Multi-view Representation Learning with Deep Features for Offline Signature Verification

Xingbiao Zhao, Changzheng Liu, Benzhuang Zhang, Limengzi Yuan, and Yuchen Zheng(✉)

College of Information Science and Technology, Shihezi University, Shihezi, China
{zxb,benzhuangzhang}@stu.shzu.edu.cn, {liucz,ylmz}@shzu.edu.cn, ouczyc@outlook.com

Abstract. Feature learning is one of the most crucial steps in offline signature verification systems. In this paper, to improve the performance of deep learning-based features for the offline signature verification task, we propose a novel framework to learn the new representations from two views of deep features by Canonical Correlation Analysis-based (CCA-based) multi-view representation learning approaches. Specifically, the features from one view can be extracted from deep learning-based feature extractors and the other view can be generated from the extracted view by adding the noise to another homologous sample. Then, the different CCA-based multi-view representation learning methods are evaluated on these two-view deep features to generate the joint features as the final features for the next verification step. Extensive experiments and discussions on three benchmark offline handwritten signature datasets demonstrate that the proposed framework improves the deep learning-based features and achieves the state-of-the-art results compared with other verification systems.

Keywords: Offline signature verification · Feature learning · Multi-view representation learning · Deep learning · Canonical correlation analysis

1 Introduction

The handwritten signature verification system is a typical human-computer interface system which automatically verifies whether a query signature is genuine or forged. Generally, according to the collection process, the signature verification systems are divided into two types: online and offline. The online systems collect the signatures as a series of dynamic sequences which include the speed, the pressure coordinate sequence, the pen-ups trajectory, etc. Different from the online verification systems, the offline verification systems collect the signatures as static digit images. Since the dynamic information can not be accessed during the training process and the skilled forgeries are very similar to the genuine

H. Gao and X. Wang (Eds.): CollaborateCom 2021, LNICST 407, pp. 261–275, 2021.
https://doi.org/10.1007/978-3-030-92638-0_16

signatures (the imitators can access the genuine signatures and imitate many times), the offline signature verification task is relatively challenging compared with the online signature verification task. Therefore, discriminating the genuine signatures and skilled forgeries is a crucial task in verification systems.

The verification task is often composed of preprocessing, feature extraction, and building Writer-Dependent (WD) or Writer-Independent (WI) classifiers [7]. Among them, feature extraction is one of the most crucial processes and determines the success or failure of the whole task. The handcrafted and deep learning-based features are typically applied to various verification systems. Comparing with traditionally handcrafted features, such as Local Binary Pattern (LBP) [8], Histogram of Oriented Gradient (HOG) [38], and Scale Invariant Feature Transform (SIFT) [28], deep learning-based features [9,12,35] are more popular in recent years.

The deep learning-based feature extractors often train the deep neural networks (DNNs) with different strategies to capture the discriminative information between different signatures. However, to achieve good performances, a large-scale training set is needed, which makes it hard to transfer to different datasets and implement in real-world applications. In addition, the problem of signature diversity that only several signatures could be accessed from the target signer also limits the learning ability of deep learning-based feature extractors.

To improve the learning ability of deep learning-based features for the offline signature verification task, in this paper, we propose a novel CCA-based multi-view representation learning framework which generates a new joint feature from different views of deep features. Detailedly, for the feature extraction process, we extract one view of deep features from a fully connected layer in a convolutional neural network (CNN). Then, the features from another view could be generated by adding random noise to another sample from the same class. Based on these two-view features, we evaluate several CCA-based multi-view representation learning approaches and generate the final learned features from one of them. After the feature learning process, we train the Support Vector Machine (SVM) as the WD classifiers for each user to build the completed verification system. The main contributions of this paper are listed as follows.

- We propose a novel framework to apply and explore the learning ability of multi-view representation learning approaches for the offline signature verification task.
- The different CCA-based multi-view representation learning approaches are evaluated and discussed on deep learning-based features, which improves the learning ability of deep features and demonstrates a novel view for the feature extraction process.
- Extensive experimental results and evaluations on three benchmark offline handwritten signature datasets demonstrate that the proposed framework achieves the state-of-the-art performance in most cases.

The rest of the paper is organized as follows. We review related work of traditional and deep learning-based feature extractors in offline signature verification systems and multi-view representation learning approaches in Sect. 2. In Sect. 3, the proposed framework is described in detail. Experimental results performed on three benchmark datasets are presented in Sect. 4. Section 5 concludes this paper with remarks and future work.

2 Related Work

2.1 Feature Extraction in Offline Signature Verification Systems

To capture the discriminative information between different signatures, many traditionally handcrafted [16,17,40] and deep learning-based [9,12,35] feature extractors are proposed in recent years. For traditionally handcrafted feature extractors, they aim to design different descriptors to capture the local or global information of signatures. In [16], it extracted parametric LBP features from the signatures, and then applied fusion using CCA to improve the discriminative features. In [17], two global (the number of connected components and the number of active pixels) and eight local (coordinates of effective mass, number of active pixels, isolated points, etc.) features were extracted from each signature, and then they were concatenated to create a final feature vector for feature learning. More recently, Zhou et al. extracted Gray-Level Co-Occurrence Matrix (GLCM) and HOG features from the static signature images and then fused them with dynamic information from online data to build the feature extraction procedure [40]. Although the handcrafted feature extractors achieve good performance in many scenarios, they are hard to capture the micro differences between the genuine signatures and their corresponding skilled forgeries.

Since the deep learning-based approaches have obtained excellent achievement in various areas, such as edge computing [4,21], Internet of Things (IoT) [25,36], document recognition and analysis [5,23], etc. [2,20], the deep learning-based feature extractors are more and more popular in modern signature verification systems [12,26,35]. In [12], a CNN-based architecture was applied to capture the difference not only between different signers but also between genuine signatures and skilled forgeries. In [26], Masoudnia et al. proposed Multi-Loss Snapshot Ensemble (MLSE) that combines a dynamic multi-loss function and a novel ensemble framework for simultaneously learning the features between different signatures. Similarly, Wan and Zou [35] used a dual triplet loss to train the model for feature learning, where two different triplets are constructed for random and skilled forgeries, respectively. On the whole, the deep learning-based feature extractors are designed to capture the discriminative information between genuine signatures and forgeries. However, only several signatures could access from the target signer, which limits the learning ability and makes it hard to train.

2.2 Multi-view Representation Learning

Multi-view representation learning is an emerging research area in machine learning which aims to learn a projection to model each view and jointly optimizes all the functions to improve the generalization performance [39]. In recent year, many multi-view learning approaches are proposed to handle different applications [1,22,29]. In [29], Nie et al. proposed a novel multi-view learning model which learns local structure and performs clustering or semi-supervised classification tasks simultaneously. In [22], the Gaussian Process Latent Variable Model (GPVLM) was applied to represent multiple views in a common subspace for visual classification. In [1], Deep Canonical Correlation Analysis (DCCA) was proposed to learn complex nonlinear transformations of two views of data. Unlike Kernel Canonical Correlation Analysis (KCCA) [27], DCCA does not require to compute inner product of original representations, which is more efficient than KCCA.

To improve the learning ability of deep learning-based feature extractors in offline signature verification systems, applying multi-view representation learning to automatically learn a new representation from multiple views of signatures is very natural. To the best of our knowledge, only several studies like [16,19] used CCA to learn the new representation from the views of traditionally handcrafted and dynamic features. It is better to use the multiple views of deep learning-based features to further improve the performance of offline signature verification systems.

3 Multi-view Representation Learning for Offline Signature Verification Systems

In this section, we first introduce how to generate the second view from the original view of deep features. Then, we introduce several CCA-based multi-view representation learning approaches. Finally, we introduce how to train SVMs as the WD classifiers for building the completed verification system.

3.1 Generating the Second View from Deep Features

After feature extraction, we obtain the deep features from a fully connected layer in a deep CNN and take them as the first view for multi-view representation learning. Based on the deep features, we first rescale the feature values to $[0, 1]$ by Min-Max normalization. The rescaled element can be described as,

$$X_{ij}^* = \frac{X_{ij} - \min\{X_{ij}\}}{\max\{X_{ij}\} - \min\{X_{ij}\}}. \tag{1}$$

For each feature from the first view, we randomly select a feature from the same class with the feature of the first view in the same specific signer and rescale the feature to $[0, 1]$. Then, we add the independently random noise uniformly sampled from $[0, 1]$ to each value of a feature. Finally, we also rescale the values to $[0, 1]$ by Min-Max normalization and take the final features as the second view features. For a robust multi-view representation learning method, it should learn an effective representation from two views of features by ignoring the noise. The next section will introduce several CCA-based multi-view representation learning approaches for further learning the features for the verification system.

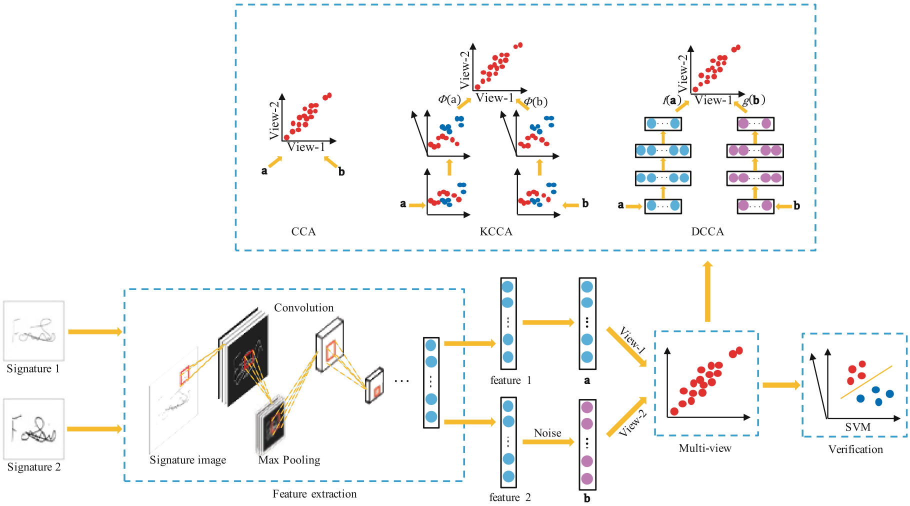

Fig. 1. The proposed framework for offline signature verification.

3.2 CCA-based Multi-view Representation Learning Approaches

In this part, we introduce three CCA-based representation learning approaches, CCA, KCCA, and DCCA which are used in our experiments. **Canonical Correlation Analysis (CCA)** [15] analyzes the linear relationship between the characteristics of two views. Let $\mathbf{A} = \{\mathbf{a}_1, ..., \mathbf{a}_n\}$ and $\mathbf{B} = \{\mathbf{b}_1, ..., \mathbf{b}_n\}$ denote the data from two views, $\mathbf{U} = \{\mathbf{u}_1, ..., \mathbf{u}_d\}$ and $\mathbf{V} = \{\mathbf{v}_1, ..., \mathbf{v}_d\}$ denote the CCA directions that project the data from two views, the objective function can be described as,

$$
\begin{aligned}
\max_{\mathbf{u}_i, \mathbf{v}_i} \quad & \mathbf{u}_i^\top \mathbf{A}\mathbf{B}^\top \mathbf{v}_i \\
\text{s.t.} \quad & \mathbf{u}_i^\top (\frac{1}{N}\mathbf{A}\mathbf{A}^\top + r_a \mathbf{I})\mathbf{u}_i = 1, \\
& \mathbf{v}_i^\top (\frac{1}{N}\mathbf{B}\mathbf{B}^\top + r_b \mathbf{I})\mathbf{v}_i = 1, \\
& \mathbf{u}_i^\top \mathbf{A}\mathbf{B}^\top \mathbf{v}_j = 0, \quad i \neq j,
\end{aligned}
\tag{2}
$$

where, r_a, $r_b > 0$ are regularization parameters which are used to handle the overfitting problem. This objective function can be solved by several approaches, such as Singular Value Decomposition (SVD). CCA tries to find linear projections of two random vectors that are maximally correlated. In CCA, the final projection mapping of a test sample $\mathbf{x}_{test}$ is $\mathbf{U}^\top \mathbf{x}_{test}$. The learned dimensions in each view should be uncorrelated and provide different information from their own perspectives.

Kernel CCA (KCCA) [14] is an extension of CCA, which tries to find nonlinear projection mappings from two views of data. The objective function of KCCA is "max $\mathbf{u}_i^\top \Phi(\mathbf{A})\Phi(\mathbf{B})^\top \mathbf{v}_i$", where, $\Phi(\mathbf{A})$ and $\Phi(\mathbf{B})$ are the projections of two views of data in a kernel space. We replace $\Phi(\mathbf{a}) = \boldsymbol{\alpha}_i^\top \kappa_a(\mathbf{a}, \cdot)$, where $\kappa_a(\mathbf{a}, \cdot)$ is a vector whose i-th element is $\kappa_a(\mathbf{a}, \mathbf{a}_i)$. Then, the objective function can be replaced as,

$$\begin{aligned} \max_{\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i} \quad & \boldsymbol{\alpha}_i^\top \mathbf{K}_a \mathbf{K}_b \boldsymbol{\beta}_i \\ \text{s.t.} \quad & \boldsymbol{\alpha}_i^\top \mathbf{K}_a^2 \boldsymbol{\alpha}_i + r_a \boldsymbol{\alpha}_i^\top \mathbf{K}_a \boldsymbol{\alpha}_i = 1, \\ & \boldsymbol{\beta}_i^\top \mathbf{K}_b^2 \boldsymbol{\beta}_i + r_b \boldsymbol{\beta}_i^\top \mathbf{K}_b \boldsymbol{\beta}_i = 1, \\ & \boldsymbol{\alpha}_i^\top \mathbf{K}_a \mathbf{K}_b \boldsymbol{\beta}_j = 0, \quad i \neq j, \end{aligned} \tag{3}$$

where, $\mathbf{K}_a \in \mathbb{R}^{n \times n}$, $\mathbf{K}_a = \mathbf{K} - \mathbf{K1} - \mathbf{1K} + \mathbf{1K1}$, and $K_{ij} = \kappa_a(\mathbf{a}_i, \mathbf{a}_j)$. For regularization, the $\mathbf{K}_a^2$ is replaced by $\mathbf{K}_a^2 + r_a \mathbf{K}_a$ here. The size of kernel matrices is $n \times n$. If n is very large, this problem is hard to solve and time consuming, which is not friendly for large-scale applications.

Deep Canonical Correlation Analysis (DCCA) [1]**.** Different from KCCA, DCCA learns the nonlinear representation of two views of data by building two DNNs. The features extracted from two DNNs from two views are $f(\mathbf{A})$ and $g(\mathbf{B})$. Then, the canonical correlation between the two views of deep features is maximized,

$$\begin{aligned} \max_{\mathbf{u}_i, \mathbf{v}_i, \boldsymbol{\Theta}_f, \boldsymbol{\Theta}_g} \quad & \mathbf{u}_i^\top f(\mathbf{A}) g(\mathbf{B})^\top \mathbf{v}_i \\ \text{s.t.} \quad & \mathbf{u}_i^\top (\frac{1}{N} f(\mathbf{A}) f(\mathbf{A})^\top + r_a \mathbf{I}) \mathbf{u}_i = 1, \\ & \mathbf{v}_i^\top (\frac{1}{N} g(\mathbf{B}) g(\mathbf{B})^\top + r_b \mathbf{I}) \mathbf{v}_i = 1, \\ & \mathbf{u}_i^\top f(\mathbf{A}) g(\mathbf{B})^\top \mathbf{v}_j = 0, \quad i \neq j, \end{aligned} \tag{4}$$

where, $\boldsymbol{\Theta}_f$ and $\boldsymbol{\Theta}_g$ are the weight parameters of two DNNs. The parameters of CCA and DNNs can be optimized by Limited-memory Broyden Fletcher Goldfarb Shanno (L-BFGS) [24] method with large mini-batches. Then, the final features for a test sample $\mathbf{x}_{test}$ can be represented as $\mathbf{U}^\top f(\mathbf{x}_{test})$. In addition, DCCA does not need to compute the inner product between two samples to obtain the nonlinear representation. However, designing suitable architectures for two DNNs is very important and decides the learning performance of DCCA.

3.3 Training the Writer-Dependent Classifiers

After feature extraction and multi-view feature learning processes, the final features are learned from two views of deep features. Then, we choose SVMs as the WD classifiers for building the completed verification system. For each signer, we take the genuine signatures from a target signer as the positive samples, and the genuine signatures from other signers as the negative samples since the skilled forgeries can not be accessed during the training process. In this scenario, the negative samples are much more than the positive samples. To solve the imbalance problem in a fair way, the different weights C to balance the relationship between the positive and negative classes,

$$C_N = \frac{N}{P} C_p, \tag{5}$$

where, N and P represent the numbers of negative and positive samples. Then, the objective function of SVM is described as,

$$\begin{gathered}\min_{\mathbf{w},\boldsymbol{\xi}} \frac{1}{2}\mathbf{w}^\top \mathbf{w} + C_P \left(\sum_{i:y_i=+1} \xi_i \right) + C_N \left(\sum_{i:y_i=-1} \xi_i \right), \\ \text{s.t.} \quad y_i \left(\mathbf{w}^\top \mathbf{x}_i + b \right) \geq 1 - \xi_i, \quad \xi_i \geq 0, \end{gathered} \tag{6}$$

where, $\mathbf{x}_i$ is a training sample with target label y_i, and ξ_i is the slack variables.

The whole framework is illustrated in Fig. 1. First, we apply a CNN-based feature extractor to extract the features for different signatures. Then, based on one view of features from a signature, we generate the second view of features from another homologous sample. Next, the multi-view representation learning approaches are used to further generate the final features for the verification process. Finally, the SVMs are trained for each user as the WD classifiers for building the completed verification system.

4 Experiment

In this section, we first introduce the datasets used in the experiments. Then, we introduce the experimental settings in detail. Finally, We demonstrate the experimental results and discussion.

4.1 Dataset

We conduct the experiments by extracting the deep features from two CNN-based architectures, named 'SigNet' and 'SigNet-F' [12], on GPDS-960 [34], CEDAR [18], and Brazilian PUC-PR [6] datasets. The SigNet aims to capture the differences only between different users. The SigNet-F aims to capture the differences not only between different users but also between genuine signatures and skilled forgeries. The SigNet and SigNet-F are trained on the development set (signatures of the final 531 users) of the GPDS-960 dataset.

For the GPDS-960 dataset, it contains 881 users. Each user has 24 genuine signatures and 30 skilled forged signatures, with a total of 47,574 signatures. GPDS-160 and GPDS-300 datasets are the subsets of the GPDS-960 dataset. They contain the signatures of the first 160 and 300 users of the GPDS-960 dataset. For the CEDAR dataset, it contains 55 users and each user has 24 genuine signatures and 24 forged signatures, with a total of 2,640 signatures. The Brazilian PUC-PR dataset contains 168 users. For the first 60 users, each user has 40 genuine signatures, 10 simple forged signatures, and 10 skilled forged signatures. For the last 108 users, each user only has 40 genuine signatures and with a total of 7,920 signatures.

4.2 Experimental Settings

For the feature extraction, we extract features from the output of the second fully connected layer from SigNet and SignNet-F. The dimension size of extracted features is 2,048. After the feature extraction, we generate the second view of extracted features and apply the CCA-based multi-view representation learning approaches for feature learning. For KCCA, we use Radial Basis Function (RBF) as the kernel function, and set $\gamma = 0.001$. Since the KCCA is hard to train when the number of training samples is too large, we use 50% samples for each user to build the training set in our experiments to reduce the memory requirement. For DCCA, we use two DNN networks with the same structure of $2048 - 1024 - 2048 - 4096 - 2048$. The epoch is set to 10, batch size = 2000 (because the training process needs a relatively big batch size for observing the behavior on different views of signatures), and learning rate = 0.001.

After feature extraction and feature learning processes, we train SVMs as the WD classifiers for establishing a completed signature verification system. We follow the data partition scheme in [12] to build the training set. For the GPDS-160 dataset, we randomly select $n \in \{1, ..., 14\}$ genuine signatures from the target user as the positive samples, and 10 genuine signatures of each user from the last 721 users in the GPDS-960 dataset as the negative samples. For the GPDS-300 dataset, we also randomly select $n \in \{1, ..., 14\}$ genuine signatures from the target user as the positive samples, and 10 genuine signatures of each user from the last 581 users in the GPDS-960 dataset as the negative samples. For the CEDAR dataset, we randomly select $n \in \{1, ..., 12\}$ genuine signatures from the target user as the positive samples, and 10 genuine signatures of each user from the remaining 54 users. For the Brazilian PUC-PR dataset, we randomly select $n \in \{1, ..., 30\}$) genuine signatures from the target user in the first 60 users as the positive samples, and 10 genuine signatures of each user from the last 108 users as the negative samples. For parameters in linear SVM, the C_N is set to 1 and C_P is calculated by Eq. 5.

For the test process, we randomly select 10 genuine signatures and 10 skilled forgeries from the target user, 10 genuine signatures from the development set as random forgeries for each user on GPDS-160 and GPDS-300 datasets. For the CEDAR dataset, we randomly select 10 genuine signatures and 10 skilled forgeries for each user as the test set. For the Brazilian PUC-PR dataset, we randomly select 10 genuine signatures, 10 skilled forgeries from the target user, and 10 genuine signatures from the last 108 users in the Brazilian PUC-PR dataset as random forgeries. Finally, the experimental results are the averages with 10 trials.

Table 1. Comparing with different evaluation measurements on CCA, KCCA, and DCCA-based multi-view representation learning approaches on GPDS-160 and GPDS-300 datasets. Here, the feature that we use is SigNet-F, the number of reference samples is 12 (the number of training samples in SVM). The red and green values represent the improvement and deterioration compared with the original features, respectively.

Method	Dataset	FRR	FAR_random	FAR_skilled	EER_global	EER_user
-	160	6.39 (±0.67)	0.01 (±0.02)	3.96 (±0.18)	5.15 (±0.28)	2.60 (±0.39)
CCA	160	5.34 (±0.33)	0.01 (±0.00)	3.22 (±0.24)	4.21 (±0.13)	1.93 (±0.22)
KCCA	160	3.24 (±0.21)	0.02 (±0.00)	5.42 (±0.21)	4.25 (±0.20)	1.98 (±0.13)
DCCA	160	19.48 (±1.10)	0.48 (±0.02)	5.06 (±0.25)	9.78 (±0.20)	7.60 (±0.27)
-	300	6.80 (±0.31)	0.00 (±0.01)	6.16 (±0.17)	6.44 (±0.17)	3.56 (±0.18)
CCA	300	5.70 (±0.36)	0.01 (±0.00)	3.53 (±0.14)	4.49 (±0.15)	2.35 (±0.17)
KCCA	300	3.60 (±0.36)	0.01 (±0.00)	5.82 (±0.17)	4.63 (±0.18)	2.43 (±0.21)
DCCA	300	20.35 (±0.89)	0.53 (±0.02)	5.05 (±0.25)	9.63 (±0.28)	7.39 (±0.29)

Table 2. Comparison with the state-of-the-art verification systems on the GPDS-160 dataset. Here, "#Refs" represents the number of reference samples during the training process.

Feature	#Refs	EER (%)
Curvelet transform [10]	12	15.07
LBP, HOG, SIFT [37]	5	7.98
LBP, HOG, SIFT [37]	12	6.97
AlexNet [11]	14	2.74 (±0.18)
SigNet-F [12]	5	3.52 (±0.28)
SigNet-F [12]	12	2.60 (±0.39)
CCA-SigNet-F	12	**1.93 (±0.22)**
KCCA-SigNet-F	12	1.98 (±0.13)
DCCA-SigNet-F	12	7.60 (±0.27)

4.3 Evaluation Measurements

We use the following evaluation measurements to measure the performance of different approaches. False Rejection Rate (FRR): the rate of the genuine signatures that are rejected as forgeries; False Acceptance Rate for random forgeries ($\mathrm{FAR}_{\mathrm{random}}$): the rate of the random forgeries that are accepted as genuine signatures; False Acceptance Rate for skilled forgeries ($\mathrm{FAR}_{\mathrm{skilled}}$): the rate of the skilled forgeries that are accepted as genuine signatures; Equal Error Rate (EER): the error when FAR = FRR. Here, the EER includes two forms, i.e. $\mathrm{EER}_{\mathrm{user}}$: using user-specific decision thresholds and $\mathrm{EER}_{\mathrm{global}}$: using a global decision threshold.

4.4 Experiments on the GPDS Dataset

Table 1 shows the performance of CCA, KCCA, and DCCA-based multi-view representation learning methods with several evaluation measurements on the GPDS-160 and GPDS-300 datasets. We can see that the KCCA-based approach achieves the lowest FRR both in the GPDS-160 and GPDS-300 datasets, and the CCA-based method also improves FRR. For the $\mathrm{FAR}_{\mathrm{skilled}}$, only the CCA-based method improves it on the GPDS-160 dataset and all multi-view representation learning methods improve it and the CCA-based method achieves the best $\mathrm{FAR}_{\mathrm{skilled}}$ on the GPDS-300 dataset. For the $\mathrm{EER}_{\mathrm{global}}$ and $\mathrm{EER}_{\mathrm{user}}$, the CCA and KCCA-based methods improve them on the GPDS-160 and GPDS-300 datasets. In addition, the DCCA-based method does not perform well. The reason might be that the deep features extracted from the SigNet-F have already included the major information of the signatures. The DNNs in DCCA-based methods may destroy this information and are hard to train from the features learned from the other deep architectures.

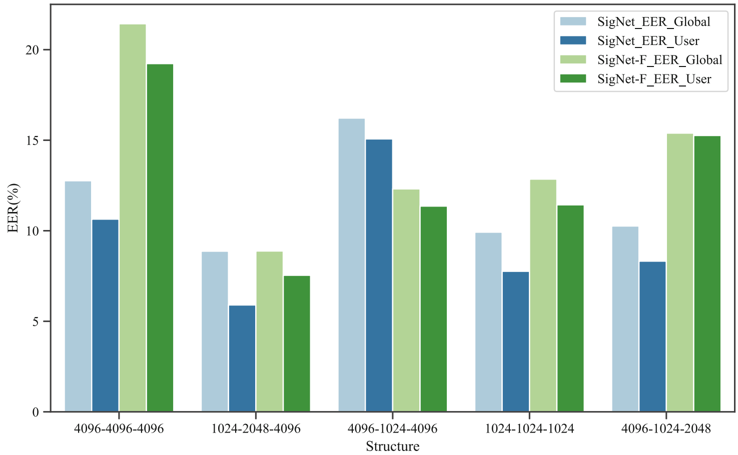

Fig. 2. The performance of the DCCA with different structures on the validation set (the users from No. 352 to No. 402) of the GPDS-960 dataset. Here, the input and output are all 2048. Therefore, we only show the middle three layers in the figure.

Tables 2 and 3 demonstrate the results that compares with the state-of-the-art verification systems on the GPDS-160 and GPDS-300 datasets. According to the tables, we can see that the CCA and KCCA-based methods improve the performance of the SigNet-F feature. The CCA-based method achieves 1.93% and 2.35% EERs which are the best results on the GPDS-160 and GPDS-300 datasets compared with the other state-of-the-art verification systems. In addition, the DCCA-based method also degrades the performance compared with CCA and KCCA-based methods, the reason might be similar to before.

Table 3. Comparison with the state-of-the-art verification systems on the GPDS-300 dataset.

Feature	#Refs	EER (%)
LBP [33]	10	20.94
Grid Feat [41]	12	3.24
GLBP [31]	12	2.84
Quad-tree [32]	10	9.30
SigNet-F [12]	5	4.84 (±0.26)
SigNet-F [12]	12	3.56 (±0.18)
CCA-SigNet-F	12	**2.35 (±0.17)**
KCCA-SigNet-F	12	2.43 (±0.21)
DCCA-SigNet-F	12	7.39 (±0.29)

Table 4. Comparison with the state-of-the-art verification systems on the CEDAR dataset.

Feature	#Refs	EER (%)
Quad-tree [32]	10	18.15
Grid Feat [41]	10	3.02
GLBP [31]	12	7.94
SigNet [12]	8	5.03 (±0.75)
SigNet [12]	12	4.76 (±0.36)
SigNet-F [12]	8	4.77 (±0.76)
SigNet-F [12]	12	4.63 (±0.42)
CCA-SigNet-F	8	2.79 (±0.53)
CCA-SigNet-F	12	**2.55 (±0.41)**
KCCA-SigNet-F	8	4.41 (±0.80)
KCCA-SigNet-F	12	3.93 (±0.41)
DCCA-SigNet-F	8	11.49 (±0.84)
DCCA-SigNet-F	12	10.90 (±0.70)

Since the DCCA-based method does not work well in most cases, we evaluate different structures of DCCA on the validation set of the GPDS-960 dataset. Figure 2 shows the performance of different structures of DCCA with the EER measurements on the features of SigNet and SigNet-F. From the figure, we can see that the structure $1024-2048-4096$ achieves the best performance compared with the other structures. That is the reason why we choose this structure in our experiments. In addition, if the structure is $4096-4096-4096$, the performance becomes much worse, which means the completed architectures will degrade the learning ability of the DCCA-based method.

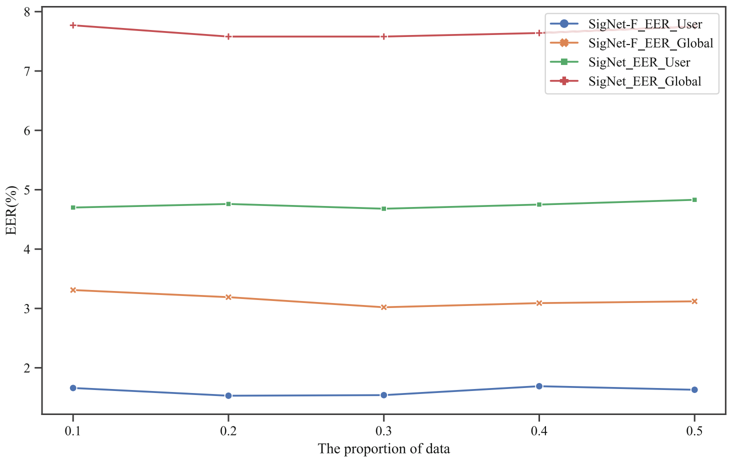

Fig. 3. The performance of the KCCA with different proportions of data on the validation set.

To evaluate the influence of the proportion of data on the KCCA-based method, we design an experiment using the different proportions of signatures from each user to train the KCCA. Figure 3 demonstrates the results with the EER measurements on the features of SigNet and SigNet-F on the validation set of the GPDS-960 dataset. We can see that the performance is not very sensitive to the change of proportions. It means that we can use a small amount of data to train the KCCA in our experiments.

4.5 Experiments on the CEDAR and Brazilian PUC-PR Datasets

To evaluate the generalization ability of the proposed framework, we test it on the CEDAR and Brazilian PUC-PR datasets. Here, the features are also extracted from the SigNet or SigNet-F which are trained on the GPDS dataset. Especially, due to the simple forgeries that are existed in the Brazilian PUC-PR dataset, some systems report the Average Error Rate (AER) that is calculated as $AER = (FRR + FAR_{random} + FAR_{simple} + FAR_{skilled})/4$, where, the FAR_{simple} is the rate of the simple forgeries that are accepted as genuine signatures. The experimental results are shown in Tables 4 and 5.

It can be seen from the tables that the proposed framework also obtains the best performance compared with the other state-of-the-art verification systems on the CEDAR and Brazilian PUC-PR datasets. The CCA and KCCA-based methods improve the performance of features that are extracted from the SigNet or SigNet-F. The CCA-based method achieves 2.65% and 2.55% EERs on the CEDAR and Brazilian PUC-PR datasets when the number of reference samples is 12, respectively, which is better than the KCCA and DCCA-based methods.

Table 5. Comparison with the state-of-the-art verification systems on the Brazilian PUC-PR datasets.

Feature	#Refs	AER/EER (%)
Deep CNN features [30]	30	4.17
Graphometric [3]	15	5.65
MAML [13]	15	5.74 (±0.84)
SigNet [12]	15	2.07 (±0.63)
SigNet [12]	30	2.01 (±0.43)
SigNet-F [12]	15	4.03 (±0.59)
SigNet-F [12]	30	3.44 (±0.37)
CCA-SigNet	15	1.06 (±0.37)
CCA-SigNet-F	15	1.42 (±0.50)
CCA-SigNet	30	**0.92 (±0.24)**
CCA-SigNet-F	30	1.08 (±0.34)
KCCA-SigNet	15	2.38 (±0.38)
KCCA-SigNet-F	15	2.29 (±0.51)
KCCA-SigNet	30	2.11 (±0.37)
KCCA-SigNet-F	30	1.98 (±0.47)
DCCA-SigNet	15	3.83 (±0.48)
DCCA-SigNet-F	15	5.40 (±0.62)
DCCA-SigNet	30	3.57 (±0.69)
DCCA-SigNet-F	30	5.20 (±0.76)

5 Conclusion

In this paper, we propose that applying multi-view representation learning approaches to build a framework to further improve the performance of the features learned from the deep learning-based architectures for offline signature verification tasks. Extensive experimental results on three benchmark datasets prove that the CCA and KCCA-based multi-view representation learning methods improve the learning ability of deep learning-based feature extractors and achieve state-of-the-art or competitive performance in most cases. In future work, since the DCCA-based method can not work well, we plan to explore different strategies on this method and evaluate more multi-view representation learning-based methods for combining dynamic information for verification systems.

Acknowledgement. This work was supported by Innovation and Cultivation Project for Youth Talents of Shihezi University (Grant Number CXPY201905), Financial and Science Technology Plan Project of Xinjiang Production and Construction Corps (Grant Number 2017DB005), Key Industry Innovation and Development Support Plan in Southern Xinjiang (Grant Number 2017DB005), Fund Project of XJPCC (2019AB001, 2017CD010, and 2017CA018), and Shihezi University Scientific Research Project for High-level Talents (RCZK202028).

References

1. Andrew, G., Arora, R., Bilmes, J., Livescu, K.: Deep canonical correlation analysis. In: ICML, pp. 1247–1255 (2013)
2. Bengio, Y., Lecun, Y., Hinton, G.: Deep learning for AI. Commun. ACM **64**(7), 58–65 (2021)
3. Bertolini, D., Oliveira, L.S., Justino, E., Sabourin, R.: Reducing forgeries in writer-independent off-line signature verification through ensemble of classifiers. Pattern Recogn. **43**(1), 387–396 (2010)
4. Chen, J., Ran, X.: Deep learning with edge computing: a review. Proc. IEEE **107**(8), 1655–1674 (2019)
5. Chen, X., Jin, L., Zhu, Y., Luo, C., Wang, T.: Text recognition in the wild: a survey. ACM Comput. Surv. **54**(2), 1–35 (2021)
6. de Freitas, C.O.A., et al.: Bases de dados de cheques bancários brasileiros
7. Diaz, M., Ferrer, M.A., Impedovo, D., Malik, M.I., Pirlo, G., Plamondon, R.: A perspective analysis of handwritten signature technology. ACM Comput. Surv. **51**(6), 1–39 (2019)
8. Ferrer, M.A., Vargas, J.F., Morales, A., Ordonez, A.: Robustness of offline signature verification based on gray level features. IEEE Trans. Inf. Forensics Secur. **7**(3), 966–977 (2012)
9. Ghosh, R.: A recurrent neural network based deep learning model for offline signature verification and recognition system. Expert Syst. Appl. **168**, 114249 (2021)
10. Guerbai, Y., Chibani, Y., Hadjadji, B.: The effective use of the one-class SVM classifier for handwritten signature verification based on writer-independent parameters. Pattern Recogn. **48**(1), 103–113 (2015)
11. Hafemann, L.G., Sabourin, R., Oliveira, L.S.: Analyzing features learned for offline signature verification using deep CNNs. In: ICPR, pp. 2989–2994 (2016)
12. Hafemann, L.G., Sabourin, R., Oliveira, L.S.: Learning features for offline handwritten signature verification using deep convolutional neural networks. Pattern Recogn. **70**, 163–176 (2017)
13. Hafemann, L.G., Sabourin, R., Oliveira, L.S.: Meta-learning for fast classifier adaptation to new users of signature verification systems. IEEE Trans. Inf. Forensics Secur. **15**, 1735–1745 (2019)
14. Hardoon, D.R., Szedmak, S., Shawe-Taylor, J.: Canonical correlation analysis: an overview with application to learning methods. Neural Comput. **16**(12), 2639–2664 (2004)
15. Hotelling, H.: Relations between two sets of variates*. Biometrika **28**(3–4), 321–377 (1936)
16. Houtinezhad, M., Ghaffary, H.R.: Writer-independent signature verification based on feature extraction fusion. Multimedia Tools Appl. **79**, 6759–6779 (2019). https://doi.org/10.1007/s11042-019-08447-7
17. Jain, A., Singh, S.K., Singh, K.P.: Signature verification using geometrical features and artificial neural network classifier. Neural Comput. Appl. **33**(12), 6999–7010 (2020). https://doi.org/10.1007/s00521-020-05473-7
18. Kalera, M.K., Srihari, S., Xu, A.: Offline signature verification and identification using distance statistics. Int. J. Pattern Recognit Artif Intell. **18**(07), 1339–1360 (2004)
19. Khan, S.H., Khan, Z., Shafait, F.: Can signature biometrics address both identification and verification problems? In: ICDAR, pp. 981–985 (2004)

20. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
21. Li, H., Ota, K., Dong, M.: Learning IoT in edge: deep learning for the internet of things with edge computing. IEEE Netw. **32**(1), 96–101 (2018)
22. Li, J., Li, Z., Lu, G., Xu, Y., Zhang, B., Zhang, D.: Asymmetric Gaussian process multi-view learning for visual classification. Inf. Fusion **65**, 108–118 (2021)
23. Lin, Q., Luo, C., Jin, L., Lai, S.: STAN: a sequential transformation attention-based network for scene text recognition. Pattern Recogn. **111**, 107692 (2021)
24. Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Math. Program. **45**(1), 503–528 (1989)
25. Lv, Z., Qiao, L., Li, J., Song, H.: Deep-learning-enabled security issues in the internet of things. IEEE Internet Things J. **8**(12), 9531–9538 (2020)
26. Masoudnia, S., Mersa, O., Araabi, B.N., Vahabie, A.H., Sadeghi, M.A., Ahmadabadi, M.N.: Multi-representational learning for offline signature verification using multi-loss snapshot ensemble of CNNs. Expert Syst. Appl. **133**, 317–330 (2019)
27. Melzer, T., Reiter, M., Bischof, H.: Kernel canonical correlation analysis. In: ICANN, pp. 353–360 (2001)
28. Nasser, A.T., Dogru, N.: Signature recognition by using sift and surf with SVM basic on RBF for voting online. In: ICET, pp. 1–5 (2017)
29. Nie, F., Cai, G., Li, J., Li, X.: Auto-weighted multi-view learning for image clustering and semi-supervised classification. IEEE Trans. Image Process. **27**(3), 1501–1511 (2017)
30. Rantzsch, H., Yang, H., Meinel, C.: Signature embedding: writer independent offline signature verification with deep metric learning. In: Bebis, G., et al. (eds.) ISVC 2016. LNCS, vol. 10073, pp. 616–625. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-50832-0_60
31. Serdouk, Y., Nemmour, H., Chibani, Y.: New gradient features for off-line handwritten signature verification. In: INISTA, pp. 1–4 (2015)
32. Serdouk, Y., Nemmour, H., Chibani, Y.: Handwritten signature verification using the quad-tree histogram of templates and a support vector-based artificial immune classification. Image Vis. Comput. **66**, 26–35 (2017)
33. Soleimani, A., Araabi, B.N., Fouladi, K.: Deep multitask metric learning for offline signature verification. Pattern Recogn. Lett. **80**, 84–90 (2016)
34. Vargas, F., Ferrer, M., Travieso, C., Alonso, J.: Off-line handwritten signature GPDS-960 corpus. In: ICDAR, vol. 2, pp. 764–768 (2007)
35. Wan, Q., Zou, Q.: Learning metric features for writer-independent signature verification using dual triplet loss. In: ICPR, pp. 3853–3859 (2021)
36. Yao, S., Zhao, Y., Zhang, A., Hu, S., Shao, H., Zhang, C., Su, L., Abdelzaher, T.: Deep learning for the internet of things. Computer **51**(5), 32–41 (2018)
37. Yılmaz, M.B., Yanıko**u**glu, B.: Score level fusion of classifiers in off-line signature verification. Inf. Fusion **32**, 109–119 (2016)
38. Yilmaz, M.B., Yanikoglu, B., Tirkaz, C., Kholmatov, A.: Offline signature verification using classifier combination of hog and LBP features. In: IJCB, pp. 1–7 (2011)
39. Zhao, J., Xie, X., Xu, X., Sun, S.: Multi-view learning overview: recent progress and new challenges. Inf. Fusion **38**, 43–54 (2017)
40. Zhou, Y., Zheng, J., Hu, H., Wang, Y.: Handwritten signature verification method based on improved combined features. Appl. Sci. **11**(13), 5867 (2021)
41. Zois, E.N., Alewijnse, L., Economou, G.: Offline signature verification and quality characterization using poset-oriented grid features. Pattern Recogn. **54**, 162–177 (2016)