Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.035279
Article

Supervised Feature Learning for Offline Writer Identification Using VLAD and Double Power Normalization

Dawei Liang[1,2,4], Meng Wu[1,*] and Yan Hu[3]

[1]College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, 210023, China
[2]Department of Computer Information and Cyber Security, Jiangsu Police Institute, Nanjing, 210031, China
[3]JinCheng College, Nanjing University of Aeronautics and Astronautics, Nanjing, 211156, China
[4]Engineering Research Center of Electronic Data Forensics Analysis, Nanjing, 210031, Jiangsu Province, China
*Corresponding Author: Meng Wu. Email: wum@njupt.edu.cn
Received: 15 August 2022; Accepted: 08 February 2023; Published: 09 June 2023

Abstract: As an indispensable part of identity authentication, offline writer identification plays a notable role in biology, forensics, and historical document analysis. However, identifying handwriting efficiently, stably, and quickly is still challenging due to the method of extracting and processing handwriting features. In this paper, we propose an efficient system to identify writers through handwritten images, which integrates local and global features from similar handwritten images. The local features are modeled by effective aggregate processing, and global features are extracted through transfer learning. Specifically, the proposed system employs a pre-trained Residual Network to mine the relationship between large image sets and specific handwritten images, while the vector of locally aggregated descriptors with double power normalization is employed in aggregating local and global features. Moreover, handwritten image segmentation, preprocessing, enhancement, optimization of neural network architecture, and normalization for local and global features are exploited, significantly improving system performance. The proposed system is evaluated on Computer Vision Lab (CVL) datasets and the International Conference on Document Analysis and Recognition (ICDAR) 2013 datasets. The results show that it represents good generalizability and achieves state-of-the-art performance. Furthermore, the system performs better when training complete handwriting patches with the normalization method. The experimental result indicates that it's significant to segment handwriting reasonably while dealing with handwriting overlap, which reduces visual burstiness.

Keywords: Writer identification; power normalization; vector of locally aggregated descriptors; feature extraction

1 Introduction

Due to the rapid growth in the interactive Internet of Things (IoT), existing traditional applications cannot meet real-time response requirements and low latency [1]. In response to this situation, a new method known as mobile edge computing (MEC) is developing, enabling highly demanding applications [2]. Moreover, biometric technology for securing commercial services has attracted more and more attention. Compared to biometric features such as face, fingerprint, or iris, handwriting represents behavior, influenced by external factors such as education level, age, writing tools, etc. The typical applications mainly lie in biometrics, forensics, historical document analysis, and other fields in recent decades. Based on this, writer identification (WI) determines the writer of handwriting by comparing features extracted from the handwriting samples. For the WI system, it is denominated as author verification which is just determining the identity of two samples. Otherwise, if it is necessary to search for the most likely query in the database, the task is denoted as writer retrieval. These tasks are similar, although the latter needs to make a similarity list in the classification link.

According to the form of handwriting processing, WI could be classified as online and offline [3]. The online WI studies dynamic information such as stroke order, writing speed, pressing force, etc., while the latter processes the structure, strokes, and statistics from the handwriting image. Since dynamic handwriting data require special electronic devices, in contrast, handwriting images are easier to obtain and process, such as historical document manuscripts. Therefore, the research on offline handwriting identification is more extensive. Currently, extracting discriminative features from handwritten images is more challenging. In WI, the content of the written text also affects the identification. If handwritten text is specified, specifically, handwritten content is consistent, the identification process is content dependent [4]. Otherwise, if there is no requirement for written content, it is defined that the identification process is content-independent [5].

For offline content independent writer identification, handwriting contains relatively few characteristics of the writer. Therefore, researchers apply various methods to achieve classification from simple handwriting images. Generally, feature extraction methods can be classified into codebook-free and codebook-based. The former method calculates the global feature descriptor directly from the handwriting image, while the latter does not. Generally, in codebook-free methods, researchers study the width of marks, the angle of stroke direction [6], etc., which are transformed into probability density functions. In some literature, this method is also called the texture-based method or statistical method. In other words, a handwriting sample is regarded as a special texture, which is defined as variations of grayscales with particular patterns.

The codebook-based method first extracts features from local handwriting and then forms a global feature descriptor through a background model that serves as a codebook. According to different local descriptors, this category can be called the allograph-based approach or shape-based approach. Typically, individual codebooks were applied using K-means [7] or the Gaussian mixture model (GMM) [8]. However, due to the simplified correspondence and distance calculation, the universal background model is more common in WI than individual codebooks.

In recent years, with the improvement of computer calculation ability, machine learning has a great development in artificial intelligence. Meanwhile, convolutional neural networks (CNN) also perform well in WI. The difference between CNN and the codebook-based method lies in the local features extracted. The former extracts biometrics features directly from the original images, while the latter requires an expert's subjective experience in language. Therefore, CNN only extracts local features, and cannot achieve end-to-end writer identification. After that, WI aggregates local features into global features, denoted as encoding. Subsequently, the system calculates the global features and

then compares them with the dictionary. In encoding, different models affect the quality of the final result. The Gaussian mixture model, the vector of locally aggregated descriptors (VLAD), and Fisher vectors (FV) have been constantly discussed in recent years. In addition, the combination of methods in CNN and computer vision (CV) has become a new hotspot in WI development [9–11].

However, handwriting feature extraction based on the neural network could be improved by further processing before and after training, for example., the size of the handwriting patch, preprocessing methods for handwriting samples and normalization of the extracted features affect the final result. In this paper, we apply novel techniques to achieve efficient offline WI. First, we employ a CNN by transfer learning, extracting discriminative features. To begin, transfer learning trains a CNN on image datasets. Then, we fine-tune the trained CNN with the target dataset. For image information and handwriting information, the general features are similar, hence WI can get better feature representation by fine-tuning. Subsequently, we employ the VLAD method with signed square rooting (SSR) to form a global feature, which represents the characteristics. When SSR is applied to the representation, it is efficient in reducing visual burstiness. Finally, the writer is identified by comparing the distance between the unknown writing and the samples.

We organize the content as below. We give a summary of the relevant research about WI in Section 2. Section 3 represents our work on handwriting preprocess, enhancement, feature extraction, and identification. In Section 4, we introduce evaluation criteria and the databases, reporting the evaluation. Finally, Section 5 is the conclusion.

2 Related Work

Du to WI cannot achieve end-to-end handwriting identification, feature extraction has become the hotspot of WI. On the basis of this, we mainly study text-independent WI. According to the analysis methods, WI can be divided into texture, allograph, and deep learning.

At the beginning of the study, texture analysis was used to extract features. Said et al. [12] proposed a texture analysis method employing multi-channel two-dimensional Gabor filtering technology. The standard deviation and mean of the filtered image are extracted as global features. Schomaker presented writer features with Δ-n Hinge, which evolved from the Hinge feature [13]. Meantime, local phase quantization (LPQ) and local binary patterns (LBP) [14] are applied in the page-level handwriting description to describe the features of local texture well.

Besides the contours and statistical information, local edges, key points, and spatial distribution also reflect the handwriting style. Some effective local texture descriptions are suitable for WI, such as speeded-up robust feature (SURF) and scale-invariant feature transform (SIFT). Fiel et al. [15] employed SIFT descriptor to extract local features in the training dataset and then calculated the GMM. After that, GMM is used to get the global features through the verification dataset. Finally, the cosine distance is calculated to identify handwritten documents. They achieved state-of-the-art (SOTA) in CVL and ICDAR 2011 datasets.

With the wide application of deep learning technology, researchers applied convolutional neural networks to WI. It achieved better results than the traditional way. CNN is generally employed to extract local or global features in the form of supervised, unsupervised, and semi-supervised. Fiel et al. [16] first employed CaffeNet in 2015. After that, they identify the writer through the Euclidean distance. Also in 2015, Christlein et al. [17] employed a six-layer CNN to extract local descriptions. Furthermore, the local descriptions were aggregated through zero phase component analysis (ZCA) whitening and GMM. On ICDAR2013 and CVL datasets, they achieved the 0.989 and

0.994 TOP-1 criteria, respectively. Then Christlein et al. [18] used the SIFT feature clustering method to obtain proxy labels of handwriting samples and train CNN to achieve unsupervised learning. In [19], the original ResNet was adjusted by conjugating deep residual networks. They evaluated the new architecture on four public datasets and achieved consistent results.

In contrast to previous work, a novel ResNet method is proposed with VLAD and SSR. We assume that the proposed pipeline could preserve the handwriting features between patches while eliminating the burst of visual elements to improve feature learning, and enhance the robustness of identification.

3 Methodology

The proposed WI consists of three modules: preprocessing, feature extraction, and encoding (see Fig. 1). After obtaining the features, we classify the samples.

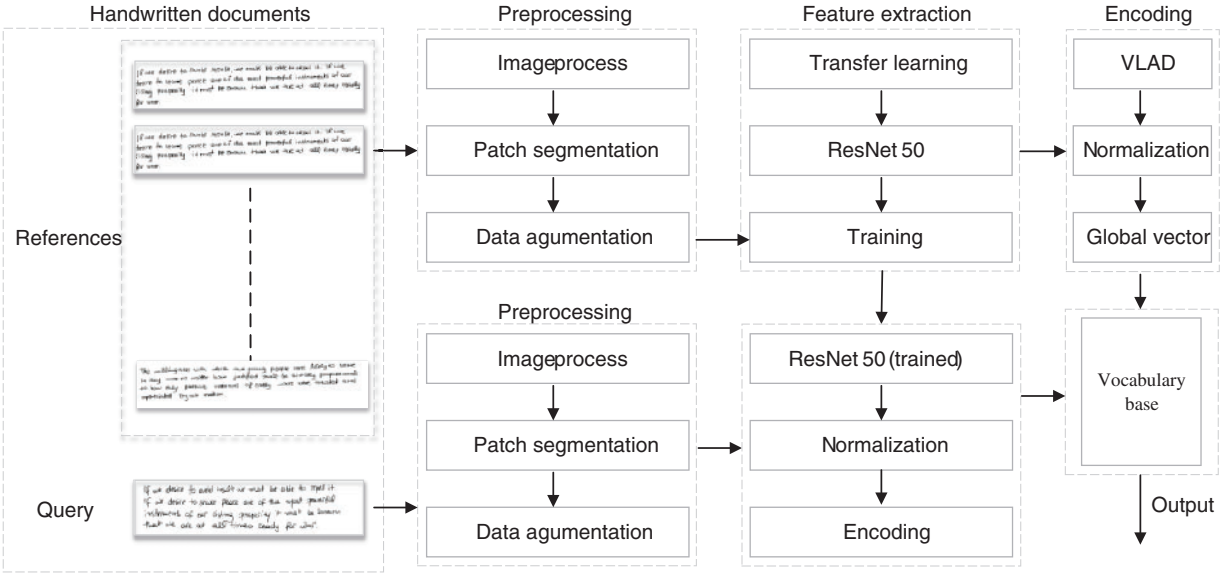

Figure 1: The proposed WI consists of two parallel lines: base set (upper line) and query classification (bottom line). The common parts include preprocessing, feature extraction, and encoding. Among them, the feature extraction of the base set is realized by fine-tuning of transfer learning, and the parameters of query architecture are updated. After encoding, the features extracted from the query are compared with the dictionary and the results are output

Compared with the common classification task, the current writer identification has some problems, i.e., it is unable to achieve an end-to-end process through CNN. First of all, due to the large parameters of CNN, the cost-intensive blocks the realization of deep learning writer identification for pages. Thus, most deep learning-based methods utilize image patches to form discriminative handwriting features to identify the writer. Second, each writer in the current dataset contributes 1 to 5 pages, which leads to the lack of training samples. The above-mentioned problem could be partly solved by dividing pages into image patches. The task of classification or retrieval is conducted following the feature extraction.

3.1 Dataset

Two classical databases are employed in this paper. The first is ICDAR 2013 [20] standard database. There are 700 handwritten pages of content in English and German provided by 350 writers. And another one is CVL [21] standard database. In CVL, 300 writers supply handwritten pages in English and German. We conduct the fine-tuning of CNN on ICDAR 2013 and evaluate it on the CVL.

3.2 Preprocessing

Preprocessing is the first step in the proposed pipeline. In preprocessing, page-level handwritten images are reduced by image processing, converted into patches, and fed to CNN.

3.2.1 Image Process

In image preprocessing, the first step is rotation correction, due to the incorrect scanning mode and the writing without reference lines. The skew in handwriting images will bring problems to segmentation, thus we adopt Probabilistic Hough Transform proposed in [22] to detect line, skew, and correct. Suddenly, OSTU's method is applied for binarization, improving the visual quality and eliminating the background influence.

3.2.2 Patch Generation

The processed handwriting needs to be segmented to create local features for the following CNN. Specifically, we remove the edge of the images first, maintaining the image height to width ratio, then the appropriate patch is generated by the sliding window. We optimize the sliding window, balancing the relationship between the processing speed and the integrities of the handwriting features. In training, we ensure less loss of handwriting features through reduce the window. Simultaneously, the patch containing less writing information is discarded. Sample patches are exhibited in Fig. 2.

Figure 2: Sample patches generated from standard database

3.2.3 Data Augmentation

For CNN, the handwriting involved in the training set affects the generalization accuracy. Therefore, data enhancement technology is applied. Common image enhancement techniques include rotation, mirroring, inversion, partial enlargement, etc. In this paper, considering the characteristics of handwriting, we utilize contour, sharpening, and inversion. These ensure the aspect ratio of the images and do not change characteristics. After that, we divided the patches for training, testing, and validation. The preprocessed patches would be fed into the CNN, while the size is adjusted according to the input.

3.3 Feature Extraction

3.3.1 ResNet

Researchers have built many classic deep learning architectures, such as GoogleNet, AlexNet, VGG, ResNet, etc. He et al. [23] first proposed the application of ResNet in WI and achieved radical results. The advantage lies in that the residual unit solves the degradation, and drives deeper. Compared to other deeper networks, we utilize adjusted ResNet50 (see Table 1) to extract handwriting characteristics while balancing the computational cost and performance.

Table 1: The adjusted architecture

Layer	Output	Remarks
Conv1	112×112	7×7, 64, stride 2
Conv2-x1	56×56	3×3 max pool, stride 2
Conv2-x2	56×56	$\begin{bmatrix} 1 \times 1, & 64 \\ 3 \times 3, & 64 \\ 1 \times 1, & 256 \end{bmatrix} \times 3$
Conv3-x	28×28	$\begin{bmatrix} 1 \times 1, & 128 \\ 3 \times 3, & 128 \\ 1 \times 1, & 256 \end{bmatrix} \times 4$
Conv4-x	14×14	$\begin{bmatrix} 1 \times 1, & 256 \\ 3 \times 3, & 256 \\ 1 \times 1, & 1024 \end{bmatrix} \times 6$
Conv5-x	7×7	$\begin{bmatrix} 1 \times 1, & 512 \\ 3 \times 3, & 512 \\ 1 \times 1, & 2048 \end{bmatrix} \times 3$
		Global pooling
		1000-d FC
	1×1	Drop out
		Softmax (according to the dataset)

Compared to the original architecture, the penultimate pooling layer is replaced by a global average one. Then, the last layer is adjusted following the writers. Thus, image patches belong to K writers, where K represents the writers. The value of K is 100 on ICDAR2013 and 50 on CVL, respectively. When training or extracting features, the final full connected layer should be adjusted according to the dataset.

3.3.2 Transfer Learning

In the previous image processing, although after data enhancement, the handwriting data are still insufficient. Simultaneously, due to the similarity of visual features and handwriting image features, the WI could be pre-trained by transfer learning through the ImageNet dataset. Subsequently, the advanced features are extracted by fine-tuning through the target dataset.

For transfer learning, there are two working modes. The first scenario is that the neural net freezes the parameters in the training process, whereas the classifier is initialized and trained randomly. The other is fine-tuning, in which the neural network parameters are loaded into the pre-trained weights and fine-tuned together with the classifier by continuing backpropagation. The above two modes are selected according to the correlation of the target dataset and features.

3.3.3 Normalization of Local Descriptors

The local features of handwriting data are acquired through feature extraction, and the normalization method improves the performance of the local features. In the traditional method, local features are normalized with L2 normalization, reducing the impact of larger bin values without increasing processing or storage requirements. Thus, we employ Hellinger normalization to improve system performance, where each element x_i of the activation feature X is normalized. See Eq. (1). The *sign( )* denotes the sign of the scalar.

$$\hat{x}_i = sign\,(x_i)\,\sqrt{|x_i|}, \forall x_i \in X \tag{1}$$

3.4 Encoding

3.4.1 Vector of Locally Aggregated Descriptors

Global features of handwriting could be obtained by integrating multiple local features. This process is called encoding. Typical VLAD yields a compact representation of local features. Thus, we utilize it with local features extracted by deep learning. For VLAD encoding, the K-means algorithm is employed to calculate the vocabulary. After that, all the vectors match with the nearest centroid. Subsequently, we accumulate the residuals among the local features and corresponding cluster centroids. See Eq. (2).

$$v_k = \sum_{f_S:NN(f_S)=c_k} (f_S - c_k) \tag{2}$$

In the equation, c_k refers to the centroid of the kth cluster. *NN(fs)* is a function of *fs,* accumulating the nearest neighbor. Subsequently, we concatenate a global feature through V_k for page V. See Eq. (3).

$$v_{VLAD} = (v_1^T, v_2^T, \ldots, v_K^T)^T \tag{3}$$

Unlike other coding methods, VLAD is based on codebooks with fixed similarity. In the case of different center clusters, that is, when new samples are mixed into the dataset, the descriptors show completely different similarities, affecting the global results. Thus, we only train in a single training set and verify in multiple verification sets.

3.4.2 Normalization of Global Descriptors

After VLAD, we deal with visual burst, correlation, and variability between sessions of global descriptors by normalization. Power normalization could effectively deal with visual contour bursts. We normalize every element *xi* in the global vector X. See Eq. (4).

$$\hat{x}_i = sign\,(x_i)\,|x_i|^n, \forall x_i \in X \tag{4}$$

The *sign( )* denotes the sign of the scalar. When the power n sets to 0.5, the power normalization equates to Hellinger normalization employed for local features described above, which is also denoted as signed square rooting.

3.5 Classification

After the training, we conduct the classification. In this process, we load the trained parameters into the neural network and remove the last layer. The preprocessed query is fed into the system for forward propagation. Then, the exportation of the FC layer is considered as the local feature representation. Subsequently, WI integrates the global features of the whole test set and query. The system calculates the L2 distance between the reference and the query to determine the writer.

4 Experiments

Before model training, we first determine the patch scales of the handwriting samples collected. Second, we compare the popular neural network frameworks and determine the structure used. Moreover, we use several normalization methods to process the extracted handwriting features. In model training, we explore the effectiveness of data preprocessing methods, compare the efficiency of various coding methods, and verify the normalization method of local features generated after coding.

4.1 Evaluation Metrics

To evaluate the model, we utilize the TOP-N index. This process is slightly different from the above classification. First, all vectors in the dataset are listed according to the distance from the query. At this point, we evaluate the system with two criteria: hard and soft standards. Soft standard means that the result is yes if there is more than one matched sample. More strictly, the hard one means that the identified value is if all the samples are matched. Subsequently, the mean of corrected values for all writers is calculated as a percentage. In this paper, taking into account the datasets, we employ soft TOP1, soft TOP5, soft TOP10, hard TOP2 and hard TOP3 to estimate the system. Soft TOP1 and hard TOP1 are the same concepts.

Mean average precision (mAP) is also employed. In the TOP-N index mentioned above, we list the distance of the query and the reference. To obtain the mAP, we first calculate *the* $AP(i)$, which means the average precision of the ith sample. See Eq. (5).

$$AP(i) = \frac{\sum_{k=1}^{n} P(k)\, r(k)}{M} \tag{5}$$

In Eq. (5), we employ $P(k)$ as the precision of rank k, while M is the number of samples matched. Besides, $r(k)$ is a function that is one when the kth handwriting is matched, zero otherwise. Finally, we calculate mAP, the mean value of AP. See Eq. (6).

$$mAP = \frac{\sum_{i=1}^{N} AP(i)}{N} \tag{6}$$

4.2 CNN Activation Features

4.2.1 Patch Scales

To get the best extracting effect, we conduct experiments on several architectures. The experiments are verified by an Intel Core i7-10875H with RTX2060. In terms of the implementation framework, we use Pytorch-lightning 0.9.0 with Pytorch 1.6.0 as the backend. The process is accelerated by the Compute Unified Device Architecture (CUDA).

We feed the image patches to CNN, which are extracted from the ICDAR2013 standard dataset. Additionally, image patches with less handwriting information are discarded. We extract about 450000, 260000, 150000, and 80000 image patches on the four scales of 32, 64, 128, and 256, respectively. We divide the patches for training and verification by 200:1. Meanwhile, the scale of the extracted image patch is optimized (see Table 2).

Table 2: Comparison of various patch scales (in %)

Scales	S-TOP1	*mAP*
32	70.7	50.4
64	77.5	60.8
128	83.3	66.7
256	**85.6**	**71.8**

The table shows that the image patches based on the 256-scale perform well in all indicators, indicating that the image patches contain more complete handwriting information on this scale, i.e., the word-based scale. At the same time, the training time of the model is relatively short due to the fewer image patches separated from the dataset in 256 sizes. In contrast, a larger image patch is a horizontal scan of the handwriting page, which is not suitable in consideration of the data volume and computing cost. Therefore, the experiments below are based on 256-scale image patches.

4.2.2 CNN Architecture

We have assessed various network models. The parameters of the networks are transferred from ImageNet training, and subsequently optimized using the stochastic gradient descent (SGD) method. The hyper-parameters operated are exhibited in Table 3.

Table 3: The hyper-parameters

Parameter	Value
Momentum	0.9
Learning rate	0.0001
Epochs	20
Batch size	64

Several popular CNN architectures were employed. We feed the image patches into CNNs after preprocessing. Subsequently, backpropagation is performed. The accuracy is shown in Table 4.

Table 4 shows that ResNet50, i.e., the deeper CNN has better accuracy, which is beneficial for retrieval. Meanwhile, the gap between VGG and GoogleNet with the highest accuracy rate is not remarkable. The reason is that the extraction of handwriting features has been sufficient, which limits the performance of the deeper network. In addition, the accuracy of the table is not acceptable. This is so because the handwriting information contained in the image patches is only the contour, which is not sufficient compared with the scene or object in ImageNet. After extracting local features, encoding is conducted to obtain global features.

Table 4: Accuracy of cross validation from different CNN architectures (in %)

Architecture	Accuracy
LeNet	52.8
AlexNet	60.1
VGG19	69.2
GoogleNet V3	72.4
ResNet50	**75.9**

4.2.3 SSR for Local Features

In the model, the normalization of local features also has a positive effect on the results. We evaluate the Hysteresis threshold and L2 normalization, which are the standard normalization techniques suggested. In addition, we investigate Hellinger normalization, also known as SSR (see Table 5).

Table 5: Evaluation of various normalization methods (in %)

Method	S-TOP1	*mAP*
Baseline	87.1	74.0
L2	88.2	75.9
Hysteresis	88.0	75.3
SSR	**88.9**	**76.7**

The outcome shows that the local feature normalization improves the recognition rate to some extent compared to the baseline. Among them, SSR had the best effect, increasing by 2.8%. In addition, normalization improves the efficiency of subsequent encoding.

4.2.4 Image Preprocessing and Augmentation

The preprocessing of handwriting images mainly consists of rotation correction and binarization. In the process of image patch segmentation, if the handwriting is conglutinated or the cutting of context is not accurate, feature learning will be affected. Therefore, it is necessary to correct the skew handwriting. Meanwhile, we employ Ostu's method to eliminate the impact of background.

To enhance the generalizability of CNN, in preprocessing step, we expand the patches by data augmentation. To preserve handwriting information, traditional enhancement methods such as flip, rotation, cropping, and deformation scaling are discarded. Moreover, contour, sharpening, and inversion are utilized (see Table 6).

Table 6 shows that the generalization ability of CNN has been improved through image preprocessing and data enhancement. The baseline is improved by 4.2% and 7.6%, respectively, on S-TOP1. However, *mAP* has poor progress, only 0.8%, and 1.7%. The reason is that the slanting of handwriting is also one kind of characteristic. Eliminating distortion is equivalent to reducing the features extracted. Otherwise, contour, sharpening, and conversion do not change the inherent

Table 6: Evaluation of image preprocessing and data augmentation employed (in %)

Method	S-TOP1	*mAP*
Baseline	88.9	76.7
Preprocessed	93.1	77.5
Augmented	96.5	78.4
Combination	**99.0**	**83.7**

characteristics of handwriting but increase the amount of data from the perspective and background of handwriting, which is beneficial to feature extraction.

4.3 Experiments on Encoding

There are many encoding methods in the literature. Currently, we evaluate FV [24], I-vector [25], and VLAD [26]. The role of SSR and L2 normalization after encoding is also discussed. We utilize standard settings in FV coding, and the global features are power normalized. For I-vector coding, we choose the dimension in training, that is, 100 components for every feature (see Table 7).

Table 7: Evaluation of various encoding methods (in %)

Method	S-TOP1	*mAP*
FV	85.5	76.7
I-vector	93.4	77.5
VLAD	96.5	80.4
VLAD + L2	97.0	81.5
VLAD + SSR	**99.0**	**83.7**

Table 7 shows the results of different encoding methods. It draws a clear picture that VLAD-based encoding generally delivers better results compared with the other methods. Moreover, through the final global descriptors, SSR has a certain improvement compared to the raw and L2 normalization. The best result is now achieved by the VLAD-based encoding with SSR.

4.4 Comparison with SOTA

4.4.1 ICDAR2013 Dataset

The handwriting in ICDAR 2013 is a great challenge to the model due to writing in English and Greek. We employ the training set to fit the parameters while not retraining by other datasets. Table 8 shows the evaluation that the soft criterion achieves a better result. This is so because the hard criterion has higher requirements for the generalization of system models. Besides, the results achieve a quite good level, compared with other models, with just a 7.9% lag in hard TOP3, and a 6.4% lag in *mAP*, respectively. Significantly, the method in [27] did better in *mAP*, due to the mixed training set in English and Greek handwriting from 100 volunteers in the International Conference on Frontiers in Handwriting Recognition (ICFHR) 2012 database. More differential handwriting data makes the model perform relatively well, whereas the reality is that the original samples are insufficient. Otherwise, the proposed model uses only part of the ICDAR dataset, more in line with

actual handwriting circumstances. In other words, the metrics of the model are relatively robust compared with SOTA.

Table 8: Comparison results on ICDAR (in %)

Models	S-TOP1	S-TOP5	S-TOP10	H-TOP2	H-TOP3	*mAP*
Fiel et al. [15]	88.5	96.0	98.3	40.5	5.8	N/A
CS-UMD-b [20]	95.0	98.6	99.2	20.2	8.4	N/A
HIT-ICG [20]	94.8	98.0	98.3	63.2	36.5	N/A
Chen et al. [27]	96.6	N/A	N/A	N/A	N/A	**90.1**
Tang et al. [28]	99.0	99.2	99.6	84.4	**68.1**	N/A
Christlein et al. [29]	97.1	98.9	99.0	42.8	23.8	67.1
Lai et al. [30]	97.1	N/A	99.2	N/A	N/A	60.3
Proposed method	**99.0**	**99.3**	**99.7**	**84.8**	60.2	83.7

4.4.2 CVL Dataset

Due to the small capacity, it is easy to achieve better results on CVL. The proposed method achieves good results in each index, in addition, the model has strong robustness, see Table 9. The proposed model is only 0.4%, and 0.2% worse in soft TOP1 and TOP10 compared to other models. The handwriting features learned by CNN are relatively common, leading to the high probability of overfitting. Thus, the features extracted are similar, which makes the final results close to each other. Moreover, due to the problems of label error and handwriting ambiguity in the dataset itself, it is challenging for the model to extract distinguishing features. In addition, the dataset demands further processing. In comparison, we found that in *mAP*, the proposed method is 0.3% behind Lai et al. [30]. The main reason is that Lai adopted pathlet and SIFT features with machine learning, which augments the local feature extraction ability. However, the calculation is relatively complicated and system architecture is not an end-to-end framework.

Table 9: Comparison results on CVL (in %)

Models	S-TOP1	S-TOP5	S-TOP10	H-TOP2	H-TOP3	*mAP*
Fiel et al. [15]	98.9	99.3	99.5	97.6	93.3	N/A
CS-UMD [20]	97.9	99.1	99.4	90.9	71.2	N/A
TSINGHUA [20]	97.7	99.0	99.1	95.3	94.5	N/A
Chen et al. [27]	99.2	N/A	N/A	N/A	N/A	97.8
Lai et al. [30]	**99.7**	N/A	99.8	N/A	N/A	**98.7**
Proposed method	99.3	**99.5**	99.6	**98.3**	**97.2**	98.4

From the above experimental results, the proposed WI shows a good capability to extract and identify characteristics, mainly in two aspects: (1) The ResNet50 pre-trained by ImageNet represents a good generalization ability. And transfer learning improves the training effect by extracting similar writing features from related samples through the pre-trained neural network; (2) The double normalization for local features and global features effectively preserves the main features of handwriting

in feature extraction and dimension reduction, solving the problems of visual burst, correlation, and intersession variability.

5 Conclusion

In this paper, a novel method for WI with CNN and double-power normalization is proposed. We employ activation features extracted from ResNet50 that are pre-trained by ImageNet. After the local descriptors are normalized, VLAD is adopted to calculate the global features, following SSR. The experiments are conducted on the ICDAR2013 and CVL datasets. The evaluation shows that the method presents a good identification result and achieves the SOTA while having better robustness and fast real-time response capability.

In future work, we will test CNN with a more effective network structure to extract handwriting features and pay attention to the new normalization method. Furthermore, appropriate data augmentation and encoding methods are also considered due to insufficient handwriting features. The work after coding, that is, the classification method, also improves the final result to a certain extent, which needs to be studied.

Acknowledgement: The authors are grateful to all who supported us in producing this article and for those who contributed to this study but cannot include themselves.

Funding Statement: This work was supported in part by the Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant KYCX 20_0758, in part by the Science and Technology Research Project of Jiangsu Public Security Department under Grant 2020KX005, in part by the General Project of Philosophy and Social Science Research in Colleges and Universities in Jiangsu Province under Grant 2022SJYB0473, in part by "Cyberspace Security" Construction Project of Jiangsu Provincial Key Discipline during the "14th Five Year Plan".

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

References

[1] Y. Zhang and X. Ran, "A step-based deep learning approach for network intrusion detection," *Computer Modeling in Engineering & Sciences*, vol. 128, no. 3, pp. 1231–1245, 2021.

[2] X. Xu, Z. Fang, L. Qi, X. Zhang, Q. He *et al.,* "Traffic flow prediction driven resource reservation for multimedia IoV with edge computing," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 17, no. 2, pp. 1–21, 2021.

[3] Y. Xiong, Y. Lu and P. Wang, "Off-line text-indenpendent writer recognition: A survey," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 95, pp. 103912, 2020.

[4] C. I. Tomai, Z. Bin and N. S. Sargur, "Discriminatory power of handwritten words for writer recognition," in *Proc. Int. Conf. on Pattern Recognition*, Cambridge, UK, pp. 638–641, 2004.

[5] D. Liang and M. Wu, "A multi-patch deep learning system for text-independent writer identification," in *Proc. Int. Conf. on Security, Privacy, and Anonymity in Computation, Communication, and Storage*, Nanjing, China, pp. 409–419, 2020.

[6] S. He and L. Schomaker, "Delta-n hinge: Rotation-invariant features for writer identification," in *Proc. Int. Conf. on Pattern Recognition*, Stockholm, Sweden, pp. 2023–2028, 2014.

[7] R. Jain and D. Doermann, "Writer identification using an alphabet of contour gradient descriptors," in *Proc. IEEE Int. Conf. on Document Analysis and Recognition*, Washington, DC, USA, pp. 550–554, 2013.

[8] F. Slimane and V. Märgner, "A new text-independent GMM writer identification system applied to arabic handwriting," in *Proc. Int. Conf. on Frontiers in Handwriting Recognition*, Hersonissos, Greece, pp. 708–713, 2014.
[9] S. He and L. Schomaker, "GR-RNN: Global-context residual recurrent neural networks for writer identification," *Pattern Recognit*, vol. 117, pp. 107975, 2021.
[10] S. N. M. Khosroshahi, S. N. Razavi, A. B. Sangar and K. Majidzadeh, "Deep neural networks-based offline writer identification using heterogeneous handwriting data: An evaluation via a novel standard dataset," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, pp. 2685–2704, 2022.
[11] A. Semma, Y. Hannad, I. Siddiqi, S. Lazrak and M. Kettani, "Feature learning and encoding for multi-script writer identification," *International Journal on Document Analysis and Recognition*, vol. 25, pp. 79–93, 2022.
[12] H. Said, T. Tan and K. Baker, "Personal identification based on handwriting," *Pattern Recognit.*, vol. 33, pp. 149–160, 2000.
[13] 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.
[14] P. Singh, P. Roy and B. Raman, "Writer identification using texture features: A comparative study," *Computers & Electrical Engineering*, vol. 71, pp. 1–12, 2018.
[15] S. Fiel and R. Sablatnig, "Writer identification and writer retrieval using the fisher vector on visual vocabularies," in *Proc. IEEE Int. Conf. on Document Analysis and Recognition*, Washington, DC, USA, pp. 545–549, 2013.
[16] S. Fiel and R. Sablatnig, "Writer identification and retrieval using a convolutional neural network," in *Proc. Int. Conf. on Computer Analysis of Images and Patterns*, Valletta, Malta, pp. 26–37, 2015.
[17] V. Christlein, D. Bernecker, A. Maier and E. Angelopoulou, "Offline writer identification using convolutional neural network activation features," in *Proc. German Conf. on Pattern Recognition*, Aachen, Germany, pp. 540–552, 2015.
[18] V. Christlein, M. Gropp, S. Fiel and A. Maier, "Unsupervised feature learning for writer identification and writer retrieval," in *Proc. IEEE Int. Conf. on Document Analysis and Recognition*, Kyoto, Japan, pp. 991–997, 2017.
[19] M. Javidi and M. Jampour, "A deep learning framework for text-independent writer identification," *Engineering Applications of Artificial Intelligence*, vol. 95, pp. 103912, 2020.
[20] G. Louloudis, B. Gatos, N. Stamatopoulos and A. Papandreou, "ICDAR 2013 competition on writer identification," in *Proc. IEEE Int. Conf. on Document Analysis and Recognition*, Washington, DC, USA, pp. 1397–1401, 2013.
[21] F. Kleber, S. Fiel, M. Diem and R. Sablatnig, "Cvl-database: An off-line database for writer retrieval, writer identification and word spotting," in *Proc. IEEE Int. Conf. on Document Analysis and Recognition*, Washington, DC, USA, pp. 560–564, 2013.
[22] A. Dengel and R. Ahmad, "A novel skew detection and correction approach for scanned documents," in *Proc. Int. Association for Pattern Recognition Workshop on Document Analysis Systems*, Santorini, Greece, 2016.
[23] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 770–778, 2016.
[24] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Proc. Computer Vision and Pattern Recognition*, Minneapolis, Minnesota, USA, pp. 1–8, 2007.
[25] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 88–98, 1998.
[26] J. Delhumeau, P. H. Gosselin, H. J´egou and P. P´erez, "Revisiting the VLAD image representation," in *Proc. ACM Multimedia Conf.*, Barcelona, Spain, pp. 653–656, 2013.
[27] S. Chen, Y. Wang, C. Lin, W. Ding and Z. Cao, "Semi-supervised feature learning for improving writer identification," *Information Sciences*, vol. 482, no. 4, pp. 156–170, 2019.

[28] Y. Tang and X. Wu, "Text-independent writer identification via CNN features and joint Bayesian," in *Proc. Int. Conf. on Frontiers in Handwriting Recognition*, Shenzhen, China, pp. 566–571, 2016.
[29] V. Christlein, D. Bernecker, F. Hönig and E. Angelopoulou, "Writer identification and verification using GMM supervectors," in *Proc. IEEE Winter Conf. on Applications of Computer Vision*, Steamboat Springs, CO, USA, pp. 998–1005, 2014.
[30] 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.