This network comprises of 10 convolution layers with skip connections being added to some of these convolution blocks transforming them into residual block as shown in Fig. 4.5(b). Similar to the writer-dependent module each convolution layer is assigned a set of filters in the range from 64 to 512 with a size of $(n \times n)$, where $n \in (1, 3, 7)$ followed by a batch normalization and a Rectified Linear Unit (ReLU) layer.

Our model uses a configuration in which the fully connected layer is preceded by a Global average pooling layer instead of a flattened version of the convolution layer. Furthermore, a large portion of the operations are implemented using separable convolutions.

As mentioned, the Siamese architecture is pre-trained on the Omniglot dataset [88] using the

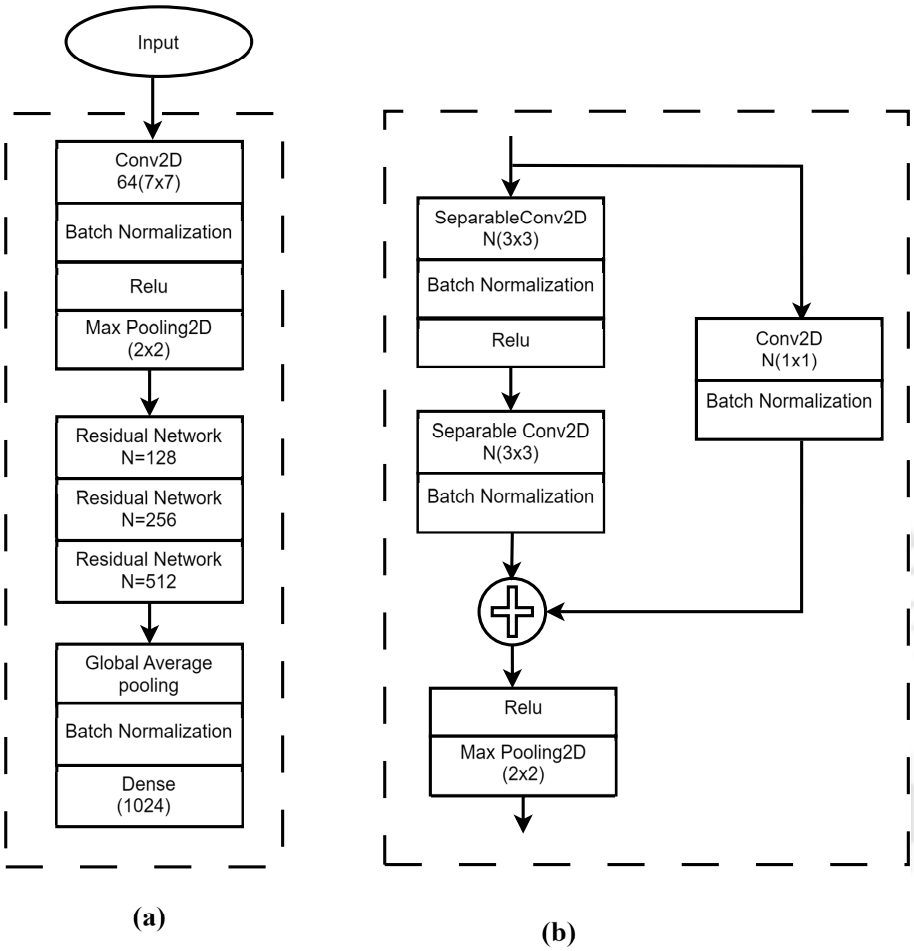



**Fig. 4.5:** Sub-figure (a) presents the convolution Network used in the Siamese architecture for extracting fragment features in the writer independent module. Sub-figure (b) depicts the operations in the residual block used in the convolution network (Here, N represents the number of filters in the residual block).

TensorFlow framework. The weights of the depth-wise and point-wise convolution operations are initialized using Glorot-uniform initialization [94], and biases are set to 0. The network is optimized using Adam Optimizer [92] for 100 epochs with an initial learning rate of 0.001 and a weight decay factor of 0.5 provided the validation loss has stopped improving for 5 continuous epochs.

It is worth emphasizing that once the Siamese architecture network is trained on the Omniglot dataset, its weights are frozen. These weights are then used to obtain the feature representation of the word fragments in the writer independent module.

### 4.3.3 Classification block

The features from both the writer-dependent and writer-independent modules are combined and then sent to a classification block, which includes a Global Average Pooling (GAP) layer, a dropout layer and a fully connected layer. Before reaching the classification module, the combined output undergoes a convolution operation, which serves as a bottleneck layer to obtain a lower-dimensional representation of the input feature. Such a configuration is intended to decrease the quantity of trainable parameters, thereby mitigating the risk of over-fitting in the overall network. In Table 4.1, we provide a snapshot on the number of parameters used to design the multi-stream CNN model.

## 4.4 Attention module

A convolution block in a CNN typically use fixed-sized kernels to process the input data. They process entire input images or sequences uniformly, regardless of the importance of different regions or features. This fixed size kernel in turn often leads to a fixed receptive field that may not adequately handle varying spatial contexts. An attention mechanism addresses this issue by allowing the network to selectively focus on specific elements within input data, thereby assigning varying levels of importance to different regions. This adaptability boosts the capability of the model to discern complex patterns and relationships, leading to a more effective feature learning.

In our proposed model, we investigate the impact of incorporating an attention mechanism by integrating the self-attention module introduced by Zhang et al. [95]. Unlike traditional attention mechanisms, which consider different parts of an input sequence relative to an external context, self-attention directly evaluates the inter-dependencies within the sequence itself. This is achieved through a sequence of mathematical transformations applied to the input sequence. This mechanism enables the model to compute attention scores for each pixel in a feature map in relation to every other pixels. Since these weights are computed over the entire height and width of the feature map, the receptive field is not limited to the size of a kernel anymore.

Let the feature maps from the previous convolution layer be denoted by $\mathbf{x} \in \mathbb{R}^{C \times N}$. Here

**Table 4.1:** An overview of the components of the CNN model in the context of the IAM dataset. The architecture of the convolution network up to the max pooling layer before the concatenation layer is identical for both the writer-dependent and writer-independent branches.

| Layer | Input size | Output size | Parameter |
|---|---|---|---|
| Input layer | 105x105x1 | 105x105x1 | - |
| Conv2D | 105x105x1 | 99x99x64 | filters=64, kernel size=7, Padding= 'valid' |
| Batch Normalization | 99x99x64 | 99x99x64 | momentum=0.99, epsilon=0.001 |
| Maxpooling2D | 99x99x64 | 49x49x64 | pooling size=2 |
| Residual Block | 49x49x64 | 49x49x128 | For convolution block filters=128, kernel size=3, padding='same'<br>For skip connection filters=128, kernel size=1, padding=same<br>For batch normalization momentum=0.99, epsilon=-0.001 |
| Maxpooling2D | 49x49x128 | 24x24x128 | pooling size=2 |
| Residual Block | 24x24x128 | 24x24x256 | For convolution block filters=256, kernel size=3, padding='same'<br>For skip connection filters=256, kernel size=1, padding= 'same'<br>For batch normalization momentum=0.99, epsilon=-0.001 |
| Maxpooling2D | 24x24x256 | 12x12x256 | pooling size=2 |
| Residual Block | 12x12x256 | 12x12x512 | For convolution block filters=512, kernel size=3, padding='same'<br>For skip connection filters=512, kernel size=1, padding='same'<br>For batch normalization momentum=0.99, epsilon=-0.001 |
| Maxpooling2D | 12x12x512 | 6x6x512 | pooling size=2 |
| Concatenate | (6x6x512),(6x6x512) | (6x6x1024) | - |
| Conv2D | 6x6x1024 | 6x6x512 | filters=512, kernel size=1, Padding= 'same' |
| Batch Normalization | 6x6x512 | 6x6x512 | momentum=0.99, epsilon=0.001 |
| Maxpooling2D | 6x6x512 | 3x3x512 | pooling size=2 |
| Global Average Pooling(GAP) | 3x3x512 | 512 | - |
| Dense + dropout | 512 | 1024 | - |
| Batch Normalization | 1024 | 1024 | momentum=0.99, epsilon=0.001 |
| Dense (Classification) | 1024 | 657 | Activation function=' Softmax' |

$N = H \times W$, where $H$ and $W$ represents the height and width of each of the feature map and $C$ represent the number of channels / feature maps. These undergo a transformation into two distinct feature spaces, represented by $\mathbf{f(x)} = \mathbf{W_f x}$ and $\mathbf{g(x)} = \mathbf{W_g x}$. The matrices $W_f$ and $W_g$ are of sizes $C^{'} \times C$ respectively. The resulting transformed representations serve as the query and key, respectively and aid in the calculation of attention weights as follows:

$$\beta_{j,i} = \frac{\exp(s_{ij})}{\sum_{i=1}^{N} \exp(s_{ij})} \tag{4.1}$$

where

$$s_{ij} = \mathbf{f(x_i)^T g(x_j)} \tag{4.2}$$

The value of $\beta_{j,i}$ essentially quantifies the importance of $j^{th}$ pixel position in the feature map relative to $i^{th}$ pixel position. Note that $\mathbf{x_i}$ and $\mathbf{x_j}$ represent slices of the input feature map along the $C$ channels at the $i^{th}$ and $j^{th}$ pixel position respectively. In other words, each of them has a size of $1 \times C$.

The attention weight undergoes further transformation upon being multiplied by a third feature space $\mathbf{h(x_i)} = \mathbf{W_h x_i}$, to obtain the final attention score vector $\mathbf{o}$ corresponding to the feature map entries of the convolution layer. It may be mentioned that the matrix $W_h$ is referred as value matrix and is also of size $C^{'} \times C$.

$$\mathbf{o} = (\mathbf{o_1}, \mathbf{o_2}, ..., \mathbf{o_j}, ..., \mathbf{o_N}) \in \mathbb{R}^{N \times C} \tag{4.3}$$

where

$$\mathbf{o_j} = v\Big(\sum_{i=1}^{N} \beta_{j,i} \mathbf{h(x_i)}\Big) \tag{4.4}$$

In the preceding Equation, $v$ acts as a mapping function employed to match the size of the output feature map with the input feature map being fed to the attention module. Further to this, the output of the attention layer is multiplied by a trainable scaling parameter ($\gamma$) and is re-integrated with the input feature map. The final output can thus be represented as:

$$\mathbf{y_i} = \gamma \mathbf{o_i} + \mathbf{x_i} \tag{4.5}$$

The introduction of the learning parameter $\gamma$ helps the network by regulating the influence of attention in the network [95]. Figure 4.6 depicts the pictorial overview of the attention module.

In our research, we have explored two configurations of the attention module. In the first configuration, the attention block is integrated separately into each of the writer-dependent and independent modules and thereafter the features are combined. In contrast to that, in the second configuration, the attention is integrated with the combined features of the writer-dependent and writer-independent modules. These configurations are illustrated in Figure 4.7. The size of the input feature map being fed to each of the configurations can be determined based on Table 4.1 while the size of $C^{'}$ in the attention network is determined experimentally. From Table 4.1, it may be inferred that the values of $N = 36$, $C = 512$, $H = 6$ and $W = 6$ are used for both configurations.

The evaluation of the above two configurations in the writer identification system is presented in Section 4.6.

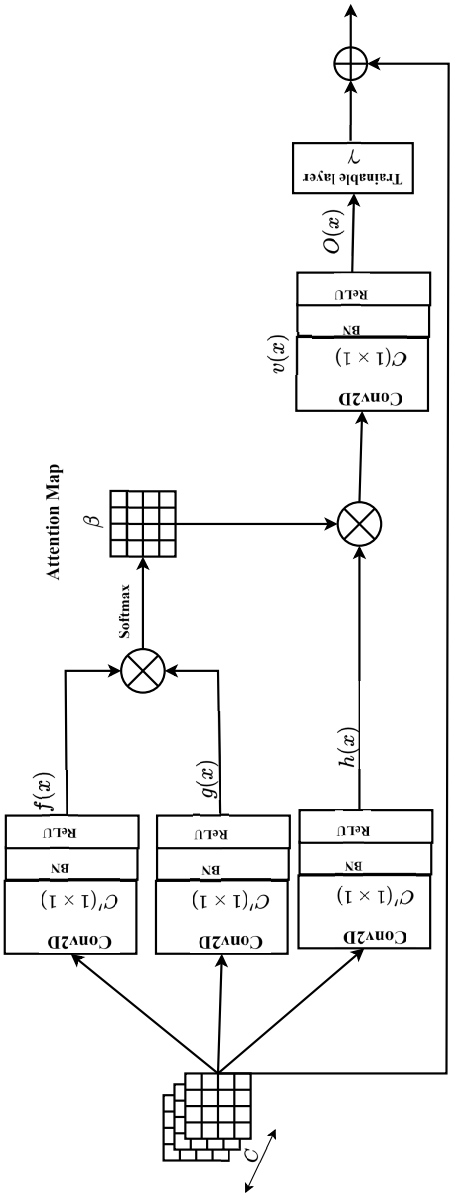



**Fig. 4.6:** Block diagram of the CNN-based Attention module. The entries after the convolution block represented by $C'(1 \times 1)$ and $C(1 \times 1)$ indicate the quantity and configuration of the convolution filters, with BN denoting the batch normalization block.

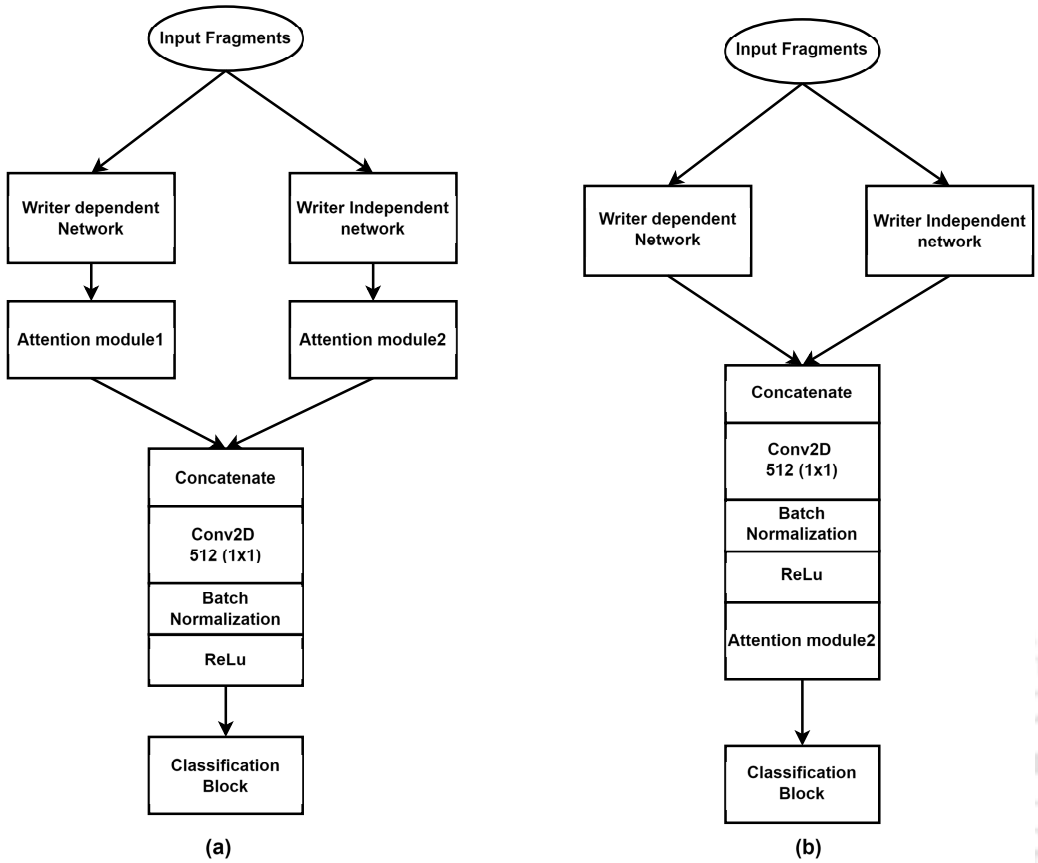



**Fig. 4.7:** Block diagram representing different configurations of attention module. Figure (a) the attention is incorporated separately to each of the writer dependent and independent networks and thereafter the features are combined. In Figure (b) the attention is incorporated with the output of the combined feature map.

## 4.5 CNN model training and testing process

The fragments extracted from the word image serves as input to the unified model, that incorporates both the writer-dependent and writer-independent modules. The writer-dependent module concentrates on capturing intricate details associated with a particular writer at the character and sub-character levels. During the training phase, the weights of this module are refined over several epochs by solely considering the fragments .

In contrast to the writer dependent module, the objective of the writer-independent block is to capture the general intricacies of the fragments as contributed from all writers being enrolled in the system. This is trained using transfer learning, where the weights of this module are frozen up to the second residual block. The pre-training is achieved by using the samples from

the Omniglot dataset (the details of which are elaborated in Section 4.3.2), The weights of the third residual block onwards are fine-tuned based on the input fragments.

With regards to the network, the back propagation of gradients during the training phase is based on the label smoothing cross-entropy loss function [96]. This loss converts the one hot encoded hard label $y_i$ into soft labels $\hat{y}_i$ as follows

$$\hat{y}_i = y_i(1-\epsilon) + (1-y_i)\frac{\epsilon}{K} = \begin{cases} 1-\epsilon, & i = target \\ \frac{\epsilon}{K}, & i \neq target \end{cases} \tag{4.6}$$

Here $K$ and $\epsilon$ denote the number of writers and the label smoothing factor respectively. Using this approach, the total loss is obtained by computing the cross-entropy loss between the predicted score in the range $(0, 1)$ with the adjusted ground truth label.

$$L_i = -\sum_{j}^{W} \hat{y}_{ij}.\log(p_{ij}) \tag{4.7}$$

where $\hat{y}_{ij}$ and $p_{ij}$ denote the modified ground truth and the predicted score respectively for the $i^{th}$ fragment of the $j^{th}$ writer. The overall training loss is subsequently computed by averaging across the $N_{TR}$ word fragments from the writers enrolled in the system.

$$L = \frac{\sum_{i}^{N_{TR}} L_i}{N_{TR}} \tag{4.8}$$

In the testing phase, given a word image $w$ yielding $N_T$ fragments, the overall score $P_i(w)$ associated with the $i^{th}$ writer is computed by averaging the responses corresponding to all the fragments as

$$P_i(w) = \frac{\sum_{k=1}^{N_T} p_{ki}}{N_T} \tag{4.9}$$

In the above Equation, $p_{ki}$ denotes the individual score of the $k^{th}$ fragment corresponding to the $i^{th}$ writer. The identity of the word image $w$ is assigned to the writer with the highest score.

$$\phi(w) = \underset{i \in \{1,\ldots,W\}}{argmax}\ P_i(w) \tag{4.10}$$

## 4.6 Experimental results and discussion

In this Section, we discuss in detail the set of experiments carried out on the word images from three databases namely IAM [75], CVL [77] and CERUG-EN [78] to validate the efficacy of our proposed algorithm. With regards to the training and evaluation protocol, we employ a similar strategy as discussed in Section 2.8. Each experiment is conducted through ten trial and the average writer identification rate is reported.

### 4.6.1 Implementation details

Our proposed network is implemented on TensorFlow framework. The batch size is set to 16 for training and validation. The network is optimized using the Adam optimizer [92] with an initial learning rate of 0.001. The weight parameters are decayed by a factor of 0.5 whenever the validation accuracy has stopped improving for 5 continuous epochs. The number of epochs for training the network is set to 150. The label smoothing factor $\epsilon$ in Equation 4.6 is set to 0.1.

With regards to the fragments of the word, they are resized to $105 \times 105$ while maintaining the aspect ratio and padding (if necessary) with white pixels. These modified image patches are then passed through the multi-stream CNN module to obtain scores that are then accumulated to get the identity of the writer,

### 4.6.2 Ablation study

In this sub-section, an ablation analysis is conducted to evaluate the effectiveness of different modules that are integrated into the proposal. In particular, the impact of adding various modules is examined by assessing its effects on the overall performance of the identification system.

As a first experiment in our study, a comparison is made between the baseline network constructed exclusively using the writer-dependent module and the combined model created by integrating the writer-independent module into the baseline configuration. Table 4.2 displays the Top 1 and Top 5 accuracy for these two set-ups on the three data sets. From the table, it can observed that integrating the writer independent features in the baseline configuration results

**Table 4.2:** Comparison of average identification rate (in %) on word level data. For brevity, we abbreviate writer dependent and writer independent module as WD and WI module respectively.

| Database | IAM | | CVL | | CERUG-EN | |
|---|---|---|---|---|---|---|
| | **Top 1** | **Top 5** | **Top 1** | **Top 5** | **Top 1** | **Top 5** |
| Baseline (WD module) | 92.87 | 97.13 | 91.00 | 96.95 | 97.47 | 99.68 |
| Multi-stream CNN (WD +WI module) | 93.43 | 97.35 | 91.71 | 97.08 | 97.62 | 99.82 |

in an increase in accuracy from 92.87% to 93.43 % and 91.00% to 91.71% for the IAM and CVL databases, respectively. Likewise, the improvement in the CERUG-EN database is from 97.47% to 97.62%. These results indicate that combining the writer-independent features with the baseline configuration leads to an enhancement in the overall performance of the proposed system by bolstering the capability of the model to represent the writer features more effectively.

In the second experiment, we examine how incorporating the attention module impacts the overall system performance. We evaluate two different configurations for integrating the attention module, as shown in Figure 4.7. In the first configuration, the attention module is integrated separately to each of the branches and thereafter the features are combined. Contrary to it , in the second configuration, the attention block is integrated into the combined features of the writer-independent and writer-dependent branches. These configurations are referred to as (a) and (b), respectively, in this study.

In each configuration, the size of the attention modules is systematically varied from $C/8$ to $C/2$, doubling it with each subsequent step. Here, $C$ denotes the number of convolution channels preceding the attention module [1]. The result of this experiment is tabulated in Table 4.3 and 4.4 respectively. Based on the the entries, it may be inferred that integrating the attention mechanism impacts the overall performance of the system. On comparing the baseline to the attention configuration (a), the identification accuracy shows notable improvements with attention head sizes of 64, 128, and 128, increasing from 92.87 %, 91.00 %, 97.47 % to 94.08 %, 92.19 % and 97.71 %, for the IAM, CVL, and CERUG-EN datasets respectively. Likewise, the configuration (b) results in best accuracy of 93.92 %, 92.25 %, 97.76 % , once again for the same number of channel sizes. Table 4.5 summarizes the results that are obtained from the different modules.

[1] For our work, $C = 512$ for both the configurations

**Table 4.3:** Performance evaluation of the attention mechanism (configuration (a)) on the multi-stream CNN network.

| # of Attention heads | IAM | | CVL | | CERUG-EN | |
|---|---|---|---|---|---|---|
| | Top 1 | Top 5 | Top 1 | Top 5 | Top 1 | Top 5 |
| 64 | 94.08 | 97.57 | 91.94 | 97.18 | 97.52 | 99.75 |
| 128 | 93.70 | 97.44 | 92.19 | 97.24 | 97.71 | 99.85 |
| 256 | 93.63 | 97.49 | 91.81 | 97.08 | 97.33 | 99.56 |

**Table 4.4:** Performance evaluation of the attention mechanism (configuration (b)) on the multi-stream CNN network

| No. of Attention heads | IAM | | CVL | | CERUG-EN | |
|---|---|---|---|---|---|---|
| | Top 1 | Top 5 | Top 1 | Top 5 | Top 1 | Top 5 |
| 64 | 93.92 | 97.45 | 91.42 | 97.05 | 97.57 | 99.78 |
| 128 | 93.46 | 97.37 | 92.25 | 97.58 | 97.76 | 99.90 |
| 256 | 93.70 | 97.40 | 91.68 | 97.18 | 97.74 | 99.85 |

Moving further, we would like to investigate on how the baseline CNN network would perform when trained on whole word images in place of its constituent fragments. It is interesting to see that the fragment level training of the network outperforms over that of whole word images (refer Table 4.6). This trend may be attributed to the fact that convolution networks trained on word images make decisions based on the holistic features, which may not fully capture all the relevant details of the writing style. Contrast to that, our fragment-based training approach treats each image as a collection of local patches, thereby possibly allowing the network to learn more relevant features. This results in a robust representation and in turn explains the effectiveness of the proposed identification process.

### 4.6.3 Statistical significance

In this sub-section, we establish that the results of the proposed multi-stream network (with and without the attention framework) are statistical significant when compared with the baseline model. The analysis is conducted with a significance level of 0.05 using the Student's t-test approach. Table 4.7 outlines the $p$-values obtained with regard to the algorithms across the three databases. A trend of low values for $p$ is empirically observed across the entries in the Table thus indicating the statistical significance of our proposed models.

**Table 4.5:** Comparison of average identification rate (in %) on the different datasets for the different network architectures proposed in this work.

| Database | IAM | | CVL | | CERUG-EN | |
|---|---|---|---|---|---|---|
| | Top 1 | Top 5 | Top 1 | Top 5 | Top 1 | Top 5 |
| Baseline (WD module) | 92.87 | 97.13 | 91.00 | 96.95 | 97.47 | 99.68 |
| Muli-stream CNN (WD+WI module) | 93.43 | 97.35 | 91.71 | 97.08 | 97.62 | 99.78 |
| Multi-stream CNN with attention configuration (a) | 94.08 | 97.57 | 92.19 | 97.24 | 97.71 | 99.85 |
| Multi-stream CNN with attention configuration (b) | 93.92 | 97.49 | 92.25 | 97.58 | 97.76 | 99.90 |

**Table 4.6:** Performance of writer identification system with convolution network trained on word and fragments.

| Dataset | Word | | Fragment | |
|---|---|---|---|---|
| | Top 1 | Top 5 | Top 1 | Top 5 |
| CVL | 87.76 | 94.60 | 91.00 | 96.95 |
| IAM | 86.08 | 92.94 | 92.87 | 97.13 |
| CERUG-EN | 65.76 | 87.52 | 97.47 | 99.85 |

### 4.6.4 Performance of proposed system with word images of different lengths

To evaluate the performance of the proposed model on word images with varying character lengths, we conducted an experiment using word images ranging from two to ten characters. Figure 4.8 illustrates the Top-1 accuracy of writer identification for different word lengths using samples from the IAM and CVL datasets. The accuracy of the word image containing up to three characters hovers around 77% and 83% respectively for the two databases. However, as the number of characters increases we observe an improvement of around 15%. This is due to the limited textual content in word images containing fewer than four characters, which hinders the ability of the model to accurately capture a writer's characteristics. Furthermore, across the two databases under consideration, the attention-based model outperforms the baseline (writer dependent module) and the multi-stream CNN network.

### 4.6.5 Performance comparison with prior end-to-end deep neural networks

In this sub-section, we compare the performance of our proposed end-to-end network with recent deep-learning networks for writer identification of handwritten word image. From Table

**Table 4.7:** Statistical significance of the multi-stream CNN architecture with and without the attention module over the baseline (writer dependent) method. For this experiment, we employ the Student's t-test.

| Database | Multi-stream CNN (without attention) | Multi-stream CNN (with attention) |
|---|---|---|
| CVL | $1.8 \times 10^{-2}$ | $1.1 \times 10^{-3}$ |
| IAM | $1.1 \times 10^{-3}$ | $1.8 \times 10^{-4}$ |
| CERUG-EN | $4.8 \times 10^{-2}$ | $3.5 \times 10^{-2}$ |

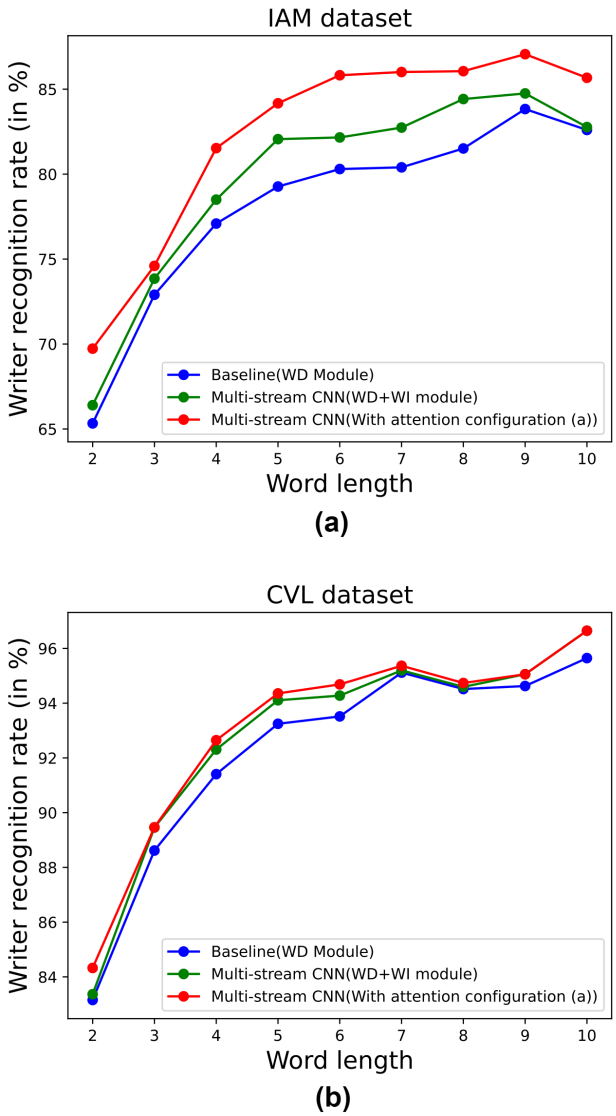



**Fig. 4.8:** Performance evaluation of different proposed methods on the word images with varying number of characters tested on (a) IAM, and (b) CVL Dataset.

4.8, it is evident from that our proposed method performs on par with state-of-the-art methods. Furthermore, for the CERUG-EN dataset, a significant improvement in accuracy is observed.

Furthermore to the above, our proposed model has fewer training parameters. Consequently,

**Table 4.8:** Performance comparison (in %) with existing Deep neural networks methods on word image data.

| Method | IAM | | CVL | | CERUG-EN | |
|---|---|---|---|---|---|---|
| | Top 1 | Top 5 | Top 1 | Top 5 | Top 1 | Top 5 |
| Deep-Adaptive [57] | 69.5 | 86.1 | 79.1 | 93.7 | - | - |
| WordImageNet [58] | 81.8 | 94.1 | 88.6 | 96.8 | 77.3 | 96.4 |
| Fragnet-64 [58] | 85.1 | 95.0 | 90.2 | 97.5 | 77.5 | 95.6 |
| Vertical GR-RNN(FGRR) [59] | 85.9 | 95.2 | 92.6 | 97.9 | 82.6 | 95.8 |
| Horizontal GR-RNN(FGRR) [59] | 86.1 | 95.0 | 92.4 | 97.8 | 83.2 | 96.2 |
| RSTC [60] | 90.7 | 96.6 | 92.7 | 97.9 | - | - |
| **Multi stream CNN with attention** | 93.9 | 97.4 | 92.3 | 97.6 | 97.7 | 99.9 |

**Table 4.9:** Computation efficiency of the proposed algorithm in terms of the number of parameters and FLOPs computed with respect to the IAM dataset

| Model | Flops (G) |
|---|---|
| FragNet [58] | 7.14 |
| GR-RNN [59] | 6.73 |
| Baseline (WD module) | 0.45 |
| Multi-stream CNN (WD +WI module) | 1.01 |
| Multi-stream CNN with attention configuration (a) | 1.03 |

it exhibits superior computational efficiency as compared to the models in [58] and [59]. In order to demonstrate this, we provide the count of floating-point operations (FLOPs) for each variant of our model on the IAM dataset. The results of the same are detailed in Table 4.9.

The comparison presented in both Table 4.8 and 4.9 clearly shows the superiority of our proposed model and its variant over prior end-to-end models both in terms of accuracy and computational efficiency.

## 4.7 Conclusion

This study explored a multi-channel convolution-based End-to-end network designed for offline text-independent writer identification systems focusing on word images. Our proposed network combines writer-specific local features with writer-independent global features to produce a strong representation of writer characteristics. Furthermore, the effect of integrating an attention mechanism on the overall performance of the system was investigated. Experimental results indicate the superiority of our proposed network in scenarios with limited samples when compared to prior networks trained on word images in terms of both accuracy and computa-

tional efficiency.

# 5

# Summary

## Contents

## 5.1 List of Contributions

In this section, we summarize the different contributions made in this thesis.

- In the first work of the thesis, we explore the merit of a framework for representing the features of a writer by exploiting novel cues from the feature maps of a CNN network.
- The main emphasis is placed on quantifying the relative information in each feature map of a convolution layer (referred to as saliency).
- Two strategies are proposed to summarize the feature map information by incorporating the saliency score.
- In the second work, a dissimilarity-based approach for predicting the identity of a handwritten sample is proposed by employing the Siamese architecture. The penultimate layer of the network is used as feature representation for each fragment of the word image.
- We explore a sparse-based model for representing the output feature representation of the Siamese network in a reduced dimensional space by employing the Sparse PCA framework.
- We formulate a novel divergence-based approach to assign saliency scores to each sparse component.
- We modify the traditional sparse-based representation by incorporating the obtained saliency values learned during the training phase.
- In the third work, an end-to-end multi-stream CNN network with attention is proposed.
- The effectiveness of the multi-stream CNN network in capturing the writer-specific writer-independent features is analyzed.
- The impact of a self attention module is also investigated in two configurations.

## 5.2 Possible avenues for future

We conclude this thesis by presenting possible research avenues that can be explored in future.

- In Chapter 2, the resultant feature representation associated with a particular convolu-

tion block are derived by linearly combining individual feature maps using the obtained saliency values. As part of future exploration, there is a possibility of exploring a non-linear weighting strategy to combine the feature map information. This concept can also be extended to combine the SVM classification scores at different layers of the convolution output.

- Chapter 3 primarily concentrated on developing a feature descriptor utilizing the concept of Sparse PCA on the Siamese representation . Many techniques exist to introduce sparsity in the principal components that were not explored in this study. Investigating the influence of these techniques and its effect on the overall system performance could offer promising prospects for future research endeavors.
- Chapter 4 focuses on training the convolution network using image fragment obtained from an input word image. Nevertheless, when considering word images, individual fragments may share spatial information with adjacent fragments. In this context, future work may entail assigning positional information to each fragment and examining its influence on the overall feature representation and identification accuracy.

Notwithstanding the above extensions, the present thesis is the first of its kind to propose strategies for writer identification that solely rely on the contributions of the fragments that make up a hand written word.

# Bibliography


[1] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] A. Riera, A. Soria-Frisch, M. Caparrini, I. Cester, and G. Ruffini, *Multimodal Physiological Biometrics Authentication*. John Wiley Sons, Ltd, 2009, ch. 18, pp. 461–482.

[3] A. K. Jain, A. Ross, and S. Prabhakar, "An Introduction to Biometric Recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, Jan 2004.

[4] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, 2nd ed. Springer Publishing Company, Incorporated, 2009.

[5] L. Li, X. Mu, S. Li, and H. Peng, "A Review of Face Recognition Technology," *IEEE Access*, vol. 8, pp. 139 110–139 120, 2020.

[6] Z. Sun, H. Zhang, T. Tan, and J. Wang, "Iris Image Classification Based on Hierarchical Visual Codebook," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, pp. 1120–1133, 2014.

[7] D. W. S. Alausa, E. Adetiba, J. A. Badejo, I. E. Davidson, O. Obiyemi, E. Buraimoh, A. Abayomi, and O. Oshin, "Contactless Palmprint Recognition System: A Survey," *IEEE Access*, vol. 10, pp. 132 483–132 505, 2022.

[8] R. Jahangir, Y. W. Teh, H. F. Nweke, G. Mujtaba, M. A. Al-Garadi, and I. Ali, "Speaker Identification through Artificial Intelligence Techniques: A comprehensive Review and Research Challenges," *Expert Systems with Applications*, vol. 171, p. 114591, 2021.

[9] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Offline Handwritten Signature Verification - Literature review," in *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, 2017, pp. 1–8.

[10] C. Adak, B. B. Chaudhuri, and M. Blumenstein, "An Empirical Study on Writer Identification and Verification From Intra-Variable Individual Handwriting," *IEEE Access*, vol. 7, pp. 24 738–24 758, 2019.

[11] P. S. Teh, A. Teoh, and S. Yue, "A survey of Keystroke Dynamics Biometrics," *The Scientific World Journal*, vol. 2013, 2013.

[12] E. Moritz, "Replicator Based Knowledge Representation and Spread Dynamics," in *1990 IEEE International Conference on Systems, Man, and Cybernetics Conference Proceedings*, 1990, pp. 256–259.

[13] M. Tapiador and J. A. Sigüenza, "Writer Identification Method Based on Forensic Knowledge," in *International Conference on Biometric Authentication*, 2004, pp. 555–561.

[14] M. Bulacu, R. van Koert, L. Schomaker, and T. van der Zant, "Layout Analysis of Handwritten Historical Documents for Searching the Archive of the Cabinet of the Dutch Queen," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, 2007, pp. 357–361.

[15] S. He, P. Sammara, J. Burgers, and L. Schomaker, "Towards Style-Based Dating of Historical Documents," in *2014 14th International Conference on Frontiers in Handwriting Recognition*, 2014, pp. 265–270.

[16] A. Gattal, C. Djeddi, I. Siddiqi, and S. Al-Maadeed, "Writer Identification on Historical Documents using Oriented Basic Image Features," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2018, pp. 369–373.

[17] M. Faundez-Zanuy, J. Fierrez, M. Ferrer, M. Diaz, R. Tolosana, and R. Plamondon, "Handwriting Biometrics: Applications and Future Trends in e-Security and e-Health," *Cognitive Computation*, vol. 12, 09 2020.

[18] T. Davis, "The Practice of Handwriting Identification," *Library*, vol. 8, pp. 251–276, 09 2007.

[19] S. Srihari, S.-H. Cha, H. Arora, and S. Lee, "Individuality of Handwriting," *Journal of forensic sciences*, vol. 47, pp. 856–72, 08 2002.

[20] I. Siddiqi and N. Vincent, "Text independent Writer Recognition Using Redundant Writing Patterns with Contour-based Orientation and Curvature Features," *Pattern Recognition*, vol. 43, no. 11, pp. 3853–3865, 2010.

[21] D. Bertolini, L. Soares de Oliveira, and E. Justino, "Texture-based Descriptors for Writer Identification and Verification," *Expert Systems with Applications*, vol. 40, p. 2069–2080, 05 2013.

[22] X. Wu, Y. Tang, and W. Bu, "Offline Text-Independent Writer Identification Based on Scale Invariant Feature Transform," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 3, pp. 526–536, 2014.

[23] F. A. Khan, F. Khelifi, M. A. Tahir, and A. Bouridane, "Dissimilarity Gaussian Mixture Models for Efficient Offline Handwritten Text-Independent Identification Using SIFT and RootSIFT Descriptors," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 289–303, 2019.

[24] V. Venugopal and S. Sundaram, "An Improved Online Writer Identification Framework using Codebook Descriptors," *Pattern Recognition*, vol. 78, pp. 318–330, 02 2018.

[25] R. Plamondon, G. Pirlo, E. Anquetil, C. Rémi, H.-L. Teulings, and M. Nakagawa, "Personal Digital Bodyguards for E-Security, e-Learning and e-Health: A Prospective Survey," *Pattern Recognition*, vol. 81, pp. 633–659, 2018.

[26] Y.-J. Xiong, Y. Lu, and P. S. Wang, "Off-line Text-independent Writer Recognition: A survey," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 31, no. 05, p. 1756008, 2017.

[27] M. Bulacu and L. Schomaker, "Text-Independent Writer Identification and Verification Using Textural and Allographic Features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 701–717, 2007.

[28] C. Djeddi, L.-S. Meslati, I. Siddiqi, A. Ennaji, H. E. Abed, and A. Gattal, "Evaluation of Texture Features for Offline Arabic Writer Identification," in *2014 11th IAPR International Workshop on Document Analysis Systems*, 2014, pp. 106–110.

[29] P. Singh, P. P. Roy, and B. Raman, "Writer Identification Using Texture Features: A Comparative Study," *Computers  Electrical Engineering*, vol. 71, pp. 1–12, 2018.

[30] A. Chahi, I. El-Khadiri, Y. El Merabet, Y. Ruichek, and R. Touahni, "Block Wise Local Binary Count for Off-Line Text-Independent Writer Identification," *Expert Systems with Applications*, vol. 93, pp. 1–14, 03 2018.

[31] A. Chahi, Y. El Merabet, Y. Ruichek, and R. Touahni, "Local Gradient Full-scale Transform Patterns Based Off-line Text-independent Writer Identification," *Applied Soft Computing*, vol. 92, p. 106277, 04 2020.

[32] Y. Hannad, I. Siddiqi, and Y. Elkettani, "Writer Identification Using Texture Descriptors of Handwritten Fragments," *Expert Systems with Applications*, vol. 47, 11 2015.

[33] Y. Hannad, I. Siddiqi, C. Djeddi, and Y. Elkettani, "Improving Arabic Writer Identification using Score Level Fusion of Textural Descriptors," *IET Biometrics*, 01 2019.

[34] Y. Kessentini, S. BenAbderrahim, and C. Djeddi, "Evidential Combination of SVM Classifiers for Writer Recognition," *Neurocomputing*, vol. 313, pp. 1–13, 2018.

[35] H. Said, T. Tan, and K. Baker, "Personal Identification Based on Handwriting," *Pattern Recognition*, vol. 33, no. 1, pp. 149–160, 2000.

[36] Z. He, X. You, and Y. Y. Tang, "Writer Identification of Chinese Handwriting Documents Using Hidden Markov Tree Model," *Pattern Recognition*, vol. 41, no. 4, pp. 1295–1307, 2008.

[37] A. Al-Dmour and R. A. Zitar, "Arabic Writer Identification Based on Hybrid Spectral-Statistical Measures," *J. Exp. Theor. Artif. Intell.*, vol. 19, no. 4, p. 307–332, Dec 2007.

[38] B. Helli and M. Ebrahimi Moghaddam, "A writer Identification Method Based on XGabor and LCS," *IEICE Electronic Express*, vol. 6, pp. 623–629, 05 2009.

[39] B. Helli and M. E. Moghaddam, "A Text-independent Persian Writer Identification Based on Feature Relation Graph (FRG)," *Pattern Recognition*, vol. 43, no. 6, pp. 2199–2209, 2010.

[40] Y. Hannad, I. Siddiqi, and M. E. Y. El Kettani, "Writer Identification Using Texture Descriptors of Handwritten Fragments," *Expert Systems with Applications*, vol. 47, pp. 14–22, 2016.

[41] C. Djeddi, I. Siddiqi, L. Souici-Meslati, and A. Ennaji, "Text-independent Writer Recognition Using Multi-script Handwritten Texts," *Pattern Recognition Letters*, vol. 34, no. 10, pp. 1196–1202, 2013.

[42] V. Christlein, D. Bernecker, F. Hönig, A. Maier, and E. Angelopoulou, "Writer Identification Using GMM Supervectors and Exemplar-SVMs," *Pattern Recognition*, vol. 63, pp. 258–267, 2017.

[43] S. Lai, Y. Zhu, and L. Jin, "Encoding Pathlet and SIFT Features With Bagged VLAD for Historical Writer Identification," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3553–3566, 2020.

[44] R. Arandjelović and A. Zisserman, "Three Things Everyone Should Know to Improve Object Retrieval," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2911–2918.

[45] L. Schomaker and M. Bulacu, "Automatic Writer Identification Using Connected-Component Contours and Edge-based Features of Uppercase Western Script," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 787–798, 2004.

[46] A. Durou, I. Aref, S. Al-Maadeed, A. Bouridane, and E. Benkhelifa, "Writer Identification Approach Based on Bag of Words with OBI Features," *Information Processing  Management*, vol. 56, no. 2, pp. 354–366, 2019.

[47] E. Khalifa, S. Al-maadeed, M. Tahir, A. Bouridane, and A. Jamshed, "Off-line Writer Identification Using an Ensemble of Grapheme Codebook Features," *Pattern Recognition Letters*, vol. 59, pp. 18–25, 2015.

[48] S. He, M. Wiering, and L. Schomaker, "Junction Detection in Handwritten Documents and its Application to Writer Identification," *Pattern Recognition*, vol. 48, no. 12, pp. 4036–4048, 2015.

[49] A. Bennour, C. Djeddi, A. Gattal, I. Siddiqi, and T. Mekhaznia, "Handwriting Based Writer Recognition Using Implicit Shape Codebook," *Forensic science international*, vol. 301, pp. 91–100, 2019.

[50] S. Fiel and R. Sablatnig, "Writer identification and retrieval using a convolutional neural network," in *International Conference on Computer Analysis of Images and Patterns*, 2015, pp. 26–37.

[51] V. Christlein, M. Gropp, S. Fiel, and A. Maier, "Unsupervised Feature Learning for Writer Identification and Writer Retrieval," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, 2017, pp. 991–997.

[52] A. Rehman, S. Naz, M. I. Razzak, and I. A. Hameed, "Automatic Visual Features for Writer Identification: A Deep Learning Approach," *IEEE Access*, vol. 7, pp. 17 149–17 157, 2019.

[53] A. Sulaiman, K. Omar, M. F. Nasrudin, and A. Arram, "Length Independent Writer Identification Based on the Fusion of Deep and Hand-crafted Descriptors," *IEEE Access*, vol. 7, pp. 91 772–91 784, 2019.

[54] P. Kumar and A. Sharma, "Segmentation-free Writer Identification Based on Convolutional Neural Network," *Computers  Electrical Engineering*, vol. 85, p. 106707, 2020.

[55] A. Semma, Y. Hannad, I. Siddiqi, C. Djeddi, and M. El Youssfi El Kettani, "Writer Identification Using Deep Learning with FAST Keypoints and Harris corner detector," *Expert Systems with Applications*, vol. 184, p. 115473, 2021.

[56] A. Chahi, Y. El-merabet, Y. Ruichek, and R. Touahni, "An Effective DeepWINet CNN Model for Off-Line Text-Independent Writer Identification," *Pattern Anal. Appl.*, vol. 26, no. 3, p. 1539–1556, Jul 2023.

[57] S. He and L. Schomaker, "Deep Adaptive Learning for Writer Identification Based on Single Handwritten Word Images," *Pattern Recognition*, vol. 88, pp. 64–74, 2019.

[58] 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.

[59] S. He and L. Schomaker, "GR-RNN: Global-context Residual Recurrent Neural Networks for Writer Identification," *Pattern Recognition*, vol. 117, p. 107975, 2021.

[60] P. Zhang, "RSTC: A New Residual Swin Transformer for Offline Word-Level Writer Identification," *IEEE Access*, vol. 10, pp. 57 452–57 460, 2022.

[61] H. Zou, T. Hastie, and R. Tibshirani, "Sparse Principal Component Analysis," *Journal of Computational and Graphical Statistics*, vol. 15, p. 2006, 2004.

[62] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[63] M. P. Beham and S. M. M. Roomi, "A Review of Face Recognition Methods," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 27, no. 04, p. 1356005, 2013.

[64] I. M. Hameed, S. H. Abdulhussain, and B. M. Mahmmod, "Content-based Image Retrieval: A Review of Recent Trends," *Cogent Engineering*, vol. 8, no. 1, p. 1927469, 2021.

[65] X. Min, G. Zhai, K. Gu, X. Yang, and X. Guan, "Objective Quality Evaluation of Dehazed Images," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 8, pp. 2879–2892, 2019.

[66] X. Min, G. Zhai, K. Gu, Y. Zhu, J. Zhou, G. Guo, X. Yang, X. Guan, and W. Zhang, "Quality Evaluation of Image Dehazing Methods Using Synthetic Hazy Images," *IEEE Transactions on Multimedia*, vol. 21, no. 9, pp. 2319–2333, 2019.

[67] X. Min, J. Zhou, G. Zhai, P. Le Callet, X. Yang, and X. Guan, "A Metric for Light Field Reconstruction, Compression, and Display Quality Evaluation," *IEEE Transactions on Image Processing*, vol. 29, pp. 3790–3804, 2020.

[68] B. Zhang, S. N. Srihari, and S. Lee, "Individuality of handwritten characters," in *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, vol. 3, 2003, pp. 1086–1086.

[69] G. X. Tan, C. Viard-Gaudin, and A. C. Kot, "Individuality of Alphabet Knowledge in Online Writer Identification," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 13, no. 2, p. 147–157, 2010.

[70] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "EMNIST: Extending MNIST to Handwritten Letters," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 2921–2926.

[71] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[72] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 886–893.

[73] X. Min, G. Zhai, J. Zhou, X.-P. Zhang, X. Yang, and X. Guan, "A Multimodal Saliency Model for Videos With High Audio-Visual Correspondence," *IEEE Transactions on Image Processing*, vol. 29, pp. 3805–3819, 2020.

[74] X. Min, G. Zhai, K. Gu, and X. Yang, "Fixation Prediction Through Multimodal Analysis," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 13, no. 1, pp. 1–23, 2016.

[75] U.-V. Marti and H. Bunke, "The IAM-database: An English Sentence Database for Offline Handwriting Recognition," *International Journal on Document Analysis and Recognition*, vol. 5, pp. 39–46, 11 2002.

[76] C. J. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[77] F. Kleber, S. Fiel, M. Diem, and R. Sablatnig, "CVL-DataBase: An Off-Line Database for Writer Retrieval, Writer Identification and Word Spotting," in *2013 12th International Conference on Document Analysis and Recognition*, 2013, pp. 560–564.

[78] S. He, M. Wiering, and L. Schomaker, "Junction Detection in Handwritten Documents and its Application to Writer Identification," *Pattern Recognition*, vol. 48, no. 12, pp. 4036–4048, 2015.

[79] A. Brink, J. Smit, M. Bulacu, and L. Schomaker, "Writer Identification Using directional Ink-trace Width Measurements," *Pattern Recognition*, vol. 45, pp. 162–171, 01 2012.

[80] H. Sheng and L. Schomaker, "Beyond OCR: Multi-faceted Understanding of Handwritten Document Characteristics," *Pattern Recognition*, vol. 63, pp. 321–333, 03 2017.

[81] S. He and L. Schomaker, "Writer Identification Using Curvature-free Features," *Pattern Recognition*, vol. 63, pp. 451–464, 2017.

[82] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[83] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," vol. 39, 06 2015.

[84] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep Learning-based Text Classification: A Comprehensive Review," *ACM Comput. Surv.*, vol. 54, no. 3, Apr. 2021.

[85] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese Neural Networks for One-shot Image Recognition," in *ICML deep learning workshop*, vol. 2, no. 1, 2015.

[86] T. B. Viana, V. L. Souza, A. L. Oliveira, R. M. Cruz, and R. Sabourin, "A Multi-task Approach for Contrastive Learning of Handwritten Signature Feature Representations," *Expert Systems with Applications*, p. 119589, 05 2023.

[87] X. Liu, W. Gao, R. Li, Y. Xiong, X. Tang, and S. Chen, "One Shot Ancient Character Recognition With Siamese Similarity Network," *Scientific Reports*, vol. 12, no. 1, p. 14820, 09 2022.

[88] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level Concept Learning Through Probabilistic Program Induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.

[89] E. Hoffer and N. Ailon, "Deep Metric Learning Using Triplet Network," in *SIMBAD*, 12 2015.

[90] J. Lin, "Divergence Measures Based on The Shannon Entropy," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 145–151, 1991.

[91] V. Christlein, D. Bernecker, and E. Angelopoulou, "Writer identification using VLAD encoded contour-Zernike moments," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 906–910.

[92] D. P. Kingma and J. Ba, "Adam: A method for Stochastic Optimization," *International Conference on Learning Representations*, 12 2014.

[93] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[94] X. Glorot and Y. Bengio, "Understanding the Difficulty of Training Deep Feedforward Neural Networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[95] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-Attention Generative Adversarial Networks," in *International conference on machine learning*, 2019, pp. 7354–7363.

[96] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

# LIST OF PUBLICATIONS

## Journal Publications

1. Vineet Kumar and S. Sundaram, "Utilization of Information from CNN Feature maps for Offline Word-level Writer Identification," *Expert Systems with Applications*, vol. 238, p. 121709, 2024
2. Vineet Kumar and Suresh Sundaram, "Siamese-based Offline Word Level Writer Identification in Reduced Subspace," *Engineering Applications of Artificial Intelligence*, vol. 130, p. 107720, 2024.