2020 International Conference on Machine Vision and Image Processing (MVIP)
Faculty of Engineering, College of Farabi, University of Tehran, Iran, 19 & 20 Feb. 2020

Offline Handwritten Signature Verification And Recognition Based On Deep Transfer Learning Using Convolutional Neural Networks (A Literature Review)

Atefeh Foroozandeh
Department of Applied Mathematics, Faculty of Sciences and Modern Technology, Graduate University of Advanced Technology
Kerman, Iran
at_foroozandeh@yahoo.com

Ataollah Askari Hemmat
Department of Applied Mathematics, Faculty of Mathematics and Computer, Shahid Bahonar University of Kerman,
Kerman, Iran
askari@uk.ac.ir

Hossein Rabbani
Department of Biomedical Engineering, School of Advanced Technologies in Medicine, Isfahan University of Medical Sciences,
Isfahan, Iran
h_rabbani@med.mui.ac.ir

***Abstract*—Recently, deep convolutional neural networks have been successfully applied in different fields of computer vision and pattern recognition. Offline handwritten signature is one of the most important biometrics applied in banking systems, administrative and financial applications, which is a challenging task and still hard. The aim of this study is to review of the presented signature verification/recognition methods based on the convolutional neural networks and also evaluate the performance of some prominent available deep convolutional neural networks in offline handwritten signature verification/recognition as feature extractor using transfer learning. This is done using four pre-trained models as the most used general models in computer vision tasks including VGG16, VGG19, ResNet50, and InceptionV3 and also two pre-trained models especially presented for signature processing tasks including SigNet and SigNet-F. Experiments have been conducted using two benchmark signature datasets: GPDS Synthetic signature dataset and MCYT-75 as Latin signature datasets, and two Persian datasets: UTSig and FUM-PHSD. Obtained experimental results, in comparison with literature, verify the effectiveness of the models: VGG16 and SigNet for signature verification and the superiority of VGG16 in signature recognition task.**

***Index Terms*—Offline Handwritten Signature Verification, Signature Recognition, Convolutional Neural Network, Deep Transfer Learning.**

I. Introduction

Every day around the world, security systems use biometrics in order to recognize or verify the authenticity. Handwritten signature is one of the most important biometrics applied in banking systems, administrative and financial applications. A signature processing system receives a signature in two ways: offline and online. Due to lack of information, offline handwritten signature processing is more difficult task than online cases and considered as a hard problem [1]. A Handwritten Signature Verification (HSV) system deals with three kinds of forgeries: random, simple, and skilled forgeries [2]. Two types of learning may be used in a Handwritten Signature Verification/Recognition (HSV/R) systems: Writer-Independent (WI, general learning) or Writer-Dependent (WD, special learning) [3]. In the case of WI, learning is done based on a large population of signature samples related to all persons in the dataset, whereas in the case of WD, learning is conducted based on the signature samples from each person, separately [3]. Although WD learning achieve good results, a classifier must be conducted for each user added to the system and therefore the complexity and the cost of the system increases [1].

Every HSV/R system includes three main steps, including preprocessing, feature extraction, and classification. Feature extraction techniques can be classified as handcrafted feature extractors or learning feature representation. Handcrafted features are widely used for signature verification [1]. With the advancement in artificial intelligence and appearance of Convolutional Neural Network (CNN) models and deep learning, a revolution has been occurred in the results of the automatic processing systems. Learning feature representation tries to learn a good feature representation directly from the raw data which is successfully carried out using deep CNNs [4]–[6] which is reviewed in Secs.II and III. One of the most successful techniques in the field of CNN models is transfer learning or knowledge transfer. Transfer learning enables us to use the potential of a CNN model from the source task into another task which suffer the limitation of huge data or lack of the time.

To the best of our knowledge, there is no study to give a benchmark for authors in the field of HSV/R methods based on CNNs, till today. In this paper, the presented signature verification/recognition methods based on CNNs have been reviewed. Also, the performance of some prominent available deep CNNs including SigNet, SigNet-F (already trained on signature datasets) and VGG16, VGG19, InceptionV3, and ResNet50 (trained on ImageNet) has been evaluated for offline

978-1-7281-6832-6/20/$31.00 ©2020 IEEE

HSV/R task as feature extractor using transfer learning.
The rest of this paper is organized as follows: in Sec.II, a brief review on CNN models and deep transfer learning is presented. The literature is reviewed in Sec.III. Sec.IV describes the methodology for HSV based on transfer learning using some current CNN models and presents the conducted experiments. The procedure of HSR using deep transfer learning along with the corresponding experiments are explained in Sec.V. Finally, conclusion and future works are presented in Sec.VI.

II. Convolutional Neural Network and Deep Transfer Learning

Traditional machine learning models have been trained based on the handcrafted features and the accuracy of the classification is directly related to these features. This dependency is considered as the major weakness of traditional models [7]. In recent years, Deep Learning (DL) and CNNs attract the attention of researchers especially in bioinformatics. There has been a significant improvement in the reported results of the signature verification systems based on DL and CNN compared to the handcrafted features.
Traditional neural networks include an input layer, a hidden layer and output layer as seen in Fig.1(a). Learning with more than one hidden layer is known as deep learning, see Fig.1(b). CNNs as a class of deep learning models, were introduced in 1980 which include several hidden layers. As shown in this figure, the hidden layers typically consist of convolutional layer, Rectified Linear Unit (ReLU) layer, pooling layer, and Fully Connected (FC) layer [5].

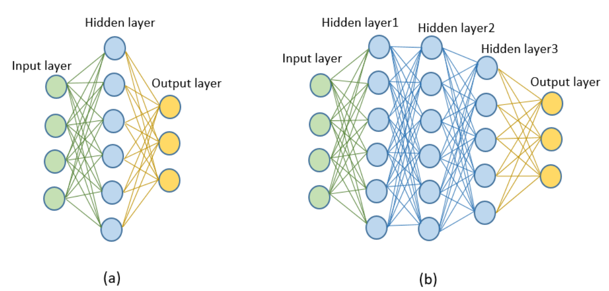

Fig. 1: The structure of a simple traditional neural network (a) and a deep neural network (b).

Convolutional layers try to represent high-level features of data. These layers compute the linear convolution. The nonlinear structure of a CNN model consists of a nonlinear activation layer after each convolutional layer such as tanh, sigmoid and Rectified Linear Unit (ReLU). A CNN model has a large set of parameters to be trained. Therefore, a large amount of data is needed for training a CNN model which is hard in the most of applications especially in biomedical researches [10]. So, the necessity of a large dataset is considered as the most important deficiency in the applications of CNN models. In order to overcome this deficiency, two concepts of transfer learning and fine tuning have been presented.
A good survey on transfer learning has been gathered by Pan et al. [9]. In this survey, a formal definition of transfer learning has been presented, as follows; Given a domain $D = \{\chi, P(X)\}$, which χ is a feature space, $P(X)$ is marginal probability of a sample point $X = \{x_1, ..., x_n\}, x_i \in \chi$, and x_i is a specific vector. A task $\tau = \{\gamma, P(Y|X)\}$ consists of a label space γ, a conditional probability distribution $P(Y|X)$ learned using training data of $x_i \in X$ and $y_i \in \gamma$. Suppose that D_S and τ_S indicate source domain and source task. Also, D_T and τ_T indicate target domain and target task. Transfer learning enables us to learn target conditional probability distribution $f_T(.) = P(Y_T|X_T)$ in D_T based on the acknowledge gained from D_S and τ_S where $D_S \neq D_T$ or $\tau_S \neq \tau_T$ [9].
It should be noted that transfer learning is divided into some categories: inductive, transductive, and unsupervised transfer learning [10]. The focus of this paper is on the inductive transfer learning where there are labeled datasets for both source and target task. Since training process of a CNN model is time consuming, some of the pioneer CNN models have been trained on the large datasets for tuning their weights (parameters) and their weights be available to the research community for application in various fields of research. These models already trained on some large datasets are known as pre-trained models. Using transfer learning, researchers can use the architecture and the weights of the pre-trained models from one task into other tasks without need to train the models from scratch and also save the time. This enables us to utilize the power of the CNN model in the situation like bioinformatics specially in biomedical fields which creating large datasets is not possible or a hard task [8].
There is a question of which pre-trained model is good for transfer learning. Based on the conducted research by Yosinski et al. [2], LeNet, AlexNet, VGG, Inception, and ResNet are good chooses in network-based deep transfer learning. Among these, four models: VGG16, VGG19 [11], InceptionV3 [12], and ResNet50 [13] have been used in this paper. These models became winner at ILSVRC (ImageNet Large Scale Visual Recognition Challenge) for several years [14]. ILSVRC challenge evaluates the presented algorithms for object detection and image classification at large scale (on a large dataset ImageNet) and fairly compares the results in the same situation. ImageNet includes 14,197,122 images of more than 20,000 categories of fruits, animals, etc. [15]. ImageNet is a large dataset and a good candidate for tuning the large set of parameters of deep CNN models. So, several CNN models have been trained on ImageNet and the corresponding weights are available to the research community to use with the help of transfer learning.
VGGNet was one of the famous model submitted to ILSVRC-2014 as 1st runner up for image classification task and won the localization task in ILSVRC 2014. VGG16 is a CNN model proposed by K. Simonyan and A. Zisserman [11]. VGG16 composed of 16 convolutional layers (3×3 convolutions) with a very uniform architecture (using lots of filters) leads to appealing results in many applications. VGG19 has the similar structure with VGG16 but composed of 19 convolutional layers and is a deeper model than VGG16.
The Inception model was introduced by GoogleNet in different

versions: InceptionV1, InceptionV2, and InceptionV3 [12]. InceptionV3 with fewer parameters (42 layers) and similar complexity as VGGNet could become the 1st Runner Up for image classification in ILSVRC-2015. Residual Neural Network (ResNet) introduced by Kaiming He et. al. [13], can have a very deep network including up to 152 layers which includs different versions: ResNet50, ResNet101, and ResNet152. ResNet uses the skip connection in order to fit the input from the previous layer to the next layer and enables ResNet became the winner of ILSVRC 2015 in image classification, detection, and localization.

Using transfer learning, the structures and the weights of the pre-trained models in the source task can be used in other tasks. One of the most important applications of transfer learning is using deep CNN as a feature extractor i.e. freeze the first layers of deep CNN from the input layer to the last layer. Forward-propagation of the freezed layers leads to project the input images onto the feature space and gives the feature vectors i.e. the output of the last layer.

It should be noted that the implementation of CNN models need to high computational complexity and also need to powerful platforms. To do this, several toolboxes have been developed including Torch, Theano, Keras, Caffe, CuDNN, TensorFlow, MXNet and deeplearning4j. TensorFlow has been developed by Google Brain team within Googles AI (Artificial Intelligence) organization and is a strong support for machine learning. Numerical computation can be performed using TensorFlow as an open source software library. In this paper, Keras as a simple and high-level model definition interface has been configured to use the TensorFlow backend.

III. Literature Review

Successful application of CNNs in computer vision and machine learning tasks motivated researchers to use CNN in their presented HSV/R systems [8]. Khalajzadeh et al. proposed an offline HSV method using CNN but only considered random forgeries and didn't considered skilled forgeries [4]. This work used Multilayer Perceptron (MLP) for classification. Using a private dataset of Persian signatures gathered by 22 writers, the verification rate of 99.86% has been reported. Cozzens et al. used a CNN model for HSV and reported the rate of 83% on SigComp2011 signature dataset [16]. A classification method named Deep Multitask Metric Learning (DMML) for offline HSV has been presented by Soleimani et al. [5]. This work reported equal error rates of 15.08% and 17.45% for verifying skilled forgery on GPDS960GraySignature and UTSig datasets, respectively.

A writer-independent feature learning for offline HSV using deep CNN has been presented by Hafemann et al. in 2016 [6]. This work reported false rejection rate of 19.81% and false acceptance rate of 5.99% on GPDS-160 signature dataset. Also, Hafemann et al. [17] used deep CNNs to analyze features learned for offline HSV.

In the following, Hafemann et al. [18], proposed a method for learning the representations directly from signature images using deep CNNs. The performance of the method was evaluated on four datasets: GPDS-960, CEDAR, MCYT-75, and Brazilian (PUC-PR) and the equal error rate of 1.72% was obtained using GPDS-160 signature dataset. It should be noted that this work proposed two pre-trained models called SigNet and SigNet-F. To the best of our acknowledge, these models are only CNN models already trained on signature datasets and their weights are freely available to the research community. These models have been trained on some popular offline signature datasets including: GPDS-960, CEDAR, MCYT-75, and Brazilian (PUC-PR). SigNet model has been trained using just genuine signatures. However, training of the SigNet-F model has been conducted using genuine signatures and a subset of skilled forgeries.

IV. Signature Verification Based on Deep Transfer Learning

Recently, as mentioned in the previous section, some CNN models are introduced and used by authors in their presented HSV/R systems. Therefore, their results should be compared with the performance of some benchmarks CNN models. To the best of our knowledge, there is no study to give a benchmark for authors, in the field of signature processing to compare their results with the current CNN models, till today. In order to address this issue, some prominent CNN models including SigNet, SigNet-F (already trained on signature datasets), VGG16, VGG19, InceptionV3, and ResNet50 (trained on ImageNet dataset) have been considered and the performance of these models have been evaluated using transfer learning, in this work.

The structure of the CNN model is considered fixed and the forward-propagation from the first layer till the last layer before the final layer (named as feature layer in this study) has been conducted. The output of the feature layer is considered as feature vector for the input signature image. Based on this process, the pre-trained model acts as feature extractor. The extracted feature vectors are used to train a support vector machine (SVM) with Radial Basis Function (RBF) kernel as classifier for decision about genuine or forgery of the input signature [19]. Tables I and II show the considered pre-trained models, its depth, the number of layers, the considered feature layer and the size of the feature vector. It should be noted that depth of a CNN model is defined as the largest number of sequential convolutional or fully connected layers from the the input layer to the output layer.

A. Datasets and Preprocessing

In order to evaluate the performance of transfer learning on the pre-trained models, four datasets have been considered: GPDS synthetic signature dataset [20], and MCYT-75 dataset [21] as two benchmark Latin signature datasets, UTSig (University of Tehran Persian Signature) dataset [22] as a benchmark Persian signature dataset, and FUM (Ferdowsi University of Mashhad) dataset [23] as another Persian signature dataset. The statistics of these datasets have been compared in Table III and some samples of signature in the used datasets have been shown in Fig. 2. In order to enhance the quality of

TABLE I: The pre-trained models, its depth, the number of layers, and image input size.

Pre-trained model	Depth	Number of layers	Image input size
SigNet	7	13	150*220
SigNet-F	7	13	150*220
VGG16	16	41	256*256 (200*200 for FUM-PHSD)
VGG19	19	47	256*256 (200*200 for FUM-PHSD)
ResNet50	50	177	256*256 (200*200 for FUM-PHSD)
InceptionV3	48	316	256*256 (200*200 for FUM-PHSD)

TABLE II: The pre-trained models, the considered feature layer and the size of feature vector.

Pre-trained model	Feature layer	Feature vector size
SigNet	Fully connected (FC7)	2048
SigNet-F	Fully connected (FC7)	2048
VGG16	Max Pooling2D	8*8*512 (6*6*512 for FUM-PHSD)
VGG19	Max Pooling2D	8*8*512 (6*6*512 for FUM-PHSD)
ResNet50	Average Pooling2D	2048 (6*6*512 for FUM-PHSD)
InceptionV3	Mixed10 (before Global Average Pooling)	6*6*2048 (4*4*512 for FUM-PHSD)

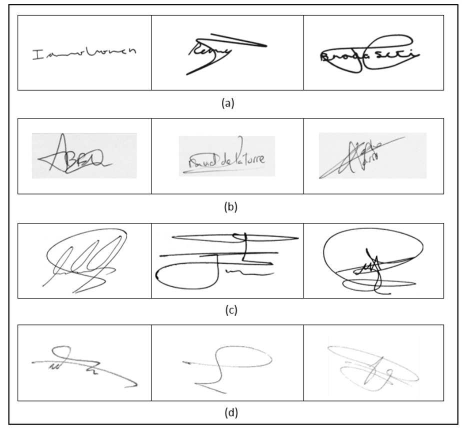

Fig. 2: Three samples of Latin signatures from GPDS synthetic (a), MCYT-75 (b), and Persian signatures from UTSig (c), and FUM-PHSD (d), datasets.

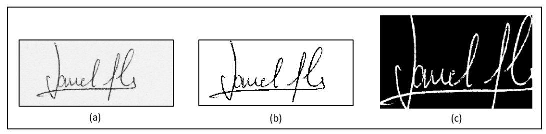

Fig. 3: A signature image from MCYT-75 signature dataset (a), binarization (b), background removal and inversion to zero-valued, noise removal, cropping and resizing the image (c).

the signature images, some preprocessing tasks have been conducted as follows; (i) conversion into the binary format using Otsu's algorithm [1], (ii) inversion the image so that the background is zero-valued, (iii) removal of salt and pepper noise created after binarization using a Gussian filter, (iv) removal of the empty space around the signature image and cropping the image, and (v) normalization of size to 256*256 (except for FUM-PHSD that is 200*200). The preprocessing tasks have been conducted on all signature images of the four datasets in TableIII. Because of the low quality of signature images in the FUM-PHSD dataset, Gaussian filter removed some parts of signature image and so it was not used in the case of FUM-PHSD. Fig.3 shows the output of the preprocessing tasks on a signature image from MCYT-75 dataset.

B. Experimental Protocol and Performance Evaluation

It should be noted that in real application, there is no forgery samples for each user. In this work, similar to [18], the assumption that forgery samples are available for just a small subset of users, has been considered. Therefore, dataset separation has been done as follows: in the case of GPDS synthetic, MCYT-75, and UTSig datasets, two genuine signatures from each user have been considered as positive class in training set and skilled forgery samples from only 20 users considered as negative class of training set. The remaining genuine signatures and skilled forgeries have been considered as testing set. Due to the smaller number of users in FUM-PHSD than other datasets (Table III), skilled forgery samples of only 5 users have been considered as negative class in training set and the remaining process was the same.

TABLE III: The statistics of four datasets used in this work.

Dataset	Users	Genuine	Forgery
GPDS synthetic	4000	24	30 (Simple)
MCYT-75	75	15	15 (Skilled)
UTSig	115	27	6 (Skilled) 36 (Simple)
FUM-PHSD	20	20	10 (Skilled)

SVM classifier in scikit-learn package has been used and the parameters empirically examined and the best results were obtained using the hyper parameter C=1000 with RBF kernel. The performance of a HSV method has been commonly evaluated using Equal Error Rate (EER) in the literature and also used in this work. The EER is computed as the error point that FRR (False Rejection Rate) equals with FAR (False Acceptance Rate) [1]. Table IV shows the obtained experimental results in terms of EER and the best results have been marked. It should be noted that in the case of GPDS Signaturesynthetic, due to the limitations on computer power, only the first 100 users have been considered during our experiments.

C. Visualizing The Signatures in Feature Space Using t-SNE Algorithm and Discussion of Verification Results

t-Distributed Stochastic Neighbor Embedding (t-SNE) algorithm as a successful method has been presented for dimensionality reduction introduced by van der Maaten and Hinton in 2008 [24]. t-SNE technique is used for visualizing high-

TABLE IV: Obtained signature verification results in terms of EER (%) using transfer learning on the pre-trained models.

Verification results	SigNet	SigNet-F	VGG16	VGG19	ResNet50	InceptionV3
GPDS Synthetic (100 users)	**2.61**	2.79	4.05	3.60	5.00	8.72
MCYT-75	5.73	5.36	4.96	7.43	5.04	5.48
UTSig (Skilled forgery)	**5**	8.59	9.27	9.75	8.49	9.81
UTSig (Simple forgery)	4.35	4.39	6.71	6.90	4.25	6.45
FUM-PHSD	8.45	6.11	**4.27**	5.00	5.98	5.00

dimensional data into a low-dimensional space of two or three. Here, t-SNE algorithm has been used to visualize the feature vectors from high dimensional feature space (Table II) into two dimensional space. In order to better intuition about how to separation of signatures in the space of the learned features induced by each of the pre-trained models, Fig.4 and 5 show 2D projections of the feature vectors corresponding to the best and the worst results in Table IV, for 50 users from GPDS synthetic, MCYT-75, and UTSig datasets and for 10 users in FUM-PHSD dataset.

As shown in Table IV, the pre-trained models: SigNet, VGG16, SigNet-F, ResNet50, VGG19, and InceptionV3 have the best performance (in average) on all datasets, respectively. It should be noted that each two models with similar architecture (Tables I and II), SigNet and ResNet50, SigNet-F and VGG16, VGG19 and InceptionV3, have similar performance on datasets. As expected, since the CNN models SigNet and SigNet-F have already been trained on signature datasets (Sec.III), the feature extracted based on these models are more discriminately and their performance is high in comparison with other models. On the other hand, the lowest performance in Table IV is corresponding to VGG19 and InceptionV3 which verifies that the HSV problem does not need very deep neural network such as VGG19 and InceptionV3 and acceptable HSV results can be obtained by shallow learning models such as SigNet and VGG16.

D. Comparison of The Obtained Signature Verification Results With Literature

In this section, obtained experimental results have been compared with similar works in the literature. In the case of GPDS Synthetic dataset, only two works used CNN in their presented HSV methods [5], [18]. However, some of the presented HSV works have been compared in Table V. As shown in this table, the obtained HSV results on GPDS Signature synthetic dataset is close to the state-of-the-art by Hafemann et al. [18] which has been obtained using SigNet-F. But in our experiments, the best EER obtained by SigNet with EER=2.61% and then by SigNet-F with EER=2.79% (Table IV), using only 2 genuine samples per user in comparison with 12 samples used by Hafemann et al. [18].

In the case of MCYT-75, the best obtained result obtained by VGG16 (EER=4.96%) is close to the state-of-the-art (EER=2.87%) by Hafemann et al. [18] which has been obtained by the SigNet model. It seems the lower number of

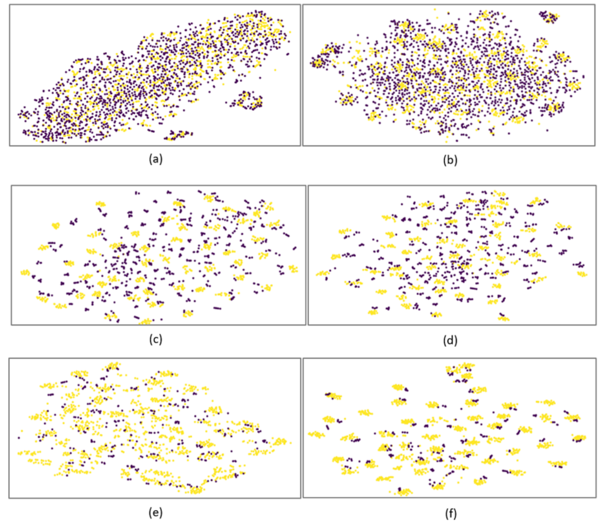

Fig. 4: Separation of signature images in the feature space for the best and the worst results in Table IV. Genuine signatures are shown in yellow (light) and the forgeries are in blue (dark) colors. (a) and (b) are separation of signatures of 50 users in GPDS synthetic dataset with SigNet and InceptionV3, respectively. (c) and (d) are separation of signatures of 50 users in MCYT-75 dataset with VGG16 and VGG19, respectively. (e) and (f) are separation of signatures of 50 users in UTSig dataset with SigNet and InceptionV3 for verifying skilled forgery, respectively.

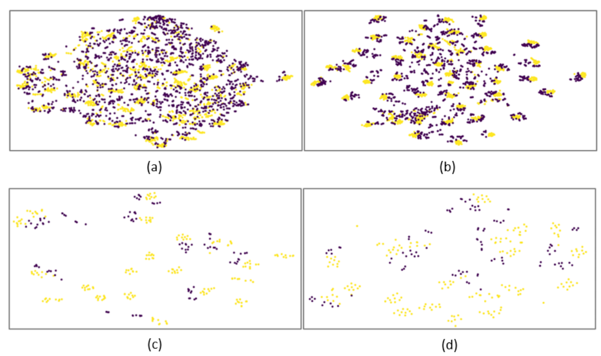

Fig. 5: Separation of signature images in the feature space for the best and the worst results in Table IV. (a) and (b) are separation of signatures of 50 users in UTSig dataset with ResNet50 and VGG19 for verifying simple forgery, respectively. Finally, (c) and (d) are separation of signatures of 10 users in FUM-PHSD dataset with VGG16 and SigNet, respectively.

genuine signatures, two samples per user, in comparison with 10 samples used by Hafemann et al. [18] leads to the lower EER by us. However, the obtained EER=5.73% by SigNet and EER=5.36% by SigNet-F (Table IV), outperforms other works in literature (except [18])) which verifies the power of SigNet and SigNet-F to extract discriminant features for HSV.

About UTSig dataset, Table V shows that the obtained EER= 5.00% outperforms the state-of-the-art (EER=15.83) and verifies the power of SigNet model. Finally, in the case of FUM-PHSD dataset, VGG16 could outperform the state-of-the-art. Notably, as shown in Table IV, EER=8.45% and EER=6.11% have been obtained by SigNet and SigNet-F, respectively, and verifies that these models also outperform the state-of-the-art (EER=15%). Therefore, SigNet and SigNet-F as two

CNN models pre-trained on the signature datasets are the best choices for extracting discriminant features from signature images. The next rank is related to VGG16 as another CNN model already trained on ImageNet dataset.

TABLE V: Comparison of the proposed signature verification method with literature on four datasets.

Dataset	Methods	Type	Users	# samples per user	EER(%)
GPDS Synthetic	Soleimani et al. [5]	WI+WD	300	10	20.94
	Serdouk et al. [25]	WD	100	16	12.52
	Hafemann et al. [18]	WD	160	12	1.72
	Hafemann et al. [18]	WD	300	12	1.69
	Proposed method (using SigNet)	WI	100	2	2.61
MCYT-75	Soleimani et al. [5]	WI+WD	75	5	13.44
	Ooi et al. [26]	WI	75	10	9.87
	Hafemann et al. [18]	WD	75	10	2.87
	Proposed method (using VGG16)	WI	75	2	4.96
UTSig	Soleimani et al. [5]	WI+WD	115	12	17.45
	Soleimani et al. [22]	WD	115	12	32.46
	Soleimani et al. [27]	WD	115	-	15.83
	Proposed method (using SigNet)	WI	115	2	5.00
FUM-PHSD	Sigari et al. [23]	WD	20	10	15
	Proposed method (using VGG16)	WI	20	2	4.27

V. Signature Recognition Based on Deep Transfer Learning using the Convolutional Neural Network Models

In this section, transfer learning of six pre-trained models including: SigNet, SigNet-F, VGG16, VGG19, InceptionV3, and ResNet50 have been used for HSR task. Using network-based deep transfer learning, the pre-trained models have been utilized as feature extractor for signature images. Final decision for the class of the query signature has been conducted using SVM with RBF kernel. The presented methodology for HSR is similar to the HSV (Sec.IV) except that for HSR only the genuine signatures in datasets are considered and the class of the query signature is determined using a classifier such as SVM with RBF kernel which has been used here.

A. Experiments

Four datasets including GPDS synthetic, MCYT-75, UTSig, and FUM-PHSD (Sec. IV-A), have also been used for HSR experiments. Dataset separation has been conducted as follows; 20% of data has been considered as validation set, 20% of data left as test set, and the remaining of data, i.e. 60%, considered as training set. The performance of the HSR method has been evaluated using true recognition rate in percent, i.e. the fraction of signature samples which have been truly classified. Table VI shows the obtained HSR results. As shown in this table, the best HSR results have been obtained using VGG16 as the pre-trained model on ImageNet dataset. In the case of MCYT-75 and FUM-PHSD datasets the accurate recognition rate have been obtained using the VGG16. Also, high recognition rates of 99.79% and 98.71% have been obtained for GPDS synthetic and UTSig datasets using the VGG16 as well.

TABLE VI: Obtained signature recognition results (%) using transfer learning on the pre-trained models.

Recognition results	SigNet	SigNet-F	VGG16	VGG19	ResNet50	InceptionV3
GPDS Synthetic (100 users)	92.29	90.83	**99.79**	99.58	87.65	97.92
MCYT-75	96.89	96.89	**100**	99.11	92.44	98.67
UTSig	89.53	88.24	**98.71**	98.55	92.59	97.58
FUM-PHSD	100	100	**100**	100	100	100

B. Visualizing The Signatures in Feature Space Using t-SNE Algorithm and Discussion of Recognition Results

In this section, t-SNE algorithm has been used to visualize the signature images in the learned feature space [24]. Figs.6 and 7 show this separation for the best and the worst results in Table VI. These figures graphically verify that the best separation between signature classes has been conducted by the feature vectors obtained from the pre-trained model on VGG16 as feature extractor.
The best performance for HSR has been obtained by VGG16, VGG19, InceptionV3, SigNet, SigNet-F, and ResNet50, respectively. Therefore, the pre-trained models on ImageNet dataset could higher performance than SigNet and SigNet-F which pre-trained on signature datasets with the aim of the verifying signature as genuine or forgery samples. It should be noted that this results was expected since the models VGG16, VGG19, and InceptionV3 are among the winners at LISVRC challenge for image classification task (Sec.II).

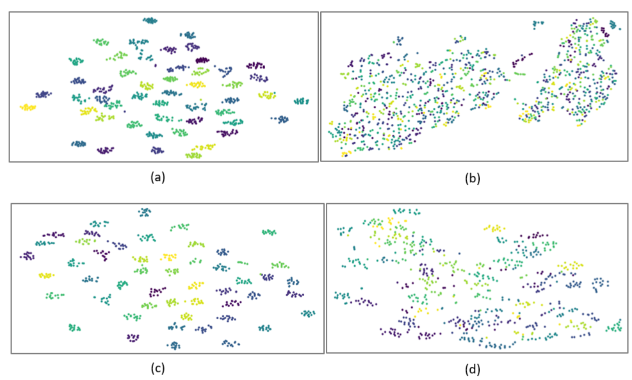

Fig. 6: Separation of signature images in the feature space for the best and the worst results in Table VI. (a) and (b) are the separation of signature classes for 50 users in GPDS synthetic dataset with VGG16 and ResNet50, respectively. (c) and (d) are separation of signature classes for 50 users in MCYT-75 dataset with VGG16 and ResNet50, respectively.

C. Comparison of The Obtained Signature Recognition Results With Literature

Unlike the HSV, only few works have been presented for HSR in the literature and non of them used CNNs. Table VII compares the obtained HSR results with some presented works. As shown in this table, obtained HSR results outperform the state-of-the-art.

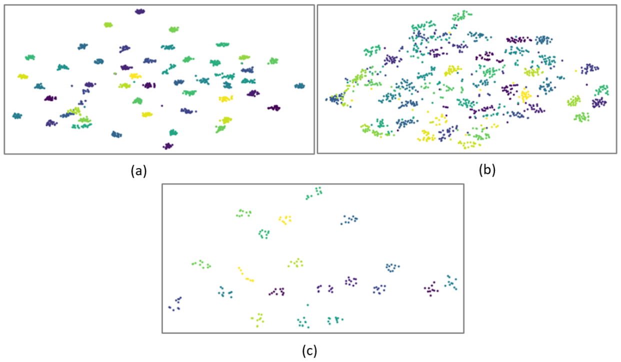

Fig. 7: Separation of signature images in the feature space for the best and the worst results in Table VI. (a) and (b) are separation of signatures for 50 users in UTSig dataset with VGG16 and SigNet-F. Finally, (c) is the separation of signatures of 10 users in FUM-PHSD dataset with VGG16.

TABLE VII: Comparison of the obtained signature recognition results (%) with the state-of-the-art on four datasets.

True recognition rate	GPDS synthetic	MCYT-75	UTSig	FUM-PHSD
Calik et al. [28]	96.91 (4000 users)	96.41	-	-
Fakhlai et al. [29]	-	-	-	98
Pourshahabi et al. [30]	-	-	-	100
Hezil et al. [31]	-	97.30	-	-
Proposed method (VGG16)	**99.79** (100 users)	**100**	**98.71**	**100**

VI. Conclusion and Future Works

In this paper, the power of transfer learning on the pre-trained CNN models has been evaluated for offline HSV and HSR. The pre-trained models: SigNet, SigNet-F, VGG16, VGG19, InceptionV3, and ResNet50 have been used as feature extractor for HSV/R systems and the classifier SVM with RBF kernel has been used for final decision about the query signature. Experimental results verified the power of the pre-trained model VGG16 and SigNet in HSV and HSR.

As a future work, we will consider the fine tuning of the CNN models with tuning learning rates for first, middle and final layers of a pre-trained model, increasing or decreasing layers of the models or changing type of layers to get better results for HSV and HSR. Also, the impact of data augmentation with fine tuning in HSV and HSR will be evaluated in the future.

References

[1] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Offline handwritten signature verification, literature review," IEEE, Seventh Int. Conf. Image Process. Theory, Tools Appl. (IPTA), pp. 1-8, 2017.
[2] J. Yosinski, J. Clune, and Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," Adv. neural inform. process. syst., pp. 3320-3328, 2014.
[3] H. Srinivasan, S. N. Srihari, and M. J. Beal, "Machine learning for signature verification," Comput. Vis. Graphics Image Process., pp. 761-775, Springer 2006.
[4] H. Khalajzadeh, M. Mansouri, and M. Teshnehlab, "Persian signature verification using convolutional neural networks," Inte. J.Engin. Res. Technol., vol. 2(1), pp. 7-12, 2012.
[5] A. Soleimani, B. N. Araabi, andK. Fouladi, "Deep multitask metric learning for offline signature verification," J. Pattern Recognit. Lett., vol. 80, pp. 84-90, 2016.
[6] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Writer-independent feature learning for offline signature verification using deep convolutional neural networks," IEEE. Int. Joint Conf. Neural Net. (IJCNN), pp. 2576-2583, 2016.
[7] L. Nanni, S. Ghidoni, and S. Brahnam, "Handcrafted vs. non-handcrafted features for computer vision classification,"J. Pattern Recognit., vol. 71, pp. 158-172, 2017.
[8] O. M. Parkhi et al., "Deep face recognition.," bmvc., vol. 1(3), pp. 6, 2015
[9] S. J. Pan and Q. Yang, "A survey on transfer learning. IEEE Transaction on Knowledge Discovery and Data Engineering," IEEE press, vol. 22(10), 2010.
[10] M. Kaboli, "A Review of Transfer Learning Algorithms," Technische Universität München, 2017.
[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.
[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," IEEE Conf. Comput. Vis. Pattern Recognit., pp. 2818-2826, 2016.
[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," IEEE Conf. Comput. Vis. Pattern Recognit., pp. 770-778, 2016.
[14] R. Olga R. et al., "ImageNet Large Scale Visual Recognition Challenge," Int. J. Comput.Vis. (IJCV), vol. 115(3), pp. 211-252, 2015
[15] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," IEEE Conf. Comput. Vis. Pattern Recognit., pp. 248-255, 2009.
[16] B. Cozzens, et al., "Signature Verification Using a Convolutional Neural Network," 2017.
[17] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Analyzing features learned for offline signature verification using Deep CNNs," 23rd Int. Conf. Pattern Recognit. (ICPR), pp. 2989-2994, 2016.
[18] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Learning features for offline handwritten signature verification using deep convolutional neural networks," J. Pattern Recognit. vol. 70, pp. 163-176, 2017.
[19] C. Cortes and V. Vapnik, "Support-vector networks," J. Mach. learn., vol. 20(3), pp. 273-297, 1995.
[20] M. A. Ferrer, M. Diaz Cabrera, and A. Morales, "Synthetic off-line signature image generation," IEEE., Int. conf. biometrics (ICB), pp. 1-7, 2013.
[21] J. Fierrez-Aguilar, N. Alonso-Hermira, G. Moreno-Marquez, and J. Ortega-Garcia, "An off-line signature verification system based on fusion of local and global information," Int. Workshop on Biometric Authentication, pp. 295-306, 2004.
[22] A. Soleimani, K. Fouladi, and B. N. Araabi, "UTSig: A Persian offline signature dataset," IET Biometrics, vol.6(1), pp. 1-8, 2016.
[23] M. H. Sigari, M. R. Pourshahabi, and H. R. Pourreza, "Offline handwritten signature identification and verification using multi-resolution gabor wavelet," Int. J. Biometrics Bioinform. (IJBB), vol.5(4), pp. 234-248, 2011.
[24] VL. an Der Maaten, "Accelerating t-SNE using tree-based algorithms," J. Mach. Learn. Res., vol.15(1), pp. 3221-3245, 2014.
[25] Y. Serdouk, H. Nemmour, and Y. Chibani, "New Off-line Handwritten Signature Verification Method based on Artificial Immune Recognition System," Expert Syst. Appl., vol. 51, 2016.
[26] S. Y. Ooi,A. B. Teoh, Y. H. Pang, and B. Y. Hiew, "Image-based handwritten signature verification using hybrid methods of discrete Radon transform, principal component analysis and probabilistic neural network," J. Applied Soft Comput., vol. 40, pp. 274-282, 2016.
[27] A. Soleimani, K. Fouladi, and B. N. Araabi, "Persian offline signature verification based on curvature and gradient histograms," Int. Conf. Comput, Knowledge Engin, (ICCKE), pp. 147-152, 2016.
[28] N. Calik, O. C. Kurban, A. R. Yilmaz, and T. Yildirim, "Large-scale offline signature recognition via deep neural networks and feature embedding," J. Neurocomputing, pp. 0925-2312, 2019.
[29] M. Fakhlai, H. R. Pourreza, R. Moarefdost, and S. Shadroo, "Off line signature recognition based on contourlet transform," Int. Conf. Mach. Learn. Comput. (IPCSIT), vol. 3, 2011.
[30] M. R. Pourshahabi, M. H. Sigari, and H. R. Pourreza, "Offline handwritten signature identification and verification using contourlet transform," Int. Conf. Soft Comput. Pattern Recognit., pp. 670-673, 2009.
[31] H. Hezil, R. Djemili, and H. Bourouba, "Signature recognition using binary features and KNN," Int. J. Biometrics, vol. 10(1), pp. 1-15, 2018.