Int. J. Appl. Math. Comput. Sci., *2025, Vol. 35, No. 2, 357–370*
DOI: 10.61822/amcs-2025-0025

VANILLA CONVOLUTIONAL NEURAL NETWORK IS ALL YOU NEED FOR ONLINE AND OFFLINE SIGNATURE VERIFICATION

MUSTAFA BERKAY YILMAZ [a,*], KAĞAN ÖZTÜRK [b]

[a] Department of Computer Engineering
Akdeniz University
Dumlupınar Boulevard, 07058 Konyaaltı, Antalya, Turkey
e-mail: berkayyilmaz@akdeniz.edu.tr

[b] Department of Computer Science and Engineering
University of Notre Dame
Holy Cross Drive 384, 46556 Notre Dame, Indiana, USA
e-mail: kaganozturk1992@gmail.com

Recent advances in deep learning have been utilized successively to improve the performance of signature verification (SV) systems. Deep models proposed in the literature are complicated and need to learn many parameters to give acceptable error rates, requiring a lot of training data. On the other hand, those models are designed and hand-crafted specializing in the problem, online or offline SV. In this work, we suggest and show on popular datasets that similar and simple convolutional neural network (CNN) models can achieve state-of-the-art results both for offline and online SV problems. For offline SV, our work outperforms its counterparts with and without data augmentation. We also show that a very similar CNN architecture can be employed for online SV. To the best of our knowledge, this is the first work to show that CNNs can be used to learn online signature representations directly from raw data.

Keywords: signature verification, representation learning, deep learning, convolutional neural networks.

1. Introduction

Signature verification systems (SVSs) aim to distinguish reference signatures from forgeries. Different numbers of genuine signatures (e.g., 1, 5, 12) of a writer can be used as reference signatures. Forgery signatures are broadly divided into two categories: random and skilled. Random forgeries are signed without knowledge of the signature to be imitated. Signatures of different writers are commonly considered random forgeries and used to evaluate the performance of SVSs. Skilled forgeries (SFs) on the other hand, are signed after some practice to imitate a genuine signature. In this case, the forger has access to one or several signatures of a writer. Since SFs have great resemblance to genuine signatures, it is important to evaluate the performance of an SVS on SFs.

SVSs are divided into two categories depending on the acquisition method: offline and online. Offline signatures are represented as a digital image with binary or gray-level pixel values. They are captured after the writing process is completed. In online case, signatures are acquired during the writing process. In addition to position trajectories, pressure, azimuth and elevation signals can be stored depending on the acquisition device. Because of the dynamic information, it is easier to detect forgeries in online SV systems.

For decades, SV has been of great interest to researchers. A comprehensive review can be found in Diaz *et al.* (2019). Direct comparison of studies is not usually possible because of various experimental protocols, such as use of databases for training and testing and evaluation metrics. A common practice to report results is the equal error rate (EER) which is the threshold point where the false accept rate (FAR) and the false reject rate (FRR) are equal. The area under the curve (AUC) is another performance metric that measures the area under the receiver operating characteristic curve.

The objective of this work is to demonstrate the

*Corresponding author

complexities of current SV approaches and the differences of models in online and offline cases. The motivation is to develop a simpler system that can handle online and offline SV without specifically designing and fine-tuning for each case. For this purpose, we present a novel feature learning strategy to detect forgeries. The discriminative power of features is evaluated for both offline and online SVSs. For offline signatures, the effectiveness of higher resolution at test time is investigated following Touvron *et al*. (2019). Significant performance gain is obtained even without data augmentation. A CNN is employed to learn online signature representation directly from raw data, to the best of our knowledge for the first time. It has been shown that a simple CNN can outperform more complex architectures (Ahrabian and BabaAli, 2019).

The rest of the paper is organized as follows. Section 2 summarizes the related works on SV. Section 3 describes the proposed methodology for representation learning. The experimental protocol and results are given in Sections 4 and 5. Finally, Section 6 concludes the paper.

2. Related work

Strenuous research effort has been devoted to extracting handcrafted features before the application of deep learning approaches. Common feature extraction methods for online signatures are categorized into two types: function-based and parametric. Function-based methods are used to represent signatures as a time function of features such as position, velocity, acceleration, pressure, direction of pen movement. On the other hand, in the parametric approach, a signature is represented as a vector of elements. Each element is a feature such as the total signature time, the number of pen ups and pen downs, the wavelet transform, the Fourier transform and so on. Common classification methods include dynamic time warping (DTW), principal component analysis (PCA), Euclidean distance, hidden Markov models (HMM) and support vector machines (SVM) (Impedovo and Pirlo, 2008). Recent literature surveys on online and offline SV can be found in (Kaur and Kumar, 2021; Hameed *et al*., 2021; Minaee *et al*., 2023). In addition to handwritten signatures, it is possible to sign a message or document digitally. A group signature scheme even allows a group member to anonymously sign a document in substitution for the group, protecting the privacy with the help of blockchain (Devidas *et al*., 2021).

2.1. Offline SV. In (Yılmaz and Yanıkoğlu, 2016) the histogram of gradients, local binary features and a scale invariant feature transform are used to deploy an offline SVS. Scores of writer-dependent (WD) and writer-independent (WI) classifiers are combined to make a decision. Sparse coding is employed to represent offline signatures in (Zois *et al*., 2017). A local feature-pooling method that uses second-order statistics of the sparse codes is proposed with a segmentation strategy that utilizes spatial pyramid and binary robust invariant scalable keypoints. Zois *et al*. (2019) propose a feature extraction method that measures the asymmetric relations between pixel templates. A decision stump committee with a boosting feature selection algorithm is utilized to build the classifier.

Recent studies in offline SVS have shifted to a feature-learning approach instead of relying on handcrafted features. While learning features to classify genuine signatures of writers is a simple task that can achieve low error rates via CNNs, learning features for detection of SFs is an important research area. Hafemann *et al*. (2017) propose a CNN to learn features for offline SV. A weighted sum of two loss functions are minimized. While a multi-class cross entropy loss term is used to classify writers of genuine signatures, a binary cross entropy term forces network to distinguish genuine signatures and SFs. They also train another network using only genuine signatures to measure the impact of the usage of SFs on feature learning. Another CNN architecture is proposed by Calik *et al*. (2019). They propose a classifier algorithm to recognize offline signatures in the case of a limited training size per writer.

A siamese network is proposed to build an offline SVS by Dey *et al*. (2017). The network is trained to minimize the Euclidean distance between similar signature pairs while maximizing the distance between genuine-forgery pairs. In (Yılmaz and Öztürk, 2018), a two-channel CNN (Zagoruyko and Komodakis, 2015) architecture is used to make a binary decision. While one channel is allocated only for genuine signatures as references, the second channel is fed with genuine signatures or SFs as queries. Score combinations of WI and WD verifiers are utilized for lower error rates.

To encode a sequential representation into static signature images, static-dynamic interaction networks (SDINet) have been utilized for offline SV (Li *et al*., 2021). A static signature image is converted to sequences by assuming pseudo-dynamic processes in the static image, followed by the extraction of deep features from signature images describing the global information of signatures. The static-to-dynamic conversion and the dynamic-to-static attention are unified into a compact framework. Accuracies of 94.42%, 95.00% and 89.66% are reported on BH-SigB, BH-SigH and GPDS-Synth4000 (different from GPDS-Synth10000), respectively.

A two-channel and two-stream transformer approach (2C2S) to cope with the SV problem is proposed by Ren *et al*. (2023), consisting of original and central streams. The original stream receives the original signature pair as input; on the other hand, the central stream receives the signature pair generated by cropping the central at the original pair as input. Verification accuracy is measured on SUES-SiG, CEDAR, BHSig-B, and BHSig-H, reaching 93.25%, 90.68%, 100%, and 72.22%, respectively.

Applying the data augmentation directly to the features instead of the signature image has been proposed (Arab *et al*., 2023). The features generator is based on mutation, cloning, and resources competition mechanisms of artificial immune systems. Experiments performed on CEDAR, GPDS-300 and MCYT-75 datasets with 5 references provided EERs of 5.00%, 7.80% and 8.30%, respectively.

A writer-independent offline SV approach using attention-based multiple siamese networks with primary representation guiding has been proposed (Xiong *et al*., 2023). The proposed system takes the reference signature images, query signature images, and their corresponding inverse images as inputs. These images are fed to four weight-shared parallel branches, respectively. A mutual attention module discovers prominent stroke information from original and inverse branches. According to the experiments on BHSig-H, BHSig-B and UTSig datasets EERs of 10.17%, 8.11% and 18.08% have been reported, respectively.

To solve the problem of a small amount of effective information in signature images, an end-to-end multi-path attention inverse discrimination network that focuses on the signature stroke parts to extract features by reversing the foreground and background of signature images has been proposed (Zhang *et al*., 2023). The problem of high intraclass variability has been handled by multi-path attention modules between discriminative streams and inverse streams. The method has been tested on CEDAR, BHSig-Bengali, BHSig-Hindi, and GPDS Synthetic datasets with accuracies of 100%, 96.24%, 93.86%, and 83.72%, respectively.

The problem of a limited number of signatures in offline SV has been addressed by Hameed *et al*. (2023). A deep learning-based image augmentation model is capable of augmenting better-quality signatures with diversity from a single signature image only.

Zois *et al*. (2023) propose the mapping of handwritten signature images to points of the tangent space of a connected symmetric positive definitive (SPD) manifold for SV. Based on the principles of differential geometry, the limited training data problem is targeted in this manifold by proposing feature augmentation methods. A meta-learning framework in the space of the SPD manifold to learn a pairwise similarity metric for writer independent offline SV has been offered (Giazitzis and Zois, 2024). Pairs of handwritten signatures are first converted into a multidimensional distance vector with elements corresponding to SPD distances between spatial segments of the corresponding covariance pairs. A meta-learning approach then follows to explore the structure of the input gradients of the SPD manifold utilizing a recurrent model, constrained by the geometry of the SPD manifold.

Learning with rejection and top-rank learning techniques are applied by Ji *et al*. (2023). To provide a single input, a pair of genuine and query signatures is stacked in a single feature vector named the paired contrastive feature (PCF), internally representing the similarity between two signatures.

A multi-task framework for learning handwritten signature feature representations based on deep contrastive learning has been proposed by Viana *et al*. (2023). As the first task, signature examples of the same writer are mapped closer within the feature space while separating the feature representations of signatures of different writers. In the second task, SF representations are adjusted by adopting contrastive losses with the goal of performing hard negative mining. On GPDSsynthetic first 300 writers, 4.02%, 3.24% and 3.33% EERs are obtained with 5, 10 and 12 references using WD models. 5.48%, 4.69% and 4.51% EERs are obtained with 5, 10 and 12 references using WI models. On CEDAR 4.43%, 3.45% and 3.50% EERs are obtained with 5, 10 and 12 references using WD models, respectively. While 5.91%, 4.91% and 4.59% EERs are obtained with 5, 10 and 12 references using WI models, respectively. On MCYT-75 4.07% and 2.71% EERs are obtained with 5 and 10 references using WD models. 4.97% and 4.07% EERs are obtained with 5 and 10 references using WI models.

2.2. Online SV. DTW is used to align online signatures of variable lengths by Kholmatov and Yanikoglu (2005). They use an SVM, a Bayes classifier and linear classifier with PCA to obtain a decision boundary between genuine and forgery signatures. An HMM-based online SV approach is proposed by Fierrez *et al*. (2007). Function-based methods with a rotation alignment procedure are utilized to represent signatures. They investigated the impact of the number of states and Gaussian mixtures per state.

Template signatures are substituted with an artificial signature called the hidden signature, generated by minimizing the misalignment between itself and the training signatures of the writer (Putz-Leszczynska, 2015).

Several artificial neural network architectures have been investigated in recent studies for online SVSs. Lai *et al*. (2017) proposed a novel descriptor, called a length-normalized path signature. A recurrent neural network (RNN) is trained with the triplet loss function to learn a distance metric.

Tolosana *et al*. (2018), proposed a siamese (Bromley *et al*., 1994) architecture with RNNs. First, 23 time functions are extracted using x and y coordinates and pressure. Then, bidirectional long-short-term memory

(BiLSTM) and bidirectional gated recurrent unit (BGRU) networks are used to make a binary decision given a signature pair. Another siamese architecture is used in Ahrabian and BabaAli (2019). A recurrent autoencoder network is employed to learn WI features. Also, an attention mechanism is utilized to improve performance. Use of CNNs (Lecun *et al.*, 1998) with a siamese architecture is investigated by Vorugunti *et al.* (2019). Global features are given to the CNN, showing that even a shallow network is sufficient to achieve low error rates.

An in-depth analysis of state-of-the-art deep learning approaches for online SV is provided by Tolosana *et al.* (2021), together with DeepSignDB online handwritten signature biometric public database and a standard experimental protocol and benchmark. A time-aligned RNN TARNN is utilized for the task of online SV. The TA-RNN achieves an EER below 2% with SF impostors and one reference signature per user on DeepSignDB.

Time-series are transformed into a 2D representation for online SV by Xie *et al.* (2022). The channel-wise weight-learning method is integrated to discover the relationship between altitude, azimuth, and pressure. Xie *et al.* (2023) combine stroke images and sensor signals for verification using the supervised fusion triplet network. An existing private dynamic signature dataset is converted into static and dynamic form for the simulations.

OSVConTramer, a combined CNN and transformer is utilized for online SV (Vorugunti *et al.*, 2023). OSVConTramer learns optimal local and global dependencies of input signature feature vectors. EERs of 10.85%, 5.45% and 6.32% are reported on MCYT-100, SVC, and SUSIG datasets, respectively.

A teacher-student collaborative knowledge distillation (TSKD) technique is proposed for online SV (Sekhar *et al.*, 2023). After training a heavy transformer-based teacher, teacher knowledge is distilled into a very lightweight CNN-based student. The teacher network results in deep representative feature learning by the student with performance improvement. One-shot learning results in 12.42%, 6.45% and 11.32% EERs on MCYT-100, SVC and SUSIG datasets, respectively.

A CNN and a convolutional gated recurrent network (CGRN) to extract spatial and temporal features have been combined for online SV (Yu and Shi, 2023). A cosine similarity for spatial features calculates the shape similarity and dynamic time warping (DTW) aligns temporal features. The distance between reference and query signatures is calculated by multiplying the DTW the distance and similarity score.

2.3. Mixed online-offline verification. Online handwriting is utilized for registration instead of handwritten images (Qiao *et al.*, 2007). The online registration is supposed to enable robust recovery of the writing trajectory from offline signature and allows effective shape matching between registration and verification signatures. The proposed technique is reported to achieve comparable performance with online SV methods; on the other hand, it requires both online and offline samples from each subject.

Offline training to offline testing, offline training to online testing, online training to offline testing and online training to online testing cases are investigated (Uppalapati, 2007). Completely different handcrafted features are extracted for offline and online samples. A feature-level combination is followed to verify the signatures.

Zimmer and Ling (2008) employ online reference data acquired through a digitizing tablet as the basis for the segmentation process of the corresponding scanned offline data. Gathering online signatures for registration is mandatory for this system to work.

Features of both online and offline handwritten signatures are verified separately, combining their results to verify the signature (Radhika and Gopika, 2015). Online data consist of the signing process captured using a webcam and offline data consist of the scanned signatures. Both the modalities are required for the system to work. The data set is collected from only 13 different subjects. The combined approach gives better results than the compared single-modality systems as expected.

According to recent works, although SV is a hot research topic, no other work focuses on simple and similar models that can handle online and offline modalities from raw data. Recent works that report promising results in the literature are becoming very complicated with millions of learnable parameters, thus demanding huge amounts of training data. At the same time those models are specially designed for the target problem, either offline or online SV. We propose to fill this gap by designing simpler and similar models both for offline and online SV, still achieving state-of-the-art results. CNNs are trained from raw data without relying on handcrafted features both for offline and online data.

3. Proposed method

CNNs have recently shown great performance in recognition tasks. Features learned for recognition can also be used for verification. The main problem in SV is not only classifying genuine signatures of writers but also detecting SFs that have great resemblance to genuine signatures.

Feature learning strategies using CNNs for SV can be categorized into two categories. In the first case, two signature pairs are given to the CNN in order to measure similarity. The network is trained to learn genuine signature pairs of each writer as similar pairs while random forgeries and SFs can be used to create dissimilar pairs (Tolosana *et al.*, 2018; Vorugunti *et al.*, 2019; Dey

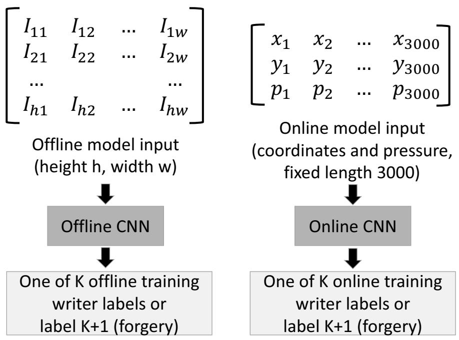


Fig. 1. Inputs and outputs for the proposed $K + 1$ CNN models for representation learning. Each writer in the training set is labeled separately and one additional class is used to represent SFs from all writers.

et al., 2017; Yılmaz and Öztürk, 2018). While it can achieve state-of-the-art results when trained and tested on the same database, a significant performance drop is observed for the cross-database experimental procedure (Dey *et al.*, 2017). It is also important to note that the requirement of pairs is inefficient for large databases compared with CNN architectures taking a single image as input. In the other case, genuine signatures of writers are categorized into different classes as a standard N-way CNN training procedure. In addition, SFs can be treated as separate classes (Yılmaz and Öztürk, 2019) or an additional loss term can be utilized to detect them (Hafemann *et al.*, 2017).

In this work, we propose a CNN architecture to learn a signature representation similar to that of Yılmaz and Öztürk (2019). Two kinds of models are investigated. In the first model, SFs are considered as separate classes as in Yılmaz and Öztürk (2019) resulting in a network with $K \times 2$ outputs. As an alternative second model, all SFs are treated as one additional class resulting in $K + 1$ outputs using signatures of K writers (Fig. 1). We hypothesize that forcing the network to consider all SF created for different writers as one group can help network to capture generic characteristics of forgeries. As a result of the $K + 1$ approach, the model is prevented from overfitting to writer-specific forgery characteristics for producing writer-specific forgery outputs.

Moreover, the proposed work applies a simple strategy for SFs and obviates the need for searching the values for additional parameter in the loss function proposed by Hafemann *et al.* (2017). It is important to emphasize that SFs that are used in any kind of model training (WI and WD) are only gathered from the training set. This is practical because an existing signature database with SFs can be easily obtained before the deployment of the system to real users.

Once the network is trained, it is used as a feature descriptor for the writers in the test and validation sets in a WI way. Then, WD classifiers are employed to detect forgeries. Experiments are conducted for both offline and online SV tasks. For offline signature feature extraction phase, usage of higher resolution at test time than that of training time, rather is also investigated.

Table 1. List of parameters of RandomResizedCrop.

Parameter	Value	Parameter	Value
Size	150×220	Scale	(0.2, 1)
Ratio	(3/4, 4/3)	Interpolation	Bilinear

3.1. Feature learning.

3.1.1. Preprocessing. Offline signatures are processed using a simple procedure (Yılmaz and Öztürk, 2018). First, gray-level values are inverted by extracting them from 255, so that background is presented by 0 values. Then, small components are removed with the assumption that they are sourcing from noise.

We use a fixed-length pen trajectory (3000) for each of the observations of (x, y) coordinates and pen pressure information, representing online signatures with the fixed size of 3000×3. This value is determined from the observations in the validation set, where the maximum length is 3552 in an SF and 1156 in a genuine sample. Only 5 SFs exceed the length of 3000. Each dimension is normalized so that the maximum and minimum values become 1 and -1. Smaller lengths are padded with 0 values and signatures having more than 3000 time steps are cropped.

3.1.2. Data augmentation. RandomResizedCrop implementation in Pytorch (Paszke *et al.*, 2019) is used to augment data for offline signatures. Parameters chosen for augmentation are listed in Table 1, simply to create a relevant artificial variability. The proposed architecture also is trained without using data augmentation for comparison. In this case, the original image is resized to 150×220 after the preprocessing step. We did not apply any augmentation procedure for online signatures.

3.1.3. CNN architecture. The proposed CNN architectures are built as shown in Algorithm 1. A motivation behind the algorithm is to keep the models simple and very similar for offline and online SV. The argument list for constructing offline and online networks is given in Table 2. Minor differences between the online and offline models in hyperparameters and structures result from the difference in input sizes (because of the representation sizes) and output sizes (because of the sizes of the training sets K for online and offline models by which the CNN output size is determined).

Algorithm 1. CNN architecture.

procedure $\mathrm{NET}(b, kf, k, s, d, p, o)$
  $net \leftarrow []$
  $n \leftarrow 32$
  $net.add(conv\_bn\_relu(n, kf, p))$
  for $i = 1$ to b **do**
    if d or i in $[1, 2, 3, 5, 7]$ **then**
      $n \leftarrow n \times 2$
    end if
    $net.add(maxpooling(k, s)$
    $net.add(conv\_bn\_relu(n, k))$
    $net.add(conv\_bn\_relu(n, k))$
  end for
  $net.add(gap())$ (feature extraction)
  $net.add(fc\_softmax(o))$
  return net

The sizes of feature maps are reduced by the max-pooling operation. Stride and padding parameters of convolutional layers are set to 1 and $(k - 1)/2$, respectively, to preserve to input shape except the first layer of the online network. The dimension of online signatures is reduced to 3000×1 after the first convolutional layer; then one-dimensional operations are applied. The number of convolutional filters is set to 32 and increased by a factor of 2 five times. $K + 1$ outputs are produced to separate genuine signatures and SFs of K writers. The same architecture with $K \times 2$ outputs is also used for comparison.

Networks are trained for 100 epochs using the Adam optimizer (Kingma and Ba, 2014). Batch normalization (Ioffe and Szegedy, 2015) is utilized before each activation function (ReLU). Label smoothing is applied to prevent the networks becoming over-confident (Müller *et al.*, 2019). Cross-entropy loss is minimized using a smoothing factor of $\alpha = 0.1$,

$$H(y, p) = \sum_{n=1}^{N} -y_n^{\mathrm{LS}} \log(p_n), \tag{1}$$

where N is the number of classes, y_n is 1 for the correct class and 0 for the rest,

$$y_n^{LS} = y_n(1 - \alpha) + \alpha/N. \tag{2}$$

A scheme depicting the $K + 1$ architecture both for online and offline cases is shown in Fig. 2, together with the number of trainable parameters for each layer. A sequential-block primitive structure is also shown in the same figure. The total numbers of trainable parameters for online and offline models are $10,319,633$ and $19,414,740$ accordingly. The total space complexities for online and offline models are 55.95 MB and 119.50 MB accordingly. The total number of floating-point operations (FLOPs) for online and offline models are 570 million and 2.44 billion, respectively. For comparison, VGG has around 19 billion FLOPs and ResNet-34 has around 3.6 billion FLOPs.

Table 2. List of hyperparameters of Algorithm 1.

Parameter	Offline	Online	Description
b	5	7	number of blocks
kf	$(7,7)$	$(7,3)$	first kernel size
k	$(3,3)$	$(3,1)$	kernel size
p	$(3,3)$	$(3,0)$	zero-padding
s	$(2,2)$	$(2,1)$	stride
d	True	False	double filters at each block
o	N	N	number of classes

3.2. Writer-dependent classification. While the network can directly be used for writer recognition and forgery detection on the training set without requiring any additional classifier, WI or WD classifiers are needed to verify signatures for a different set of writers. In the literature, WD classifiers trained for specific writers have always provided better results than WI classifiers as expected (Yılmaz and Yanıkoğlu, 2016; Yılmaz and Öztürk, 2018; Viana *et al.*, 2023). For this reason after the CNN learns signature representations, two-class WD SVM classifiers are trained for each writer to accept (genuine class) or reject (forgery class) query signatures for the corresponding writer. For offline signatures, a set of input resolutions from which features are extracted is used to measure the effect of higher resolution. Hyperparameters of WD SVM are determined using the validation set, as described in Section 4.1. Here c-SVM with radial basis function (RBF) kernel is used to build WD classifiers. After a brief and coarse grid-search, the cost is determined as 32 and `gamma` of the RBF is determined as 0.125 both for online and offline cases.

4. Experimental protocol

For the implementation of CNNs, the PyTorch library is used with Python. For the implementation of WD SVMs, LIBSVM with Matlab is used.

4.1. Offline SV.

4.1.1. Standard setup. Experiments with offline signatures are conducted on GPDS-960 (Vargas *et al.*, 2007), CEDAR (Kalera *et al.*, 2004) and MCYT-75 (Ortega-Garcia *et al.*, 2003) databases containing 881, 55 and 75 writers respectively. GPDS-960 is partitioned into 3 parts, following a very similar protocol to those in the literature (Hafemann *et al.*, 2017; Yılmaz and Öztürk, 2018).

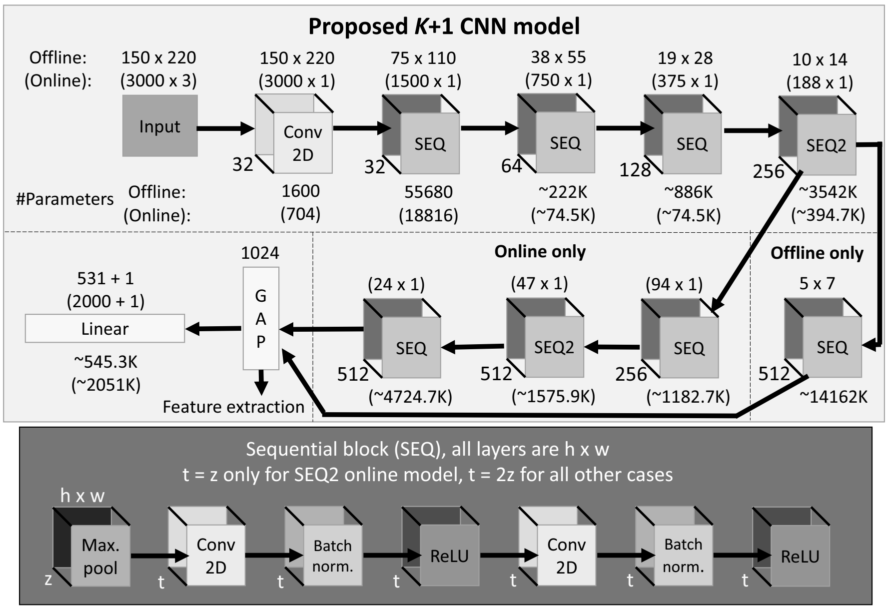


Fig. 2. $K + 1$ architecture for online and offline cases indicating the number of trainable parameters for each layer.

The training set T consisting of 531 writers in GPDS-960 (writer indices [351–881]), is used to train CNNs. SFs from T may or may not be used, where the two approaches are investigated separately. Two networks, one with data augmentation (Section 3.1.2) and the other without augmentation are trained to measure the effect of the augmentation policy.

The validation set V consisting of 50 GPDS-960 writers (indices [301–350]) is utilized to choose the best model during training. At every 5 epochs, a linear SVM classifier is trained to separate 51 classes of V for the validation purpose. Unlike WD verification of RBF SVMs, the purpose of the linear SVM is to perform the writer recognition for CNN validation faster, where the ultimate goal of the CNN training is to learn feature representations for SV.

While 5 genuine signatures per writer are used to learn 50 classes, SFs in T are used to learn the 51st class. The remaining genuine signatures and all SFs in V are used for evaluating the linear SVM. The model with the highest accuracy is picked for feature extraction. Hyperparameters of WD classifiers are coarsely determined by training and testing on the subsets of V.

First 300 writers in GPDS-960 (writer indices [1–300]), all writers in CEDAR and MCYT-75 databases are used as test set to evaluate the performance. Two-class WD SVMs are trained for each writer in the test set using features extracted from the global average pooling (GAP) layer of the CNN (trained on T). For cross-validation of WD SVMs performance, each writer's genuine samples in the test set are divided into two parts P_1 and P_2, each having 12 distinct samples per writer. Partitioning is randomly repeated 2 times. Then, r genuine samples are randomly selected from P_1 as reference signatures to train WD SVMs. The selection of references is randomly repeated 3 times. Negative samples for training the WD SVMs are selected from SFs of writers in V, which can be considered as random forgeries. Note that it is fair to use other writers' SFs outside of the test set as one can collect SFs from some random writer set before seeing any test subject.

The union of P_2 and writer-specific SFs is used as query samples for each test writer. Here $r = 5$ and $r = 12$ are considered to report the results. In total, 6 (two partitioning, each with three times of reference selection) tests are performed for $r = 5$. For $r = 12$, P_1 is completely covered so that there is no random reference set selection inside the partition, resulting in 2 tests (only for partitioning). Random forgeries (forgeries from other writers) are not used during any test.

While the CNN is trained using the fixed 150×220 input size, either directly resized from the original signature or from the cropped images in case of data augmentation, we use signatures with a higher resolution to extract features. Since signature representations are obtained from the output of the GAP layer, feature dimensions remain fixed for higher input resolutions. 150×220, 200×300, 250×375 and 300×450 are used for experiments. For 300×450, we observed a direct increase in error rates and did not report results. We did not perform a detailed investigation to find optimal resolutions and did not apply any fine-tuning procedure as opposed to Touvron *et al.* (2019).

Unless explicitly stated, all of the experiments follow this standard setup explained in this section. An overview of the proposed method demonstrating the protocol is shown in Fig. 3.

4.1.2. Mixed model setup. For further investigation, we set up an additional experimentation where we train the system using samples both from GPDS-960 and GPDSsyntheticOnLineOffLineSignature offline (GPDSsynthOff).

GPDSsyntheticOnLineOffLineSignature database (Ferrer *et al.*, 2016) contains 24 genuine and 30 SF signatures for each of the 10000 synthetic writers both online and offline. T_s is defined as writers [1001–1531] of GPDSsynthOff. V_s is defined as GPDSsynthOff writers [301–1000]. Only a $K + 1$ output CNN is trained in this setup. The exact setup for the mixed model case is then given as follows:

- the training set T_m: $T_s \cup T$,
- the validation set V_m: random samples from $V_s \cup V$,
- the test set: GPDSsynthOff first 300 writers and GPDS-960 first 300 writers (separately tested on both sets).

4.1.3. GPDSsynthOff model setup. For the investigation of the effect of the training set, the model $(K + 1)$ is inspected with the following setup:

- the training set: T_s,
- the validation set: V_s,
- test set: the same as in Section 4.1.2.

4.2. Online SV. GPDSsyntheticOnLineOffLineSign database online samples are utilized to conduct experiments with online signatures, which is abbreviated GPDSsynthOn. It contains 24 genuine and 30 SF signatures for each of the 10000 synthetic writers. Three disjoint sets are used for training, testing and validation. Three CNNs are trained containing 500, 1000 and 2000 writers in the training set (writer indices [4001–6000]). 300 (writer indices [1–300]) and 100 (writer indices [2001–2100]) writers are used for testing and validation, respectively. The same protocol (Section 4.1) is applied to pick the best CNN model, determining the hyperparameters of WD SVM verifiers and cross-validation of WD SVMs.

5. Results and discussion

EER and AUC are used to report the results, calculated from WD SVM test scores. EER_{user} and $\text{EER}_{\text{global}}$ depict the EERs with user-specific and global thresholds, accordingly. Global thresholds and user-based thresholds are calculated from the test scores. Mean AUC results are reported using global thresholds and depicted as Mean $\text{AUC}_{\text{global}}$, averaged for AUC values over all test writers. Table 3 shows the effect of different resolutions at the feature extraction phase for offline signatures. It can be seen that for all cases, the proposed architecture $K + 1$ (all SFs as one class) outperforms $K \times 2$ (SFs as separate classes) (Yılmaz and Öztürk, 2019). Although the difference between 200×300 and 250×375 is not obvious, the use of the higher resolution than training leads to better results in all cases.

In Tables 4 and 5 we compare our work with another signature representation learning method proposed by Hafemann *et al.* (2017). Results with 250×375 are shown for the proposed work. The effect of using SFs from the training set of the CNN is also reported. When SFs are not used, the proposed network and that of Hafemann *et al.* (2017) are trained with the same objective, i.e., to classify genuine signatures of 531 writers. It is important to note that, even though their work achieves lower EERs when only genuine signatures are considered during the training of the CNNs, the proposed work yields better results utilizing SFs, even without data augmentation. Table 4 shows that the performance gain of the proposed method not only comes from usage of a higher resolution or a hyperparameter choice, but also from the learning strategy.

Close results between models trained with and without data augmentation can indicate that our resolution choices for feature extraction are not optimal, and better results can be achieved via a more detailed search of resolutions as by Touvron *et al.* (2019). As shown in Hafemann *et al.* (2018), usage of multiple sizes during training can be considered to lower error rates. They achieve an EER of 0.41% compared with their base network (1.69%).

The performance of the proposed method is also investigated for online signatures in Table 6. 500, 1000 and 2000 synthetic writers are used to learn signature representations. The $K + 1$ model achieves lower error rates than the $K \times 2$ model as in offline results (Table 3). Compared with the state-of-the-art results (Ahrabian and BabaAli, 2019), the proposed solution achieves lower error rates when 2000 writers are used for feature learning, even though a much simpler architecture is utilized. The effect of the number of training subjects is also investigated by Ahrabian and BabaAli (2019) to report higher accuracy with 150 training subjects than 2000 training subjects. Although we observe close results between networks trained with 500 and 1000 writers in some cases, the lowest error rates are obtained with the training set containing 2000 writers, indicating the scalability of our learning strategy.

We use t-SNE (Maaten and Hinton, 2008) to visualize offline signature representations on V.

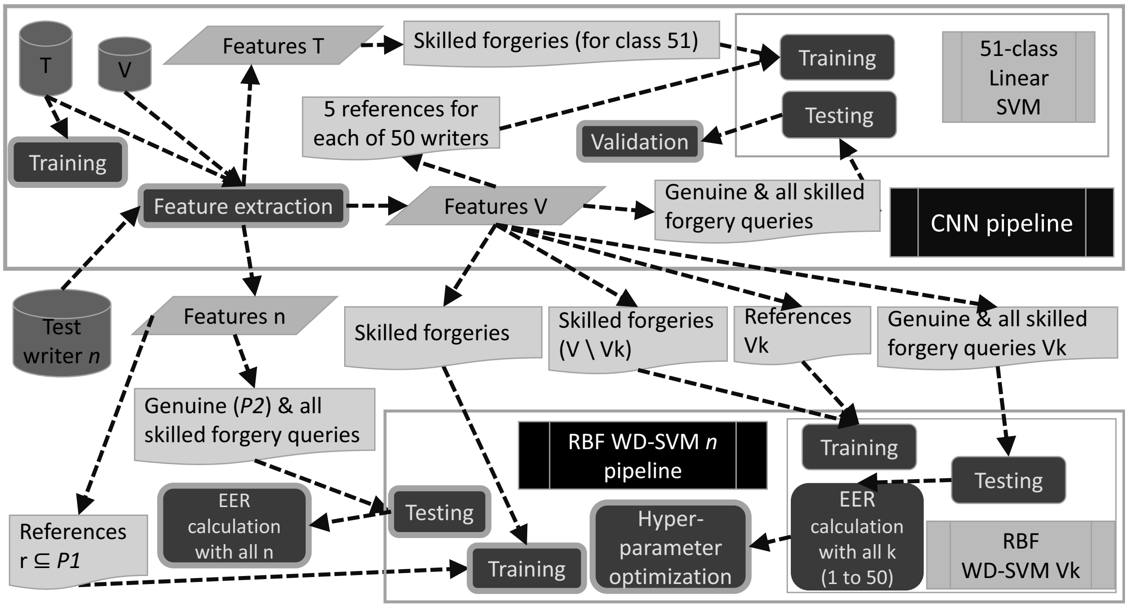


Fig. 3. Overview of the proposed method demonstrating the protocol.

Table 3. Test results on GPDS-300 [%].

# ref.	Resolution	Data aug.	$\text{EER}_{\text{global}}$ $K+1$	$\text{EER}_{\text{global}}$ $K\times 2$	EER_{user} $K+1$	EER_{user} $K\times 2$	Mean $\text{AUC}_{\text{global}}$ $K+1$	Mean $\text{AUC}_{\text{global}}$ $K\times 2$
5	150×220	–	3.71 ± 0.13	5.20 ± 0.13	2.45 ± 0.13	3.40 ± 0.08	98.45 ± 0.09	97.93 ± 0.12
		✓	4.17 ± 0.15	4.51 ± 0.09	2.74 ± 0.13	3.04 ± 0.14	98.08 ± 0.11	98.12 ± 0.10
	200×300	–	3.29 ± 0.09	4.38 ± 0.21	1.92 ± 0.09	2.65 ± 0.13	98.85 ± 0.09	98.45 ± 0.08
		✓	3.08 ± 0.12	3.33 ± 0.12	2.05 ± 0.18	2.07 ± 0.12	98.49 ± 0.12	98.73 ± 0.12
	250×375	–	3.28 ± 0.12	4.82 ± 0.17	1.78 ± 0.07	2.84 ± 0.13	99.07 ± 0.05	98.42 ± 0.06
		✓	2.96 ± 0.09	3.48 ± 0.14	1.86 ± 0.11	2.14 ± 0.13	98.76 ± 0.08	98.68 ± 0.10
12	150×220	–	3.14 ± 0.46	4.49 ± 0.07	1.77 ± 0.21	2.54 ± 0.27	99.09 ± 0.08	98.64 ± 0.11
		✓	3.31 ± 0.33	3.73 ± 0.23	1.92 ± 0.21	2.21 ± 0.29	99.01 ± 0.11	98.88 ± 0.11
	200×300	–	2.69 ± 0.23	3.62 ± 0.12	1.38 ± 0.15	1.92 ± 0.26	99.32 ± 0.11	98.98 ± 0.05
		✓	2.56 ± 0.16	2.93 ± 0.19	1.30 ± 0.25	1.46 ± 0.32	99.22 ± 0.10	99.31 ± 0.10
	250×375	–	2.73 ± 0.25	3.88 ± 0.35	$\mathbf{1.33\pm 0.13}$	1.96 ± 0.49	99.43 ± 0.13	99.02 ± 0.18
		✓	2.28 ± 0.04	3.03 ± 0.11	$\mathbf{1.12\pm 0.18}$	1.54 ± 0.31	99.34 ± 0.14	99.21 ± 0.16

Table 4. Effect of SF usage on training set T for CNN training on GPDS-300, as compared with SigNet (Hafemann *et al.*, 2017) [%].

	SF	EER_{user} # ref = 5	EER_{user} # ref = 12
SigNet	–	3.92 ± 0.18	3.15 ± 0.18
	✓	2.42 ± 0.24	1.69 ± 0.18
Proposed work w/o aug.	–	7.15 ± 0.19	6.00 ± 0.23
	✓	1.78 ± 0.07	$\mathbf{1.33\pm 0.13}$
Proposed work w/ aug.	–	5.02 ± 0.13	3.99 ± 0.35
	✓	1.86 ± 0.11	$\mathbf{1.12\pm 0.18}$

1024-dimensional features are first reduced to a size of 50 using PCA, then projected onto a 2-dimensional space. The perplexity value of 30 is used with an exact gradient calculation algorithm, besides other default parameters described.[1] Three networks trained with data augmentation are used for comparison. A 250×375 image size is used to extract features. t-SNE projections on V are shown in Fig. 4 for the three approaches, depicting the significance of the proposed $K+1$ model. SFs are much better separated than the other networks. While SFs for all writers are clustered as one group in $K\times 2$, they are separated into small groups in $K+1$, even though an opposite outcome is expected. The reason behind this phenomenon needs to be investigated.

[1] scikit-learn.org/dev/modules/generated/sklearn.manifold.TSNE

Table 5. Test results on CEDAR and MCYT [%].

	SF on CNN training	References CEDAR / MCYT	EER_{user} CEDAR	EER_{user} MCYT
(Hafemann *et al.*, 2017)	–	4 / 5	5.87 ± 0.73	3.58 ± 0.54
		12 / 10	4.76 ± 0.36	2.87 ± 0.42
	✓	4 / 5	5.92 ± 0.48	3.70 ± 0.79
		12 / 10	4.63 ± 0.42	3.00 ± 0.56
Proposed work w/o aug.	–	5	4.91 ± 0.36	6.88 ± 0.44
		12	4.41 ± 0.08	4.02 ± 0.85
	✓	5	5.21 ± 0.33	6.78 ± 0.38
		12	3.88 ± 0.19	4.04 ± 1.70
Proposed work w/ aug.	–	5	5.24 ± 0.52	6.05 ± 0.56
		12	4.39 ± 0.11	2.76 ± 1.38
	✓	5	4.23 ± 0.30	5.20 ± 0.27
		12	$\mathbf{3.41 \pm 0.11}$	$\mathbf{2.20 \pm 1.54}$

Table 6. Test results on online samples of GPDSsynthOn [%].

# Subjects in T	# ref.	$\text{EER}_{\text{global}}$		EER_{user}		Mean $\text{AUC}_{\text{global}}$	
		$K+1$	$K \times 2$	$K+1$	$K \times 2$	$K+1$	$K \times 2$
500	5	0.19 ± 0.02	0.91 ± 0.12	0.06 ± 0.02	0.29 ± 0.07	99.94 ± 0.02	99.81 ± 0.04
	12	0.11 ± 0.00	0.52 ± 0.13	0.01 ± 0.00	0.08 ± 0.05	99.99 ± 0.01	99.96 ± 0.04
1000	5	0.16 ± 0.03	0.60 ± 0.06	0.11 ± 0.02	0.14 ± 0.04	99.85 ± 0.03	99.89 ± 0.03
	12	0.10 ± 0.03	0.30 ± 0.03	0.06 ± 0.02	0.06 ± 0.02	99.92 ± 0.03	99.96 ± 0.00
2000	5	0.05 ± 0.01	0.24 ± 0.02	$\mathbf{0.01 \pm 0.01}$	0.07 ± 0.02	100.00 ± 0.00	99.97 ± 0.02
	12	0.04 ± 0.01	0.14 ± 0.01	0.01 ± 0.00	0.01 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
150	5	–		0.09 (Ahrabian and BabaAli, 2019)		–	

Offline SV results of the mixed model and the GPDSsynthOff model are shown in Table 7. It can be seen that training the CNN on the mixed set improved the test results for GPDSsynthOff, compared with only training on GPDSsynthOff. Only training on GPDSsynthOff degraded the results for GPDS-960 drastically. Training on the mixed set degraded GPDS-960 test results slightly (Table 3, 150×220, data augmentation, $K + 1$).

Comparison of the reported results with the literature can be found in Tables 8 and 9 for offline and online SVs, respectively. As can be seen from the results, our simple models can outperform or produce competing results compared with more complicated recent models present in the literature. For instance, our model achieves better results on the GPDS-300 offline dataset while giving competitive results on MCYT-75. On the other hand, a similar and simple CNN model that works on raw data for online SV achieves better results compared with other works.

For offline SV, our model achieves some of the best results for the GPDS-300 and MCYT-75. For GPDS-300 12 reference case, Hafemann *et al.* (2018) report 0.7% better EER with an approximately doubled number of parameters. For MCYT-75 and CEDAR, we report results close to the state-of-the-art although we never specifically trained for WI feature learning with those datasets. For GPDSsynthetic, Viana *et al.* (2023) report better results with an unknown number of parameters. Even when the number of parameters is higher compared with another model, it should be noted that the model is trained only once and inference is straightforward afterwards. On the other hand, our online model is similar to and based on the offline model which copes with images, needing many parameters. Having more parameters than in the compared work, our online model achieves the best results. In summary, our models can provide state-of-the-art results, usually with fewer parameters.

5.1. Running times. All simulations are performed on a computer with Intel(R) Core(TM) i7-8750H 2.20GHz CPU, 32 GB RAM, NVIDIA GeForce GTX 1070 GDDR5 8 GB graphics card and Windows 10 Pro 64 bit operating system. $K + 1$ CNN training times are 12548.33 seconds for the online model and 9561.48 seconds for the offline model. Feature extraction with the CNN takes approximately 3 milliseconds for one input, file-saving time included. The total time (training and then testing all test samples) spent with WD SVM per writer takes 0.60 seconds on the average, both for online and offline cases.

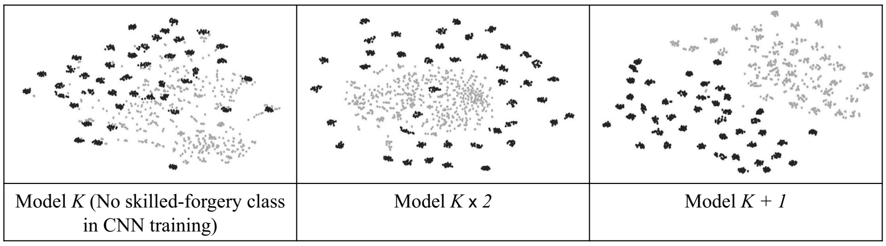


Fig. 4. t-SNE projections on V for different offline models. Genuine signatures are shown as darker and SFs are shown as lighter points.

Table 7. Offline test results of the mixed and GPDSsynthOff models [%].

Training set	Test set	# ref.	$\mathrm{EER}_{\mathrm{global}}$	$\mathrm{EER}_{\mathrm{user}}$	Mean $\mathrm{AUC}_{\mathrm{global}}$
Mixed	GPDS-960 (300)	5	5.35 ± 0.16	3.47 ± 0.07	98.04 ± 0.14
		12	4.47 ± 0.25	2.71 ± 0.45	98.69 ± 0.19
	GPDSsynthOff	5	12.92 ± 0.15	10.24 ± 0.17	93.24 ± 0.20
		12	9.99 ± 0.02	7.64 ± 0.24	95.51 ± 0.19
GPDSsynthOff	GPDS-960 (300)	5	24.15 ± 1.07	21.46 ± 1.04	82.20 ± 1.28
		12	22.00 ± 0.52	20.26 ± 0.28	83.74 ± 0.58
	GPDSsynthOff	5	14.22 ± 0.35	11.58 ± 0.27	91.96 ± 0.22
		12	11.18 ± 0.58	9.20 ± 0.20	94.67 ± 0.19

Table 8. Offline results comparison with the literature [%].

Method	Dataset	EER (# ref.s)
SigNet-SPP-F (Hafemann *et al.*, 2018)	GPDS-300	**0.41 (12)**
($\sim$37M parameters)	CEDAR	2.33 (10)
CBCapsNet (Parcham *et al.*, 2021)	CEDAR	0 (5)
(4.93M parameters)	GPDS-300	7.06 (5)
R-SigNet (Avola *et al.*, 2021)	CEDAR	0 (12)
(8.75M parameters)	MCYT-75	2.25 (8)
Static feature (Sadak *et al.*, 2022)	GPDS-100	9.90 (4)
	MCYT-75	11.55 (4)
WI (Longjam *et al.*, 2023)	CEDAR	**0 (WI)**
(43.12M parameters)	GPDS-300	10.16 (WI)
Feature augmentation (Arab *et al.*, 2023)	CEDAR	5.00 (5)
	MCYT-75	8.30 (5)
	GPDS-300	7.80 (5)
Multi-task framework (WD) (Viana *et al.*, 2023)	GPDSsynthetic (300)	**4.02 (5), 3.33 (12)**
	CEDAR	4.43 (5), 3.50 (12)
	MCYT-75	**4.07 (5)**
Proposed (WD, $K+1$, global threshold)	GPDSsynthetic (300)	12.92 (5), 9.99 (12)
	GPDS-300	2.96 (5), 2.28 (12)
Proposed (WD, $K+1$, writer thresholds)	GPDSsynthetic (300)	10.24 (5), 7.64 (12)
(19.41M CNN parameters)	CEDAR	4.23 (5), 3.41 (12)
	MCYT-75	5.20 (5), **2.20 (12)**
	GPDS-300	**1.86 (5)**, 1.12 (12)

Table 9. Online results comparison with the literature [%].

Method	Dataset	EER
OSVConTramer (Vorugunti *et al*., 2023) (188K parameters)	MCYT-100 SVC SUSIG	10.85 5.45 6.32
TSKD (1 ref.) (Sekhar *et al*., 2023) (6658 parameters)	MCYT-100 SVC SUSIG	12.42 6.45 11.32
Autoencoders (Ahrabian and BabaAli, 2019)	GPDSsynthOn	0.09
Proposed (WD, $K+1$, global threshold)	GPDSsynthOn (5 ref.) (12 ref.)	0.05 0.04
Proposed (same, user thold.) (10.32M parameters)	GPDSsynthOn (5 ref.) (12 ref.)	**0.01** **0.01**

6. Conclusions

In this work, we have shown that simpler and similar CNN models can still achieve state-of-the-art results both for offline and online SV while reducing the need for the amount of training data. For this purpose, a representation learning method for SV is demonstrated. Experiments on offline and online signatures show the effectiveness of the proposed work. The impact of the higher resolution than the train time resolution is investigated for offline signatures to extract features. As can be seen from the results, our representation learning strategy can outperform its counterparts even without utilizing data augmentation.

Similar CNN architectures are employed to build offline and online SVSs. Offline and online signature representations are directly learned from images and raw signals respectively, without relying on any handcrafted feature. While obtaining local and global features from raw data to feed to a neural network is a common approach used in recent works, we have shown that CNNs can be successfully used to learn online signature representations from sensory information.

Although the siamese architecture is a common way to detect forgeries, its accuracy dramatically drops when tested on different databases (Dey *et al*., 2017). Additionally, since it requires a pair of signatures as input, it is not efficient to train a siamese network on large databases. Our work showed that, in parallel with Hafemann *et al*. (2017), learned representations in one database can generalize successfully to other databases (Table 5). The proposed simple and similar models achieve state-of-the-art results for offline and online SV, reducing the need for more complicated and specialized models.

References

Ahrabian, K. and BabaAli, B. (2019). Usage of autoencoders and siamese networks for online handwritten signature verification, *Neural Computing and Applications* **31**(12): 9321–9334.

Arab, N., Nemmour, H. and Chibani, Y. (2023). A new synthetic feature generation scheme based on artificial immune systems for robust offline signature verification, *Expert Systems with Applications* **213**(Part C): 119306.

Avola, D., Bigdello, M.J., Cinque, L., Fagioli, A. and Marini, M.R. (2021). R-sigNet: Reduced space writer-independent feature learning for offline writer-dependent signature verification, *Pattern Recognition Letters* **150**(C): 189–196.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E. and Shah, R. (1994). Signature verification using a "siamese" time delay neural network, *in* J. Cowan *et al.* (Eds), *Advances in Neural Information Processing Systems*, Morgan Kaufmann, San Francisco, USA, pp. 737–744.

Calik, N., Kurban, O.C., Yilmaz, A.R., Yildirim, T. and Ata, L.D. (2019). Large-scale offline signature recognition via deep neural networks and feature embedding, *Neurocomputing* **359**(C): 1–14.

Devidas, S., Rao, Y.S. and Rekha, N.R. (2021). A decentralized group signature scheme for privacy protection in a blockchain, *International Journal of Applied Mathematics and Computer Science* **31**(2): 353–364, DOI: 10.34768/amcs-2021-0024.

Dey, S., Dutta, A., Toledo, J.I., Ghosh, S.K., Lladós, J. and Pal, U. (2017). SigNet: Convolutional siamese network for writer independent offline signature verification, *arXiv:* 1707.02131.

Diaz, M., Ferrer, M.A., Impedovo, D., Malik, M.I., Pirlo, G. and Plamondon, R. (2019). A perspective analysis of handwritten signature technology, *ACM Computing Surveys* **51**(6): 1–39.

Ferrer, M.A., Diaz, M., Carmona-Duarte, C. and Morales, A. (2016). A behavioral handwriting model for static and dynamic signature synthesis, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**(6): 1041–1053.

Fierrez, J., Ortega-Garcia, J., Ramos, D. and Gonzalez-Rodriguez, J. (2007). HMM-based on-line signature verification: Feature extraction and signature modeling, *Pattern Recognition Letters* **28**(16): 2325–2334.

Giazitzis, A. and Zois, E. (2024). SigmML: Metric meta-learning for writer independent offline signature verification in the space of SPD matrices, *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, Hawaii, United States*, pp. 6300–6310.

Hafemann, L.G., Oliveira, L.S. and Sabourin, R. (2018). Fixed-sized representation learning from offline handwritten signatures of different sizes, *International Journal on Document Analysis and Recognition (IJDAR)* **21**(3): 219–232.

Hafemann, L.G., Sabourin, R. and Oliveira, L.S. (2017). Learning features for offline handwritten signature verification using deep convolutional neural networks, *Pattern Recognition* **70**(C): 163–176.

Hameed, M.M., Ahmad, R., Kiah, L.M., Murtaza, G. and Mazhar, N. (2023). OffSig-SinGAN: A deep learning-based image augmentation model for offline signature verification, *Computers, Materials & Continua* **76**(1): 1267–1289.

Hameed, M.M., Ahmad, R., Kiah, M.L.M. and Murtaza, G. (2021). Machine learning-based offline signature verification systems: A systematic review, *Signal Processing: Image Communication* **93**: 116139.

Impedovo, D. and Pirlo, G. (2008). Automatic signature verification: The state of the art, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **38**(5): 609–635.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift, *arXiv:* 1502.03167.

Ji, X., Suehiro, D. and Uchida, S. (2023). Paired contrastive feature for highly reliable offline signature verification, *Pattern Recognition* **144**(C): 109816.

Kalera, M.K., Srihari, S. and Xu, A. (2004). Offline signature verification and identification using distance statistics, *International Journal of Pattern Recognition and Artificial Intelligence* **18**(07): 1339–1360.

Kaur, H.P. and Kumar, M. (2021). Signature identification and verification techniques: State-of-the-art work, *Journal of Ambient Intelligence and Humanized Computing* **14**: 1027–1045.

Kholmatov, A. and Yanikoglu, B. (2005). Identity authentication using improved online signature verification method, *Pattern Recognition Letters* **26**(15): 2400–2408.

Kingma, D.P. and Ba, J. (2014). Adam: A method for stochastic optimization, *arXiv:* 1412.6980.

Lai, S., Jin, L. and Yang, W. (2017). Online signature verification using recurrent neural network and length-normalized path signature descriptor, *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Vol. 01, Kyoto, Japan, pp. 400–405.

Lecun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998). Gradient-based learning applied to document recognition, *Proceedings of the IEEE* **86**(11): 2278–2324.

Li, H., Wei, P. and Hu, P. (2021). Static-dynamic interaction networks for offline signature verification, *Proceedings of the AAAI Conference on Artificial Intelligence* **35**(3): 1893–1901.

Longjam, T., Kisku, D.R. and Gupta, P. (2023). Writer independent handwritten signature verification on multi-scripted signatures using hybrid CNN-BiLSTM: A novel approach, *Expert Systems with Applications* **214**(C): 119111.

Maaten, L.v.d. and Hinton, G. (2008). Visualizing data using t-SNE, *Journal of Machine Learning Research* **9**(Nov): 2579–2605.

Minaee, S., Abdolrashidi, A., Su, H., Bennamoun, M. and Zhang, D. (2023). Biometrics recognition using deep learning: A survey, *Artificial Intelligence Review* **56**(8): 8647–8695.

Müller, R., Kornblith, S. and Hinton, G.E. (2019). When does label smoothing help?, *in* H.M. Wallach *et al.* (Eds), *Advances in Neural Information Processing Systems, Red Hook, NY, USA*, pp. 4694–4703.

Ortega-Garcia, J., Fierrez-Aguilar, J., Simon, D., Gonzalez, J., Faundez-Zanuy, M., Espinosa, V., Satue, A., Hernaez, I., Igarza, J.-J., Vivaracho, C., Escudero, D. and Moro, Q.-I. (2003). MCYT baseline corpus: A bimodal biometric database, *IEE Proceedings-Vision, Image and Signal Processing* **150**(6): 395–401.

Parcham, E., Ilbeygi, M. and Amini, M. (2021). Cbcapsnet: A novel writer-independent offline signature verification model using a cnn-based architecture and capsule neural networks, *Expert Systems with Applications* **185**(C): 115649.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L. *et al.* (2019). PyTorch: An imperative style, high-performance deep learning library, *in* H. Wallach *et al.* (Eds), *Advances in Neural Information Processing Systems*, Curran Associates Inc., Red Hook, NY, pp. 8026–8037.

Putz-Leszczynska, J. (2015). Signature verification: A comprehensive study of the hidden signature method, *International Journal of Applied Mathematics and Computer Science* **25**(3): 659–674, DOI: 10.1515/amcs-2015-0048.

Qiao, Y., Liu, J. and Tang, X. (2007). Offline signature verification using online handwriting registration, *IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, United States*, pp. 1–8.

Radhika, K. and Gopika, S. (2015). Online and offline signature verification: A combined approach, *Procedia Computer Science* **46**(C): 1593–1600.

Ren, J.-X., Xiong, Y.-J., Zhan, H. and Huang, B. (2023). 2c2s: A two-channel and two-stream transformer based framework for offline signature verification, *Engineering Applications of Artificial Intelligence* **118**(C): 105639.

Sadak, M.S., Kahraman, N. and Uludağ, U. (2022). Dynamic and static feature fusion for increased accuracy in signature verification, *Signal Processing: Image Communication* **108**(C): 116823.

Sekhar V., C., Gautam, A., Pulabaigari, V., S.R., S. and Sai G., R.K. (2023). TSOSVNet: Teacher-student collaborative knowledge distillation for online signature verification, *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France*, pp. 742–751.

Tolosana, R., Vera-Rodriguez, R., Fierrez, J. and Ortega-Garcia, J. (2018). Exploring recurrent neural networks for on-line handwritten signature biometrics, *IEEE Access* **6**: 5128–5138.

Tolosana, R., Vera-Rodríguez, R., Fiérrez, J. and Ortega-Garcia, J. (2021). Deepsign: Deep on-line signature verification, *IEEE Transactions on Biometrics, Behavior, and Identity Science* **3**(2): 229–239.

Touvron, H., Vedaldi, A., Douze, M. and Jégou, H. (2019). Fixing the train-test resolution discrepancy, *in* H. Wallach *et al.* (Eds), *Advances in Neural Information Processing Systems*, Curran Associates Inc., Red Hook, NY, pp. 8252–8262.

Uppalapati, D. (2007). *Integration of Offline and Online Signature Verification Systems*, Master's thesis, IIT Kanpur.

Vargas, F., Ferrer, M., Travieso, C. and Alonso, J. (2007). Off-line handwritten signature GPDS-960 corpus, *9-th International Conference on Document Analysis and Recognition (ICDAR 2007)*, Vol. 2, Curitiba, Brazil, pp. 764–768.

Viana, T.B., Souza, V.L., Oliveira, A.L., Cruz, R.M. and Sabourin, R. (2023). A multi-task approach for contrastive learning of handwritten signature feature representations, *Expert Systems with Applications* **217**(C): 119589.

Vorugunti, C., Gautam, A. and Pulabaigari, V. (2023). Osvcontramer: A hybrid cnn and transformer based online signature verification, *Proceedings of the IEEE International Joint Conference on Biometrics (IJCB), Ljubljana, Slovenia*, pp. 1–10.

Vorugunti, C.S., Devanur S.G., Mukherjee, P. and Pulabaigari, V. (2019). Osvnet: Convolutional siamese network for writer independent online signature verification, *International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia*, pp. 1470–1475.

Xie, L., Wu, Z., Zhang, X. and Li, Y. (2023). Synchronous spatio-temporal signature verification via fusion triplet supervised network, *Engineering Applications of Artificial Intelligence* **123**(B): 106378.

Xie, L., Wu, Z., Zhang, X., Li, Y. and Wang, X. (2022). Writer-independent online signature verification based on 2D representation of time series data using triplet supervised network, *Measurement* **197**(6): 111312.

Xiong, Y.-J., Cheng, S.-Y., Ren, J.-X. and Zhang, Y.-J. (2023). Attention-based multiple siamese networks with primary representation guiding for offline signature verification, *International Journal on Document Analysis and Recognition* **27**(2): 195–208.

Yılmaz, M.B. and Öztürk, K. (2019). Recurrent binary patterns and CNNs for offline signature verification, *in* A. Kohei *et al.* (Eds) *Proceedings of the Future Technologies Conference*, San Francisco, United States, pp. 417–434.

Yılmaz, M.B. and Yanıkoğlu, B. (2016). Score level fusion of classifiers in off-line signature verification, *Information Fusion* **32**(B): 109–119.

Yu, H. and Shi, P. (2023). A novel deep ensemble framework for online signature verification using temporal and spatial representation, *in* D. Wang *et al.* (Eds), *Information and Communications Security*, Springer, Singapore, pp. 534–549.

Yılmaz, M.B. and Öztürk, K. (2018). Hybrid user-independent and user-dependent offline signature verification with a two-channel CNN, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, United States,* pp. 526–534.

Zagoruyko, S. and Komodakis, N. (2015). Learning to compare image patches via convolutional neural networks, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, United States*, pp. 4353–4361.

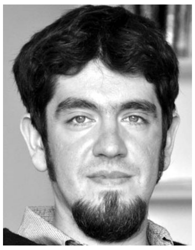
Zhang, X., Wang, Y., Sun, W., Cui, Q. and Wei, X. (2023). Multi-path attention inverse discrimination network for offline signature verification, *Intelligent Automation & Soft Computing* **36**(3): 3057–3071.

Zimmer, A. and Ling, L.L. (2008). Offline signature verification system based on the online data, *EURASIP Journal on Advances in Signal Processing* **2008**(1): 492910.

Zois, E.N., Alexandridis, A. and Economou, G. (2019). Writer independent offline signature verification based on asymmetric pixel relations and unrelated training-testing datasets, *Expert Systems with Applications* **125**(C): 14–32.

Zois, E.N., Said, S., Tsourounis, D. and Alexandridis, A. (2023). Subscripto multiplex: A Riemannian symmetric positive definite strategy for offline signature verification, *Pattern Recognition Letters* **167**: 67–74.

Zois, E.N., Theodorakopoulos, I., Tsourounis, D. and Economou, G. (2017). Parsimonious coding and verification of offline handwritten signatures, *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, Hawaii, United States,* pp. 636–645.



Mustafa Berkay Yılmaz is an assistant professor at Akdeniz University. He received his BS degree from the Department of Computer Engineering at Bahçeşehir University, his MSc degree from the Department of Mechatronics at Sabancı University, and his PhD degree from the Department of Computer Science and Engineering at Sabancı University, in 2007, 2009 and 2015, respectively. His research interests focus on computer vision, biometrics and machine learning.

Kağan Öztürk is a PhD candidate at the University of Notre Dame. He received his BS degree from the Department of Electronics and Communication Engineering at Yıldız Teknik University, and his MSc degree from the Department of Computer Engineering at Akdeniz University, in 2015 and 2020, respectively. His research interests focus on computer vision, machine learning, biometrics and face recognition.

Received: 13 August 2024
Revised: 15 December 2024
Re-revised: 4 February 2025
Accepted: 14 February 2025