IET Biometrics

Research Article

ISSN 2047-4938
Received on 14th February 2020
Revised 12th May 2020
Accepted on 9th July 2020
E-First on 16th September 2020
doi: 10.1049/iet-bmt.2020.0032
www.ietdl.org

DeepFuseOSV: online signature verification using hybrid feature fusion and depthwise separable convolution neural network architecture

Chandra Sekhar Vorugunti[1] ✉, Viswanath Pulabaigari[1], Prerana Mukherjee[1], Abhishek Sharma[2]

[1] Computer Science and Engineering, Indian Institute of Information Technology-SriCity, Andhra Pradesh, India

[2] Electronics and Communication, Indian Institute of Information Technology-Naya Raipur, Chattisgarh, India

✉ *E-mail: chandrasekhar.v@iiits.in*

Abstract: Online signature verification (OSV) is a widely utilised technique in the medical, e-commerce and m-commerce applications to lawfully bind the user. These high-speed systems demand faster writer verification with a limited amount of information along with restrictions on training and storage cost. This study makes two major contributions: (i) A competent feature fusion technique in which traditional statistical-based features are fused with deep representations from a convolutional auto-encoder; and (ii) a hybrid architecture combining depth-wise separable convolution neural network (DWSCNN) and long short term memory (LSTM) network delivering state-of-the-art performance for OSV is proposed. DWSCNN is utilised for extracting deep feature representations and LSTM is competent in learning long term dependencies of stroke points of a signature. This hybrid combination accomplishes better classification accuracy (lower error rates) even with one-shot learning, i.e. achieving higher classification accuracies with only one training signature sample per user. The authors have extensively evaluated their model using three widely used datasets MCYT-100, SVC and SUSIG. These exhaustive experimental studies confirm that the DeepFuseOSV framework results in the state-of-the-art outcome by achieving an equal error rate (EER) of 13.26, 2.58, 0.07% in Skilled 1, Skilled 10 and Random 10 categories of MCYT-100, respectively, 7.71% in Skilled 1 category of SVC, 1.70% in Random 1 category of SUSIG.

1 Introduction

The traditional writer verification techniques like passwords, patterns, personal identification numbers etc. are prone to risks like stealing or forgetting them. Hence, online signature verification (OSV) is widely used in financial, e-commerce and m-commerce applications to legally bind an individual by detecting the forgery of online signatures [1–3]. As presented in Fig. 1, online signature encompasses a combination of static features (x, y co-ordinates) and dynamic features (pressure, azimuthal angle etc.) which makes them more efficient in representing writer characteristics compared to offline signatures [1, 4, 5].

OSV is a promising and stimulating area of research in the field of handwritten text recognition using artificial intelligence. In the literature, several studies on OSV model/frameworks exist which can be approximately characterised into (i) local feature-based; (ii) global feature-based; (iii) function-based; and (iv) deep learning-based [6]. In local feature-based technique, the features are captured at each stroke point $[x, y, p, \theta, \propto]$ i.e. x, y co-ordinates, pressure, azimuthal angle, the tilt angle of the device etc. are used to classify a test signature. In global features-based OSV, the features are extracted by considering the whole signature, e.g.; number of strokes/lognormal, the standard deviation of x-coordinate etc. The feature-based (local/ global) OSV frameworks learn intra-class variations of online signatures. In function-based OSV, the local features at each stroke point of a reference and test signature are compared. Based on the predefined threshold, the test signature is delegated as genuine or counterfeit. The other function-based OSV systems incorporate interval-valued classifier

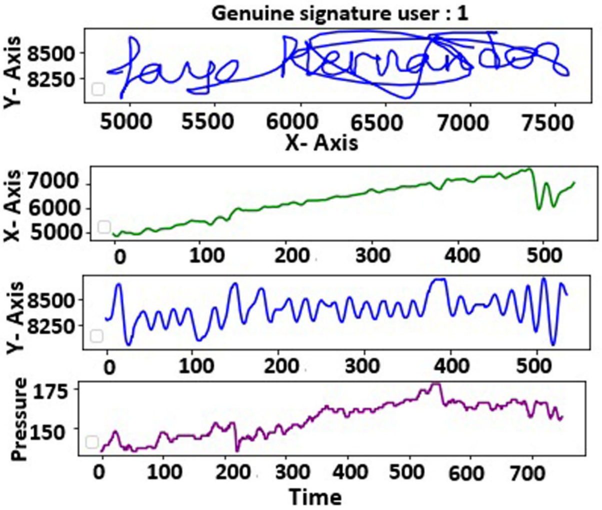


Fig. 1 *Example online signature*

Table 1 Complete particulars of datasets used in the experimental evaluation of proposed framework

DataSet→	SVC	SUSIG	MCYT
total number of users	40	94	100
features per signature	47	47	100
genuine samples per user	20	20	25
forgery samples per user	20	10	25
total real signatures	800	1880	2500
total of fake signatures	800	940	2500

[7, 8], hidden Markov model (HMM) [4], dynamic time warping [1, 9–12], edit distance [13], neuro fuzzy [14], Gaussian mixture models [1], support vector machine [15] and neural machine [16], long short term memory (LSTM) [12, 13, 17–20] etc.

The paper is organised in this fashion. In Section 2, we discuss the related work. Section 3 discusses about the contribution of our proposed work. Preliminaries are discussed in Section 4. Proposed 'DeepFuseOSV' framework and its basic constituents are discussed in Section 5. In Section 6, particulars of training, testing, experimental investigation of our framework with most recent and best in class systems are examined. Conclusions are specified in Section 7.

2 Related work

Very few OSV systems are proposed utilising deep learning techniques. Recently Ruben *et al.* [21] structured a novel semi-supervised system for assessing the signature complexity utilising recurrent neural networks (RNNs). The proposed model is utilised to assign a complexity label to each signature.

In our previous work [22], we defined a deep convolutional Siamese network-based OSV framework. In the case of one-shot learning, the model resulted in an EER of 21.84%. As Siamese network is a twin network, the main pitfall in Siamese networks compared to CNN-based OSV frameworks is: (i) requiring a large number of parameters to train the framework. (ii) For effective learning about the differences between genuine and forgery pairs, for each probe signature, Siamese networks demand a huge number of signature pairs, i.e. (genuine, genuine) and (genuine, forgery). This results in two main drawbacks, the higher training time of the network and the output from these networks cannot be fed to conventional classifiers. To overcome these pitfalls, in this work, we have adapted the combination of depth-wise separable convolution-based CNN (DWSCNN) plus LSTM framework. DWSCNN results in a reduced number of parameters and LSTM networks learn the inter and intra writer variations of signature sequences effectively.

2.1 Essential of one/few shot learning in OSV

In the context of online signatures, it is impossible and difficult to obtain an adequate number of signature samples from the users. This can be credited to the sensitivity of applications with which the signatures are associated, e.g. m-banking and m-payment [12, 23, 24] etc. As depicted in Table 1, the maximum genuine and forgery signature samples available per user is 25 which is in the case of MCYT dataset. Therefore, the OSV system must fulfil the vital prerequisite: The system must learn intra and inter personal variants with few signature samples at least as one.

Relatively very few works have been proposed with regards to OSV systems with one/few-shot learning, i.e. learning the writer dependent characteristics with one/few signature patterns. Galbally *et al.* [1] formulated a HMM-based OSV technique, in which plausible signature samples are produced from one real signature. On similar lines, Diaz *et al.* [25–27] designed an OSV model, in which, sigma-lognormal parameters are figured from a single real signature sample, and plausible samples are created by mimicking the writer hand movements. Very recently, Rousseeuw and Croux [28] proposed an OSV by formulating a system which simulates the writer's skeletal arm. A new set of features is computed positioned on the movement of the wrist, elbow and shoulder combination.

The above few-shot learning-based OSV frameworks [1, 25–28] experiences the accompanying disadvantages:

(i) These models are not widely investigated on multiple data sets.

(ii) These models are not assessed on every single category, i.e. Skilled_01(S_01), Skilled_05 (S_05), Skilled_10 (S_10), …, S_(N −5), Random_01 (R_01), Random_05 (R_05), Random_10 (R_10), …, R_(N−5), where 'N' = number of genuine signature samples per user. In general, the above models are assessed on only S_01 or S_05 categories.

In this context, in our exploratory work [29], we have given a first of its attempt to address the above pitfalls. Very limited and insufficient contribution in one/few-shot learning-based OSV motivated us to contribute to this work. To address the above difficulties and disadvantages of real-time OSV systems, we propose DeepFuseOSV that can realise a few-shot learning, including one-shot learning.

3 Major contributions of our work

The novel contribution of this work is the use of hybrid feature fusion and hybrid LSTM, CNN architecture to achieve one/few-shot learning.

1. A feature-level fusion is implemented, where in a representation feature vector is formed using handcrafted features and deep features returned by a convolution autoencoder. The resultant feature vector utilises the full advantage of both handcrafted and deep-learned features and resulted in the enhanced discriminating power of DeepFuseOSV to classify a test signature.
2. We have effectively utilised a novel hybrid convolutional neural network which is a stacking of DWSCNN and LSTM layers. The hybrid set of features from step 1 is given as an input to the deep DWSCNN. The depth-wise separable layers learn a complete short-term spatial relationship and LSTM layers learn long term temporal dependencies for effective test signature classification from a lesser number of samples as minimum as one.
3. To make the empirical justification of our findings, we have led a broad evaluation of our proposed OSV system by implementing experiments on the three most widely used datasets, i.e. MCYT-100, SVC and SUSIG.
4. We have appraised the superiority of the proposed framework by comparing the proposed OSV framework with the modern and state of the art frameworks. It is found to exceed the recent state of the artworks, which will be discussed in subsequent sections.

4 Preliminaries

4.1 Long short-term memory

To acquire the long-term dependencies between stroke successions of an online signature, the framework must be able to associate critical patterns such as peaks and valleys to comprehend the present stroke patterns.

LSTM is a proficient model to control the information flow, by fixing the amount of time to store, delete the previous stroke patterns and correlate the stored information to the new signature patterns to classify the test signature, which is an advantage compared to other sequence models like RNNs, HMM etc. [3, 27]. Fig. 2*a*, portrays the variations of pressure profiles of the first three users of MCYT-100 dataset.

It is clear that the genuine signature profiles have low inter writer variations and high intra writer variations. Fig. 3*b* illustrates the genuine and skilled forgery signatures of user-19 of MCYT-100 dataset. The last subfigure of Fig. 3*b* illustrates the plotting of pressure and azimuthal angle signals of genuine and skilled forgery signature. Both the genuine profiles are tightly overlapped with corresponding skilled forgery profiles, which reflects the real-time scenario. To classify a test signature, the framework must learn the dependencies/variations/structure of patterns of each profile end to end. Learning a subset of patterns is inadequate to classify a test signature. Hence, in this work, we have used LSTM layers to learn the long-term dependencies of signature strokes. LSTM units contain three gates and a cell. The functional equations of LSTM are given as follows [19, 20]:

$$f_t = \text{sig}\,(W_\text{f} \cdot [\text{hidden}_{t-1}, x_t] + \text{bias}_\text{f}) \tag{1}$$

$$i_t = \text{sig}(W_\text{i} \cdot [\text{hidden}_{t-1}, x_t] + \text{bias}_\text{i}) \tag{2}$$

$$C_t^{\sim} = \tanh(W_\text{c} \cdot [\text{hidden}_{t-1}, x_t]x_t + \text{bias}_\text{c}) \tag{3}$$

$$C_t = f_t \times C_{t-1} + i_t \times C_t^{\sim} \tag{4}$$

$$o_t = \text{sig}(W_\text{o}[\text{hidden}_{t-1}, x_t] + \text{bias}_\text{o}) \tag{5}$$

$$h_t = o_t \times \tan h(C_t) \tag{6}$$

where sig is the Sigmoid function, 't' is the current time. $'x_t'$ represents the input at the current time $'t'$. $'\text{hidden}'_{t-1}$, $'\text{hidden}'_t$ represents the output from the previous and current block, respectively. 'i','o', 'f' represent the input, output and forget gates, respectively. $'C'_{t-1}$,$'C'_t$ represent the memory from previous and current blocks, respectively. W_i, W_f, W_o, W_c represent the weights and $b_\text{i}, b_\text{f}, b_\text{o}, b_\text{c}$ are biases of LSTM to be learned during training at the input, forget and output gates, respectively. tanh is the hyperbolic tangent function, ×, + represent element-wise multiplication and summation, respectively.

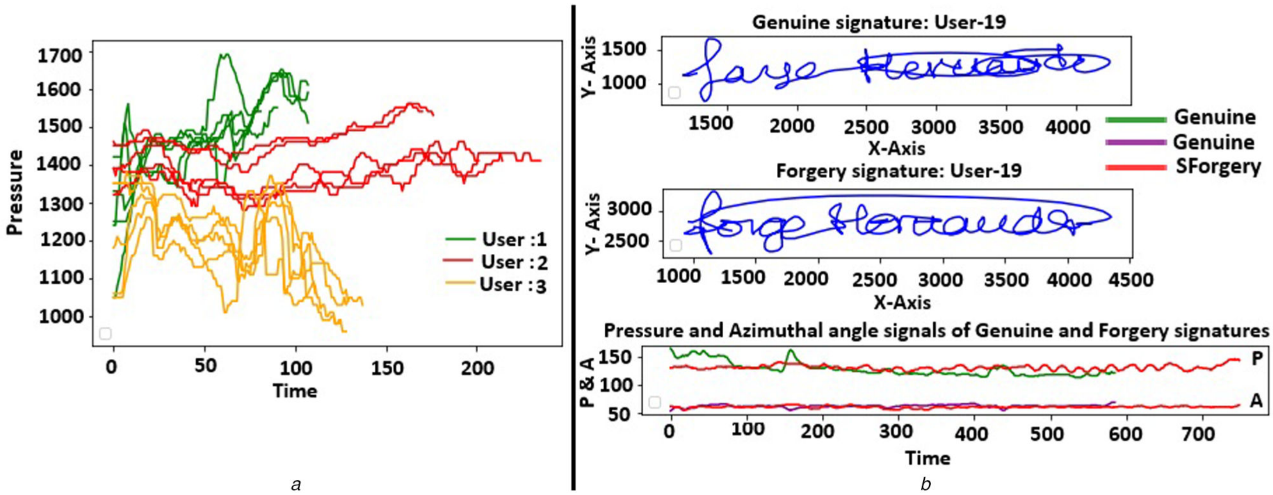
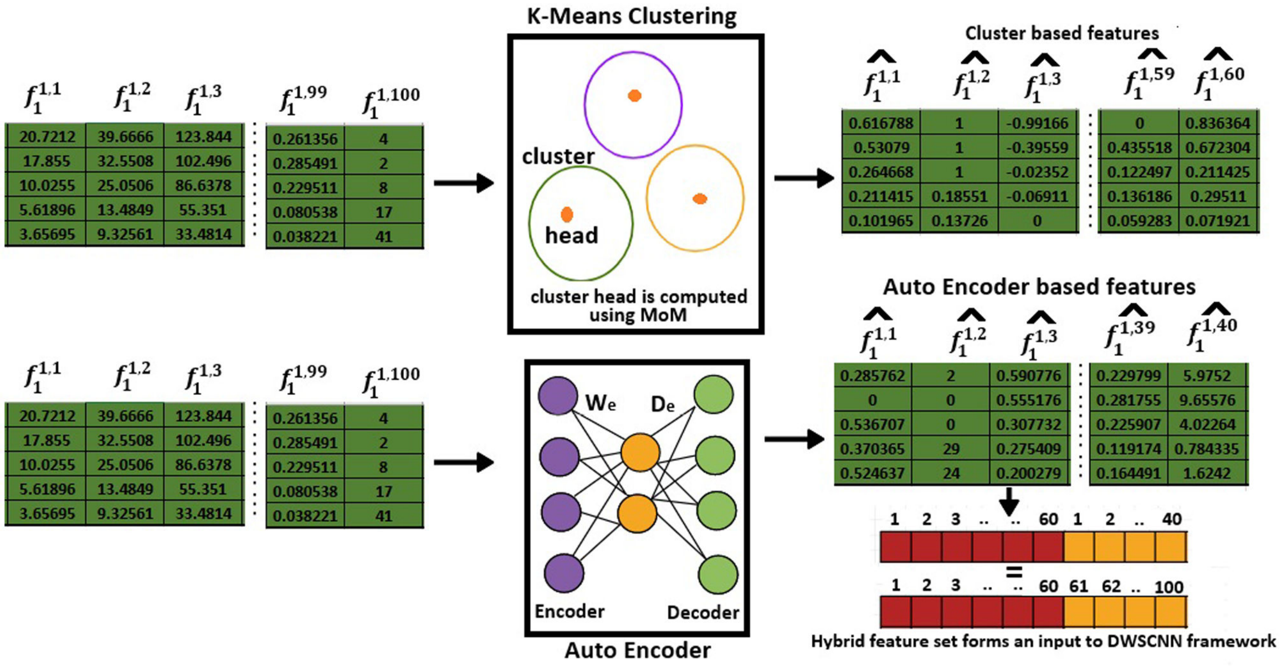


Fig. 2 *The intra, inter writer variations and pressure profiles in online signatures*
(a) The intra and inter writer variations (dependencies) of pressure signals of different users, ***(b)*** The pressure (p) and azimuthal (A) angle signals of a genuine and skilled forgery signature of user-19



$f_1^{1,1}$	$f_1^{1,2}$	$f_1^{1,3}$		$f_1^{1,99}$	$f_1^{1,100}$
20.7212	39.6666	123.844	:	0.261356	4
17.855	32.5508	102.496	:	0.285491	2
10.0255	25.0506	86.6378	:	0.229511	8
5.61896	13.4849	55.351	:	0.080538	17
3.65695	9.32561	33.4814	:	0.038221	41

K-Means Clustering

cluster head is computed using MoM

Cluster based features

$\hat{f}_1^{1,1}$	$\hat{f}_1^{1,2}$	$\hat{f}_1^{1,3}$		$\hat{f}_1^{1,59}$	$\hat{f}_1^{1,60}$
0.616788	1	-0.99166	:	0	0.836364
0.53079	1	-0.39559	:	0.435518	0.672304
0.264668	1	-0.02352	:	0.122497	0.211425
0.211415	0.18551	-0.06911	:	0.136186	0.29511
0.101965	0.13726	0	:	0.059283	0.071921

$f_1^{1,1}$	$f_1^{1,2}$	$f_1^{1,3}$		$f_1^{1,99}$	$f_1^{1,100}$
20.7212	39.6666	123.844	:	0.261356	4
17.855	32.5508	102.496	:	0.285491	2
10.0255	25.0506	86.6378	:	0.229511	8
5.61896	13.4849	55.351	:	0.080538	17
3.65695	9.32561	33.4814	:	0.038221	41

Auto Encoder based features

$\hat{f}_1^{1,1}$	$\hat{f}_1^{1,2}$	$\hat{f}_1^{1,3}$		$\hat{f}_1^{1,39}$	$\hat{f}_1^{1,40}$
0.285762	2	0.590776	:	0.229799	5.9752
0	0	0.555176	:	0.281755	9.65576
0.536707	0	0.307732	:	0.225907	4.02264
0.370365	29	0.275409	:	0.119174	0.784335
0.524637	24	0.200279	:	0.164491	1.6242

Hybrid feature set forms an input to DWSCNN framework

Fig. 3 *Overview of hybrid feature set which forms as input to our DeepFuseOSV framework*

Table 2 Set of global features computed for each signature

total duration of signature	$(X_{\text{1st-pendown}} - x_{\text{min}})/\Delta x$	local maxima in y
number of pen ups	$(Y_{\text{1st-pendown}} - Y_{\text{min}})/\Delta y$	local maxima in x
sign changes of dx/dt, dy/dt	$(X_{\text{last-pendown}} - x_{\text{max}})/\Delta x$	$(X_{\text{max}} - X_{\text{min}})/X_{\text{accquisitionrange}}$
average jerk	$(Y_{\text{last-pendown}} - Y_{\text{max}})/\Delta y$	(max distance between two points)/(A_{min})
standard deviation of $v_y/\Delta y$	(Acceleration rms a)/a_{max}	θ (initial direction).
standard deviation of acceleration in 'x' and 'y' directions	StdDev of x/ Δx	standard deviation of velocity in 'x' and 'y' directions.
histogram based features	StdDev of $y/\Delta y$	θ (before last pen up).

Complete set of features is available in [32].

4.2 Intuition behind fusion of handcrafted and deep representative features

The handcrafted features are computed based on certain statistical properties or pre-defined algorithm, which reflects the domain-specific characteristics of an input signature, e.g. number of pen ups, average velocity of a pen in the *X*-direction etc. In deep CNNs, the initial convolution layers extract low-level latent representations from the input signature. The lower level features extract local spatial relationships, which are quite comprehensive and independent of any definitive classification task. Based on the low-level latent representations, the middle layers of CNN compute intermediate features, representing the underlying contextual structure of the output [30]. The final layer extracts the higher-level features from intermediate features through complex non-linear transformation. The high level deep representational features are more sophisticated, through which it learns intra-class variability of an input signature. Hence, CNN learns the feature representations incrementally and low-level features form an essential part of learning for CNN. The representation ability of CNN increases with the increase of the depth of the layers [31].

As depicted in Fig. 3, the feature vector which is computed based on the statistical properties detailed in Table 2 represents the global characteristics of a signature.

The global feature vector forms an input to both the clustering algorithm and convolutional auto-encoder (CAE). The clustering algorithm, outcomes the best domain-specific set of features as cluster heads. The CAE outcomes deep representative features which represent the spatial, local and temporal dependencies among the global features. The fusion of output features from the clustering algorithm (domain-specific) and the CAE (context-aware) compliment each other and increase the discriminating capability compared to individual sets of features.

5 Proposed deepFuseOSV architecture

5.1 Writer dependent feature clustering

Let $S_k = [S_k^1, S_k^2, S_k^3, \ldots, S_k^p]$ represents 'p' signature samples of writer 'k', where 'k' ranges from 1 to T. 'T' means the total number of writers in the dataset. A signature is categorised as a set of 'm' global features $= S_k^1 = \{f_{k,1}^1,\ f_{k,1}^2, f_{k,1}^3, \ldots, f_{k,1}^m\}$.

Let $\boldsymbol{F}_k = [F_k^1,\ F_k^2,\ F_k^3, \ldots, F_k^m]$ represents 'm' feature vectors of writer 'k'. $\boldsymbol{F}_k^j = [f_{k,1}^j,\ f_{k,2}^j, f_{k,3}^j, \ldots, f_{k,p}^j]^{\mathrm{T}}$ is a column vector representing the jth feature of signature samples of writer 'k'. 'p' represents the total number of signature samples per user. We have applied *K*-means clustering on a set of feature vectors $\boldsymbol{F}_k = [F_k^1, F_k^2, F_k^3, \ldots, F_k^m]$ to aggregate similar feature vectors. For each cluster, a cluster representative which best represents the entire set of features in a cluster is selected, where 'K' is set to 60% of the total number of features.

Subject to MCYT, the feature vectors are grouped into '60' clusters and in case of SVC and SUSIG into '30' clusters. To pick the best percentage of clusters from the total set of features, we have led a careful experimental analysis by varying the percentage of the number of clusters from 20 to 80% of the total number of features and EER resulted with respect to Skilled 1 and Random 1 categories for each dataset is recorded. The percentage of clusters which brought about least EER is considered as final clustering value for experimentation process, i.e. 60%.

5.2 Computing the cluster representatives

The statistical measures like median, mode and mean absolute difference etc. suffer from major pitfalls like less representative capabilities in the context of time series data like online signatures, impact of fluctuations of sampling, effective for symmetric distributions etc. Subsequently, to choose one feature as a cluster representative, we have chosen median of medians (MoM) which was proposed by Rousseeuw and Croux [28]

$$\mathrm{MoM_n} = c\ \mathrm{med}_p\{\mathrm{med}_q|x_p -\ x_q|\}\,. \tag{7}$$

For each 'p', compute the median of $\{|x_p -\ x_q|; q = 1, 2, \ldots, m\}$, which outputs '$m$' values. The median of these set of 'm' values provides the final value of MoM, (the factor c is a consistency factor, and its default value is 1.1926.). MoM indicates the typical distance between the features of a cluster. The computational complexity of MoM is $O(m \log m)$. The user is advised to read [28] for further analysis of MoM. To choose the element which best qualifies as a representative of the corresponding cluster, for each feature, we have calculated MoM based on (7). The feature which has minimum MoM value is picked as a head of the cluster.

5.3 Input signature format

As shown in Fig. 3, the input online signature is represented as a row vector of dimension 1×100 subject to MCYT and 1×47 subject to SVC and SUSIG dataset. The dimensions 100, 47 means the total number of global features for each signature. Subject to MCYT, the feature vectors are grouped into '60' clusters and in context of SVC and SUSIG into '30' clusters. A statistical dispersion metric MoM is utilised to compute the cluster representative for each cluster. Top 60% of features, i.e. (60, 30) are chosen depending on the above criteria. The same set of global features is fed into a CAE, which outputs the latent representations of size 80 for MCYT and 40 for {SUSIG, SVC} datasets. Out of final representations, top 50% of the features (40,17) are selected and combined with the set of cluster representatives which results a hybrid set of features of length (100,47) and these joint representations form an input to our proposed DWSCNN framework for classification.

5.4 Separable convolution layer

As depicted in Fig. 4, our proposed DWSCNN comprise a group of six layers. The primary layers are depth wise separable convolutional and batch normalisation layers. As illustrated in Fig. 3, an online signature of dimension (1×100) is given to the first DWS convolution layer. Each DWS convolution layer includes 64 filters of dimension 1×3 which results in three feature maps of size 1×100. A 1×1 pointwise convolution operation is implemented on these feature maps, which yields a feature vector of dimension 100×64.

To standardise the input to each layer with zero mean/unit variance, to balance out the framework learning process and for faster convergence, batch normalisation [23] activity is performed on the output feature vector of size 100×64 from the DWS

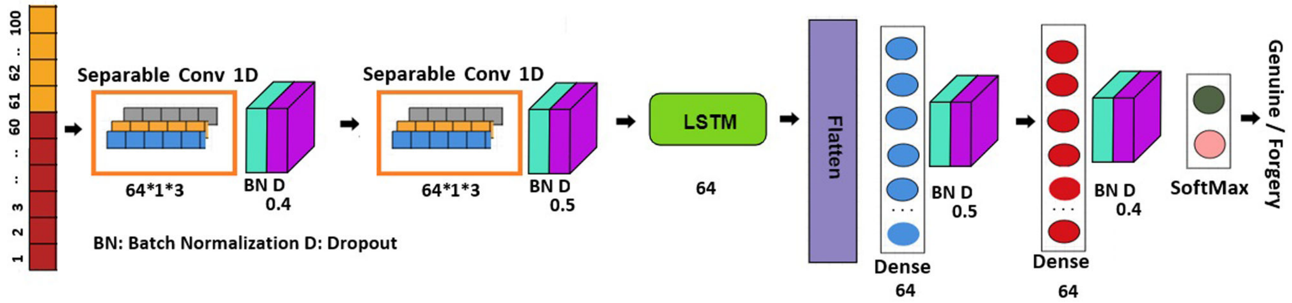


Fig. 4 *Diagrammatic representation of separable 1D convolution based proposed OSV framework architecture*

Table 3 Assessing the amount of operations and weights for standard and separable convolution operations

Convolution layer type	Standard	Separable
total operations	$W \times K \times C_{\text{in}} \times C_{\text{out}}$	$W \times K \times C_{\text{in}} + W \times C_{\text{in}} \times C_{\text{out}}$
total weights	$K \times C_{\text{in}} \times C_{\text{out}}$	$K \times C_{\text{in}} + C_{\text{in}} \times C_{\text{out}}$
reduction factor (2.03%)	$1/ C_{\text{in}} + 1/(N^2)$	$1/ C_{\text{in}} + /(N^2)$
	248,066	24,3121

C_{in}, C_{out} = Number of input and output channels, respectively, W = width of kernel 'K'. N = Number of kernels.

Table 4 Overview of the constituting building blocks of proposed framework

Layer	Size	Parameters
separable convolution 1D + Batch normalisation + DropOut	64 × 1 × 3	{kernel_initialiser, bias_initialiser, depthwise_initialiser} = 'random_uniform', pad = 'same', activation = 'relu', 0.4
separableConvolution + BatchNormalisation + DropOut	64 × 1 × 3	{kernel_initialiser, bias_initialiser, depthwise_initialiser} = 'random_uniform', pad = 'same', activation = 'relu', 0.5
LSTM layer	64 units	recurrent_dropout = 0.2, return_sequences = True
fully connected + BatchNormalisation + DropOut	64	{kernel_initialiser, bias_initialiser} = 'random_uniform', pad = 'same', activation = 'relu', 0.5
fully connected + BatchNormalisation + DropOut	64	{kernel_initialiser, bias_initialiser} = 'random_uniform', pad = 'same', activation = 'relu', 0.4

convolution layer. A 1×1 pointwise convolution activity is actualised on the output feature vector, which yields a feature vector of dimension 100×64. Same set of operations is executed on the second DWS layer, which consists of 64 filters, each of dimension 1×3 and yields a feature vector of measurement 100×64. The deep features from the DWS layers better capture the short-term dependencies of the signature stroke points, these deep features representing short term feature dependencies are passed as input to the LSTM layer. The LSTM layer consists of 64 units and yields a feature vector of dimension 100×64. The deep representation features are passed to the fully connected layers for classification.

As presented in Table 3, the depth wise separable 1D convolution requires 2.03% reduced parameters contrasted to a standard 1D convolution operation. The reduced parameter count leads to elevated representational efficiency and faster framework learning. As a best practice [33], a dropout of 40%, 50% is practised at both the DWS convolutional layers.

5.5 Fully connected network

The deep features of dimension 100×64 captured by the DWSCNN and LSTM layer are reshaped into 1×6400 by the flatten layer. The deep feature representations pass on as a contribution to the multilayer perceptron (MLP), which acts as a classifier to classify the test signature. The MLP includes an input layer, a hidden layer and SoftMax layer for final classification. The first fully connected layer consists of 64 neurons and yields a vector of dimension 1×64, which is moved as an input to the hidden layer, which contains 128 neurons. The hidden layer yields a feature vector of dimension 1×128. Batch normalisation and dropout are applied on the output from the hidden layer which brings about a high-level feature vector of measurement 1×128, which is given to the sigmoid layer of size 2, which finally classifies the signature as real or fake. A dropout of 50%, 40% is executed at each fully connected layer, respectively. To reproduce the results, we have shown all the parameters and their values in Table 4. We have set 'ReLU' as an activation function in convolution and hidden layers and sigmoid function in classification layer. The above final parameters of the framework were chosen dependent on the training and validation accuracies of K-fold cross validation.

6 Experimentation and results

6.1 Parameter settings

Attaining one-shot learning is a challenging requirement for OSV framework. Accordingly, to select the best combination of parameters for the proposed OSV framework, we apply 10-fold cross-validation to appraise the proposed OSV framework. To set the parameters, we have assessed the proposed framework on the Skilled_01 category of MCYT-100 dataset, in which the dataset is split into a training set and test set. The train set consists of samples of 90% of users and test set with 10% of users. We sample the training set 10 times by random sampling with replacement. The intuition behind selecting MCYT-100 dataset is that it consists of a greater number of signature samples per user compared to SVC and SUSIG datasets. The parameters to set in our framework are: the number of separable convolution layers $(SL) = \{2, 3, 4\}$, number of kernels in each separable convolution layer $(K) = \{16, 32, 64\}$, kernel size $(KS) = \{3, 5\}$, number of dense layers $(DL) = \{2, 3\}$, number of nodes in each dense layer $(DN) = \{16, 32, 64, 128\}$, activation function $(A) = \{\text{tanh, Relu}\}$, optimiser $(O) = \{\text{Adam, SGD}\}$, the initialisation values for kernel and bias = {random_uniform, Glorot_uniform}.

Recent research by the authors of [19, 34] summarises that, A = 'Relu', optimiser = 'Adam', kernel initialiser = 'random uniform' is the optimal values. A K-fold cross-validation ($K = 10$) is performed

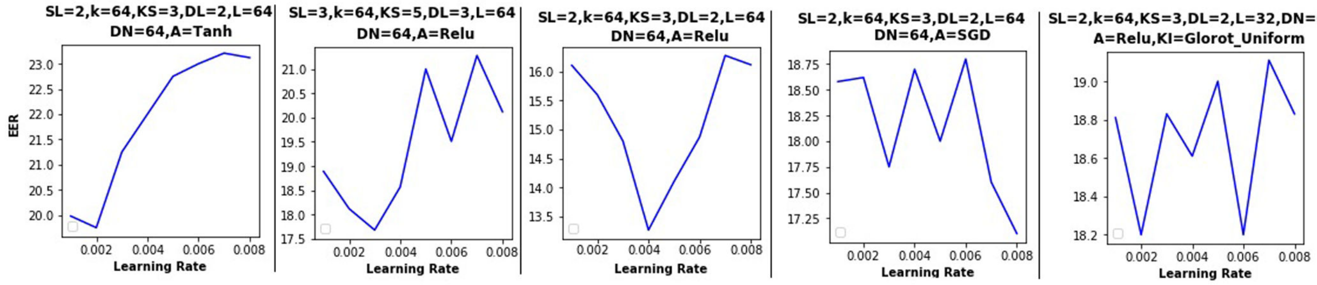


Fig. 5 *The EER outcome of the proposed framework by fixing the number of separable convolution layers and varying other parameters of the framework*

and the classification accuracy outcome of the proposed framework in ten cases is computed and the average is noted. The same technique is followed for all possible combination of parameters. The combination of parameters resulted in the least EER is considered as final. The outcome of parameter analysis is depicted in Fig. 5.

In our case, as depicted in Fig. 5*c*), the combination of parameters {$SL = 2$, $K = 64$, $KS = 3$, $DL = 2$, $DN = 64$, LSTM nodes $L = 64$, $LR = 0.004$, optimiser = Adam} resulted in least EER value. Hence, we have performed a set of experiments using this set of parameters.

6.2 Experimental analysis

On selecting the optimal set of parameters, to appraise the proposed OSV framework, we have performed a wide set of experiments which covers every single possible category of experimentation using three broadly utilised publicly accessible datasets, i.e. MCYT-100 [1, 2, 4, 5, 35], SVC – Task 2 [24, 36, 37], SUSIG [3, 26, 27]. Recently Ruben *et al.* [20] developed a novel largest online signature dataset named 'DeepSignDB', which is a collection of signature samples from widely used datasets like MCYT-330, BiosecurID, Biosecure DS-2, e-BioSignDS1 and e-BioSignDS2. 'DeepSignDB' consists of 70K signatures of 1526 users. Having the largest number of signature samples, 'DeepSignDB' address the challenge of data scarcity in developing novel deep learning frameworks for OSV. Even though it is the largest online signature database, very few OSV frameworks are experimented with 'DeepSignDB'. Hence, we have chosen the most widely used publicly available datasets, i.e. MCYT-100, SVC and SUSIG. We have executed our experiments on Nvidia, GForce RTX 2080 4×12 390 GB GPU. We have adopted standard procedures as described in [16, 23, 38] to split the train and test signature samples for various categories and datasets.

A concise depiction of each dataset, related to a total count of users, the total number of forgeries and genuine signature samples available for each user, number of global features per signature etc. are listed in Table 1. Similar to the latest LSTM-based OSV frameworks [12, 13, 17–20, 39–41], to evaluate our framework thoroughly, we have considered five categories of experimentation, i.e. S_01, S_05, S_10, S_15 and S_20 in skilled forgery category and R_01, R_05, R_10, R_15 and R_20 in the random forgery category. If the dataset consists of a total '*T*' number of users, each user with '*G*' genuine and '*F*' number of forgery signature samples, in case of skilled forgery, i.e. S_*N*, where $N = 01, 05, 10, \ldots, (G{-}5)$, the framework is trained with '*N*' genuine, '*N*' forgery signature samples and tested with '$(G{-}N)$' genuine, '$(F{-}N)$' forgery samples. In case of random forgery, i.e. R_*N*, where $N = 01, 05, 10, \ldots, (G{-}5)$, the framework is trained with '*N*' genuine and a set of '$T{-}1$' randomly selected one genuine signature sample per user as forgery samples and tested with '$(G{-}N)$' genuine samples and '$T{-}1$' randomly selected one signature sample per user excluding the samples selected for training. It infers that the number of training and testing samples are more in case of random categories compared to skilled category. The frameworks yielded maximum EER values are represented with (*) and the frameworks with the next top EER are marked with (**). The proposed framework accomplished state-of-the-art outcomes in S_01, S_10, S_15, R_10, R_15 and R_20 categories in context of MCYT-100. The framework accomplished the superlative EER in S_01, S_05, S_10, S_15, R_10 and R_15 categories with SVC dataset. Subject to SUSIG, the framework attained the best EER in S_10, R_01 and R_10 categories.

Based on Tables 5–7, we could showcase that in all datasets and for all categories of experimentation, the framework accomplishes reducing the EER with the increase of training samples. In one case, i.e. R_15 subject to SVC, the values are bit higher compared to the categories trained with lesser number of samples. This can be possibly attributed to the early set of weights fixed to the framework through 'random_uniform' as a weight initialiser. Also, Tables 5–7 conclude that, in the case of skilled and random categories, SUSIG and MCYT-100, respectively, illustrate the steep decrease in EER value trained with lesser number of samples. Also, Tables 5–7 show that our proposed framework accomplishes one-shot learning, i.e. when the model is trained with 1 genuine and 1 forgery signature sample per each user, the model achieves the cutting-edge result by beating the state-of-the-art works. Realising one-shot learning by the proposed OSV framework qualifies it to deploy in real time scenarios in which gaining more signature samples is impossible, for example, such as e-commerce and m-commerce applications and so forth. Similarly, the tables confirm that only a few works [2, 23] evaluated their proposed frameworks with S_10, S_15, R_10, R_15 categories in case of MCYT, R_10, R_15 categories of SVC and R_05 and R_10 categories of SUSIG datasets.

The EER output of the proposed framework subject to the SUSIG dataset in a skilled category is relatively high with respect to corresponding EER obtained from the MCYT and SVC datasets. The fundamental explanation for this low EER performance (in the case of skilled categories) is that the majority of the forged signature samples acquired from the forgers in the SUSIG dataset are significantly similar to those of genuine signature samples. Consequently, the framework does not learn feature representations efficiently to differentiate the genuine and forgery signature samples. Subsequently, it brings lower performance. However, in case of random categories R-01 and R-10 the proposed framework resulted in a state of the art EER. Fig. 6 portrays the EER of each user in Skilled 1 and Random 10 categories of SUSIG and SVC dataset, respectively, in the form of a 2D-Histogram.

Fig. 6*a* depicts that the users from 45 to 50 contribute to higher EER of the framework compared to others and the average EER varies between 15 and 35%. Correspondingly, Fig. 6*b* delineates that users from 25 to 35 contributes to higher EER of the framework and the average EER varies between 15 and 20%. We have implemented our framework using both the normal convolutions and depth-wise separable convolutions. Fig. 7 outlines that depth-wise separable convolution results in decreased EER and converges faster as compared to normal convolutions. Fig. 7 illustrates that depth-wise separable based framework started reaching zero EER with 5 signature samples in both skilled and random categories, whereas in case of normal convolutions, the framework requires a minimum 20 signature sample to reach zero EER.

In the case of random category, with 10 signature samples, the depth-wise separable convolutions-based framework reaches zero EER. Based on Fig. 7, Table 3, we can confirm that depth-wise separable convolutions result in decreased parameters and improved learning even with one training signature sample. Fig. 8, outlines the EER patterns of the proposed framework with three datasets with varying input signature samples.

Table 5 EER evaluation of the proposed OSV framework with recent frameworks on MCYT-100 dataset (‘-’indicates the experimentation is not done in that category)

Technique	S_01	S_05	S_10	S_15	S_20	R_01	R_05	R_10	R_15	R_20
Proposed (Clustering + feature fusion + few shot learning)#	**13.26***	2.66	**2.58***	**3.01***	1.2	6.01	2.91	**0.07***	**0.05***	**0.04****
GMM+DTW [1]	-	3.05	-	-	-	-	-	-	-	-
Hausdorff distance [2]	-	6.05	4.23	**3.10****	-	-	2.95	1.81	**1.20****	-
information divergence [35]	-	3.16	-	-	-	-	-	-	-	-
cancelable templates [4]	-	10.29	-	-	-	-	-	-	-	-
common threshold and feature – interval valued[5]	-	10.36	-	-	5.82	-	10.32	-	-	0.74
curvature [2]	-	10.22	8.25	6.38	-	-	4.12	3.33	2.58	-
cancelable templates[4]	-	13.30	-	-	-	-	-	-	-	-
writer specific parameters (conventional) [5]	-	6.79	-	-	**0.00***	-	1.73	-	-	**0.00***
histogram [23]	-	4.02	**2.72****	-	-	-	1.15	**0.44****	-	-
writer specific parameters (symbolic) [14]	-	2.2	-	-	0.6	-	1.0	-	-	0.1
common threshold and feature (conventional) [5]	-	13.12	-	-	11.23	-	5.61	-	-	1.66
writer specific classifiers and features [42]	-	19.4	-	-	1.1	-	7.8	-	-	0.8
KNN-regional features [10]	-	4.65	-	-	-	-	1.33	-	-	-
SW [25]	13.72	-	-	-	-	**5.04****	-	-	-	-
PDTW(case 2) [9]	-	-	-	-	-	-	**0.018****	-	-	-
DTW(F13) [10]	-	8.36	-	-	-	-	6.25	-	-	-
VQ+DTW[11]	-	**1.55***	-	-	-	-	-	-	-	-
TW [25]	**13.56****	-	-	-	-	**4.04***	-	-	-	-
DTW+ RL(SF) [12]**#**		**1.62 ****					0.23			
histograms [23]	-	4.02	-	-	2.72	-	1.15	-	-	0.35
DTW + RL(RF) [12]**#**		1.81					0.24			
secure KNN [10]	-	5.15	-	-	-	-	1.70	-	-	-
SM-DTW (F13) [10]	-	13.56	-	-	-	-	4.31	-	-	-
PDTW(case 1) [9]	-	-	-	-		-	**0.011***	-	-	-
writer specific parameters (interval valued representation) [5]	-	2.51	-	-	**0.03****	-	0.70	-	-	**0.00***
torsion feature [2]	-	9.22	7.04	5.12	-	-	3.42	2.25	1.90	-
WP+BL DTW[36]	-	2.76	-	-	-	-	-	-	-	-
VSA DTW[33]	-	3.24	-	-	-	-	0.80	-	-	-
VSA_ r DTW[33]	-	2.68	-	-	-	-	0.75	-	-	-

#: The skilled forgeries were added to the training set.

The bold values represent the first best and second best EER value in each category.

Table 6 EER evaluation of the proposed OSV framework with recent frameworks on SVC dataset (‘-’indicates the experimentation is not done in that category)

Technique	S_01	S_05	S_10	S_15	R_01	R_05	R_10	R_15
Proposed Model – (clustering + feature fusion + few shot learning)#	**7.71***	**3.43***	**2.75****	**0.45****	3.09	0.41	**0.12****	**0.22***
LCSS[37]	-	-	5.33	-	-	-	-	-
Relief-1 [24]**#**	-	-	8.1	-	-	-	-	-
Hausdrorff distance [2]	-	**9.83****	6.61	**3.10***	-	3.54	1.24	**1.81****
DTW (common threshold) [36]	-	-	7.80	-	-	-	-	-
Relief-2 [24] **#**	-	-	5.31	-	-	-	-	-
PDTW [9]	-		-	-	-	**0.0025***	-	-
LNPS + RNN[17]	-	-	-	-	-	2.37	-	-
TW [25]	18.63	-	-	-	**0.50***	-	-	-
DCT [3]#	-	-	3.98	-	-	-	**0.10***	-
PDTW(case 2) [9]	-	-	-	-	-	**0.0175****	-	-
stroke-wise [25]	**18.25****	-	-	-	**1.90****	-	-	-
SPW[15]	-	-	**1.00***	-		-	-	-
variance selection [24] **#**	-	-	13.75	-	-	-	-	-
SVM +mRMR (10-Samples) [15]	-	-	**1.00***	-	-	-	-	-

The bold values represent the first best and second best EER value in each category.

In the event of the skilled category, SVC dataset brings about lesser EER and shows a diminishing pattern as training samples increases. Comparable is the situation with MCYT-100 followed by SUSIG. In the event of Random category, MCYT-100 outcomes in somewhat higher EER contrasted with the other two, yet the EER results from each of the three datasets convergence to zero with increasing training samples.

As exhibited in Tables 5–7, regardless of the systems proposed in [2, 5, 9, 11, 12, 25, 35, 43] is yielding better EER values contrasted with the proposed framework, the essential pitfall with these models is that these models are assessed with only one category, i.e. S_01, R_01. Whereas, the proposed model is thoroughly evaluated with wide experimentation in Skilled 1, 5, 10, 15, 20 and Random 1, 5, 10, 15, 20. Hence, its superiority is validated as compared to [2, 5, 9, 11, 12, 25, 35, 43].

Table 7 EER evaluation of the proposed OSV framework with recent frameworks on SUSIG dataset ('-'indicates the experimentation is not done in that category)

Technique	S_01	S_05	S_10	R_01	R_05	R_10
Proposed model – (clustering + feature fusion + few shot learning)#	15.84	4.95	**1.68****	**1.70****	1.37	**0.21***
Hausdorff distance [2]	-	7.05	-	-	**1.02***	-
Rapid human movements [26]	7.87	-	-	3.61	-	-
DCT [3]**#**	-	-	**0.51***	-	-	-
with all domain [27]	-	-	3.88	-	-	-
SW [25]	**7.74****	-	-	2.23	-	-
writer dependent features and classifiers [42]	-	-	1.92	-	-	-
pole-zero [43]**#**	-	3.91	-	-	1.97	-
TW [25]	**6.67***	-	-	**1.55***	-	
with stable domain [27]	-	-	2.13	-	-	-
information divergence [35]	-	**1.6****	2.13	-	-	-
normalisation [39]	-	-	3.52	-	-	-
Cosα + enhanced DTW [44]	-	-	3.06	-	-	-
VSA DTW[33]	-	3.48	-	-	-	-
VSA_ r DTW[33]	-	3.09	-	-	-	-
Parzen Window + DCT[45] **#**	-	**1.49***			**1.23****	

#: The skilled forgeries were added to the training set.

The bold values represent the first best and second best EER value in each category.

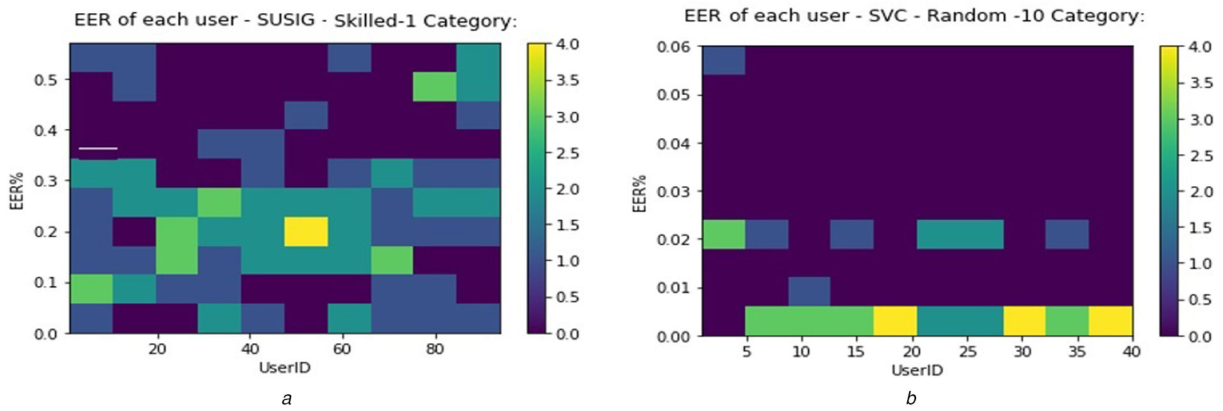


Fig. 6 *2D Histogram of EER of each user*
(a) The EERs of 94 users of SUSIG dataset obtained for Skilled_1, ***(b)*** The EERs of 40 users of SVC dataset obtained for Random_10 categories

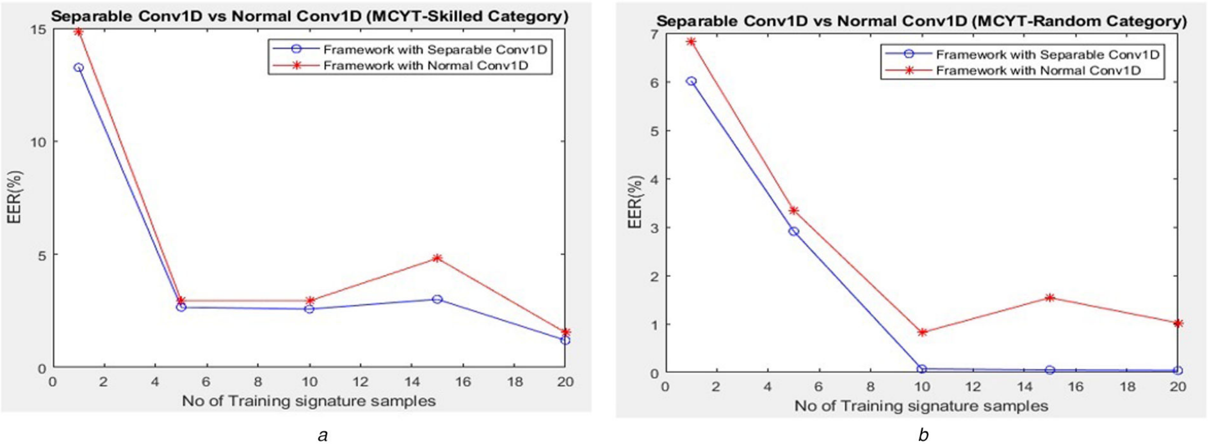
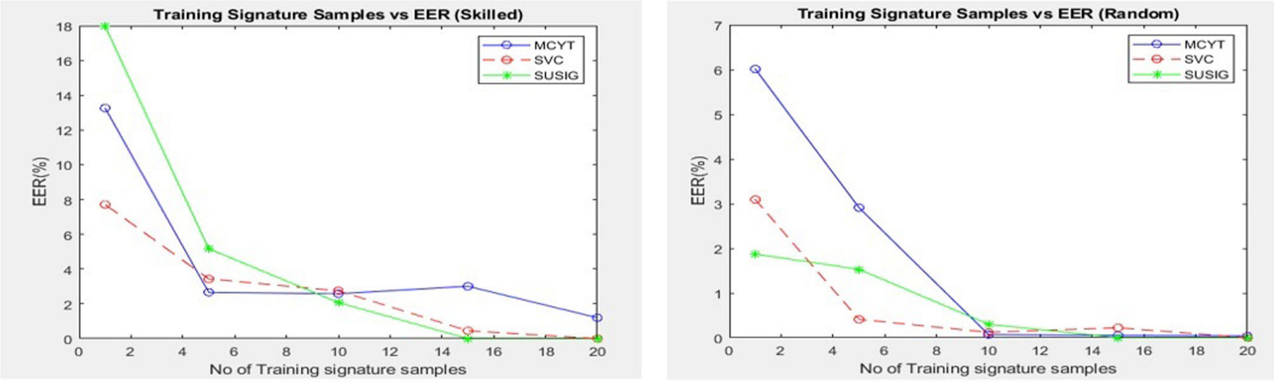


Fig. 7 *EER outcome of the proposed framework in scenarios of using separable and standard convolutions*
(a) Skilled, ***(b)*** Random forgeries

7 Conclusion and future work

In this work, the performance of the proposed writer dependent DWSCNN and LSTM-based model for OSV is demonstrated, whereby an extensive set of experiments was conducted on three widely used datasets. A hybrid feature set is formed from an ensemble of writer dependent feature cluster heads and deep representations returned by an auto-encoder. A novel DWSCNN and an LSTM architecture for signature classification is proposed, in which the hybrid feature set is given as input to the DWSCNN layer, where depth-wise separable convolution operations automatically learn hidden representation, which are given as an input to the LSTM layers to categorise the input online signature. The foremost advantage of our DeepFuseOSV is that it realises few-shot learning by achieving state of the art outcomes in S_01, S_10, S_15, R_10, R_15 and R_20 categories of MCYT-100,



a

b

Fig. 8 *EER outcome by the proposed framework with three datasets with changing training signature samples –*
(a) Skilled, ***(b)*** Random categories

S_01, S_05, S_10, S_15, R_10 and R_15 categories of SVC, R_01, R_05 and R_10 categories of SUSIG. The ability to result in lower EER values, even with a single training sample, qualifies the proposed framework for real-time deployment.

8 References

[1] Galbally, J., Fiérrez, J., Diaz, M., *et al.*: 'Improving the enrollment in dynamic signature verfication with synthetic samples'. Int. Conf. on Document Analysis Recognition (ICDAR), Barcelona, Spain, 2009, pp. 1295–1299

[2] He, L., Tan, H., Huang, Z.: 'Online handwritten signature verification based on association of curvature and torsion feature with Hausdorff distance', *Multimedia Tools Appl.*, 2019, **78**, pp. 19253–19278

[3] Liu, Y., Yang, Z., Yang, L.: 'Online signature verification based on DCT and sparse representation', *IEEE Trans. Cybern.*, 2015, **45**, (11), pp. 2498–2511

[4] Maiorana, E., Campisi, P., Fierrez, J., *et al.*: 'Cancelable templates for sequence-based biometrics with application to on-line signature recognition', *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, 2010, **40**, (3), pp. 525–538

[5] Vorugunti, C.S., Guru, D.S., Viswanath, P.: 'An efficient online signature verification based on feature fusion and interval valued representation of writer dependent features'. IEEE fifth Int. Conf. on Identity, Security and Behavior Analysis (ISBA), Hyderabad, India, 2019

[6] Diaz, M., Ferrer, M.A., Impedovo, D., *et al.*: 'A perspective analysis of handwritten signature technology', *ACM Comput. Surv. (CSUR)*, 2019, **51**, (6), pp. 1–39

[7] Guru, D.S., Manjunatha, K.S., Manjunath, S., *et al.*: 'Interval valued symbolic representation of writer dependent features for online signature verification', *Expert Syst. Appl.*, 2017, **80**, pp. 232–243

[8] Guru, D.S., Prakash, H.N.: 'Online signature verification and recognition: an Approach based on symbolic representation', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009, **31**, pp. 1059–1073doi:

[9] Al-Hmouz, R., Pedrycz, W., Daqrouq, K., *et al.*: ' quantifying dynamic time warping distance using probabilistic model in verification of dynamic signatures', *Soft Comput.*, 2019, **23**, pp. 407–418

[10] Doroz, R., Kudlacik, P., Porwika, P.: 'Online signature verification modeled by stability oriented reference signatures', *Inf. Sci.*, 2018, **460-461**, pp. 151–171

[11] Sharma, A., Sundaram, S.: 'An enhanced contextual dtw based system for online signature verification using vector quantization', *Pattern Recognit. Lett.*, 2016, **84**, pp. 22–28

[12] Lai, S., Jin, L.: 'Recurrent adaptation networks for online signature verification', *IEEE Trans Inf. Forensics Secur.*, 2018, **14**, pp. 1624–1637

[13] Ruben, T., Ruben, V.-R., Julian, F., *et al.*: 'Exploring recurrent neural networks for on-line handwritten signature biometrics', *IEEE Access*, 2018, **6**, pp. 5128–5138

[14] Ansari, A.Q., Hanmandlu, M., Kour, J., *et al.*: 'Online signature verification using segment-level fuzzy modelling', *IET Biometrics*, 2014, **3**, pp. 113–127

[15] Kar, B., Mukherjee, A., Dutta, P.K.: 'Stroke point warping-based reference selection and verification of online signature', *IEEE Trans. Instrum. Meas.*, 2018, **67**, pp. 2–11doi:

[16] Van den Oord, A., Kalchbrenner, N., Espeholt, L., *et al.*: 'Conditional image generation with pixel cnn decoders'. Advances in Neural Information Processing Systems (NIPS), Barcelona, Spain, 2016, pp. 4790–4798

[17] Lai, S., Jin, L., Yang, W.: 'Online signature verification using recurrent neural network and length-normalized path signature descriptor'. 14th IAPR Int. Conf. on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 2017

[18] Marta, G.B., Javier, G., Julian, F., *et al.*: 'Enhanced on-line signature verification based on skilled forgery detection using Sigma-LogNormal features'. Int. Conf. on Biometrics (ICB, 2015), Phuket, Thailand, 2015, pp. 501–506

[19] Kian, A., Bagher, B.: 'Usage of autoencoders and siamese networks for online handwritten signature verification', *Neural Comput. Appl.*, 2019, **31**, pp. 9321–9334

[20] Ruben, T., Ruben, V-R., Julian, F., *et al.*: 'Do you need more data? The DeepSignDB on-line handwritten signature biometric database'. 15th Int. Conf. on Document Analysis and Recognition, Sydney Australia, 2019, pp. 1143–1148

[21] Ruben, V.R., Ruben, T., Miguel, C., *et al.*: 'DeepSignCX: signature complexity detection using recurrent neural networks'. 15th Int. Conf. on Document Analysis and Recognition, Sydney, Australia, 2019, pp. 1482–1487

[22] Chandra, S.V., Prerana, M., Viswanath, P., *et al.*: 'OSVNet: convolutional siamese network for writer independent online signature verification'. 15th Int. Conf. on Document Analysis and Recognition, Sydney Australia, 2019, pp. 1470–1475

[23] Sae-Bae, N., Memon, N.: 'Online signature verification on mobile devices', *IEEE Trans. Inf. Forensics Secur.*, 2014, **9**, (6), pp. 933–947

[24] Yang, L., Cheng, Y., Wang, X., *et al.*: 'Online handwritten signature verification using feature weighting algorithm relief', *Soft Comput.*, 2018, **22**, pp. 7811–7823doi:

[25] Diaz, M., Fischer, A., Ferrer, M.A., *et al.*: 'Dynamic signature verification system based on one real signature', *IEEE Trans Cybern.*, 2018, **48**, pp. 228–239doi:

[26] Diaz, M., Fischer, A., Plamondon, R., *et al.*: 'Towards an automatic on-line signature verifier using only one reference per signer'. Int. Conf. Document Analysis Recognition (ICDAR), Tunis, Tunisia, 2015, pp. 631–635

[27] Pirlo, G., Cuccovillo, V., Diaz-Cabrera, M., *et al.*: 'Multidomain verification of dynamic signatures using local stability analysis', *IEEE Trans Hum.-Mach. Syst.*, 2015, **45**, pp. 805–810

[28] Rousseeuw, J.P., Croux, C.: 'Alternatives to the median absolute deviation', *J. Am. Stat. Assoc.*, 1993, **88**, pp. 1273–1283

[29] Chandra, S.V, Rama krishna, G., Viswanath, P, *et al.*: 'Online signature verification by few-shot separable convolution based deep learning'. 15th Int. Conf. on Document Analysis and Recognition, Sydney, Australia, 2019

[30] Daksha, Y., Naman, K., Akshay, A., *et al.*: 'Fusion of handcrafted and deep learning features for large-scale multiple iris presentation attack detection'. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)., Salt Lake City, UT, USA, 2018, pp. 685–692

[31] Muhammad, A.K., Muhammad, S., Tallha, A., *et al.*: 'Hand-crafted and deep convolutional neural network features fusion and selection strategy: an application to intelligent human action recognition', *Appl. Soft Comput. J.*, 2020, **87**, p. 105986

[32] Aguilar, J.F.: 'Adopted Fusion Schemes for Multimodal Biometric Authentication', PhD thesis, Biometric Research Lab-AVTS, 2006

[33] Diaz, M., Miguel Ferrer, A., Jose Quintana, J.: 'Anthropomorphic features for On-line signatures', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019, **41**, pp. 2807–2819

[34] Aneja, J., Deshpande, A., Schwing, A.G.: 'Convolutional image captioning'. The IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 5561–5570

[35] Tang, L., Kang, W., Fang, Y.: 'Information divergence-based matching strategy for online signature verification', *IEEE Trans. Inf. Forensics Secur.*, 2018, **13**, pp. 861–873

[36] Sharma, A., Sundaram, S.: 'On the exploration of information from the DTW cost matrix for online signature verification', *IEEE Trans. Cybern.*, 2018, **48**, pp. 611–624doi:

[37] Barkoula, K., Economou, G., Fotopoulos, S.: 'Online signature verification based on signatures turning angle representation using longest common subsequence matching', *Int. J. Doc Anal. Recog.*, 2013, **16**, pp. 261–272

[38] Chollet, F.: 'Xception: deep learning with depthwise separable convolutions'. The IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), USA, 2017, pp. 1251–1258

[39] Tolosana, R.V., Rodriguez, R, Fierrez, J., *et al.*: 'Biometric signature verification using recurrent neural networks'. 14th Int. Conf. on Document Analysis and Recognition (ICDAR), Kyoto Japan, 2017

[40] Chuang, L., Xing, Z., Feng, L., *et al.*: 'Stroke-based RNN for writer-independent online signature verification'. Int. Conf. on Document Analysis and Recognition (ICDAR), Australia, 2019, pp. 526–532

[41] Ruben, T., Ruben, V.-R., Julian, F., *et al.*: 'Deepsign: deep on-line signature verification', arxiv. Feb 2020, 2002o.10119

[42] Manjunatha, K.S., Manjunath, S., Guru, D.S., *et al.*: 'Online signature verification based on writer dependent features and classifiers', *Pattern Recognit. Lett.*, 2016, **80**, pp. 129–136

[43] Rashidi, S., Fallah, A., Towhidkhah, F.: 'Authentication based on pole-zero models of signature velocity', *J. Med. Signals Sens.*, 2013, **3**, pp. 195–208

[44] Khalil, M.I., Moustafa, M., Abbas, H.M.: 'Enhanced DTW based on-line signature verification'. Proc. of the 16th IEEE Int.Conf. on Image Processing (ICIP), Cairo, Egypt, 2009

[45] Rashidi, S., Fallah, A., Towhidkhah, F.: 'Feature extraction based DCT on dynamic signature verification', *Scientia Iranica*, 2012, **19**, (6), pp. 1810–1819