Sahand Communications in Mathematical Analysis (SCMA) Vol. 17 No. 3 (2020), 1-31
http://scma.maragheh.ac.ir
DOI: 10.22130/scma.2019.99098.536

Use of the Shearlet Transform and Transfer Learning in Offline Handwritten Signature Verification and Recognition

Atefeh Foroozandeh[1], Ataollah Askari Hemmat[2*] and Hossein Rabbani[3]

Abstract. Despite the growing growth of technology, handwritten signature has been selected as the first option between biometrics by users. In this paper, a new methodology for offline handwritten signature verification and recognition based on the Shearlet transform and transfer learning is proposed. Since, a large percentage of handwritten signatures are composed of curves and the performance of a signature verification/recognition system is directly related to the edge structures, subbands of shearlet transform of signature images are good candidates for input information to the system. Furthermore, by using transfer learning of some pre-trained models, appropriate features would be extracted. In this study, four pre-trained models have been used: SigNet and SigNet-F (trained on offline signature datasets), VGG16 and VGG19 (trained on ImageNet dataset). Experiments have been conducted using three datasets: UTSig, FUM-PHSD and MCYT-75. Obtained experimental results, in comparison with the literature, verify the effectiveness of the presented method in both signature verification and signature recognition.

1. Introduction

With the increasing advancement of technology, artificial intelligence becomes more tangible in different aspects of daily life. In 2017, Mark Cuban, a successful entrepreneur, said: "artificial intelligence, deep learning, machine learning, whatever you're doing, if you don't under-

2010 *Mathematics Subject Classification.* 65T60, 35Q79.
Key words and phrases. Offline handwritten signature, Signature verification, Signature recognition, Shearlet transform, Transfer learning.
Received: 06 December 2018, Accepted: 13 July 2019.
* Corresponding author.

stand it, learn it. Because, otherwise you're going to be a dinosaur within 3 years."[1]

An important field of artificial intelligence includes biometric systems. Among these systems, verification and recognition systems based on fingerprint, iris, voice and handwritten documents including signatures are the most popular biometric systems. Handwritten signatures are daily used to register ownership in banking systems (including checks), as well as in administrative and financial applications, all over the world.

It should be noted that automatic signature verification/recognition systems receive handwritten signatures in two ways: offline and online. Offline signature processing is a more difficult and challenging task than online processing because only the scanned images of signatures are received by the offline system, whereas for online signature, system has access to more information (dynamic information) such as spatial coordinates, pressure applied to the pen tip, axial angles and angle of inclination [60, 75].

The aim of a signature recognition system is identifying the owner of the input signature. However, a signature verification system aims to verify the individuals, i.e. the input signature is genuine or it is a forgery sample [69]. A signature verification system deals with three kinds of forgeries: random, simple and skilled forgeries [3]. In the case of random forgery, the forger without any information about the author's name and his/her signature, reproduces a random signature. If the forger knows the author's name but does not access to any signature sample, the reproduced signature is known as a simple forgery. In the case of skilled forgery, a forger has signature samples and tries to reproduce them. Therefore, skilled forgeries are much more similar to the genuine signatures than random and simple forgeries.

Based on the 2017 AFP Payments Fraud and Control Survey, handwritten checks are used in the most of business transactions (8 out of 10)[2] and some forms of payment fraud including signature forgery have occurred in seventy-four percent of organizations including banks [59, 71]. Hence, researchers have paid special attention to the problem of verification of skilled forgeries [1, 21, 27, 47] and this is also the focus of this paper.

The most important visual features of signature images are edges and corners and hence precise encoding of these features is the most important issue in a signature verification/recognition system. Sparse

[1] https://bothsidesofthetable.com/mark-cuban-on-why-you-need-to-study-artificial-intelligence-or-youll-be-a-dinosaur-in-3-years-db3447bea1b4

[2] https://www.afponline.org/trends-topics/topics/articles/Details/afp-survey-payments-fraud-hits-record-high-of-78

representation systems especially wavelet theory play an important role in the encoding of the visual features of images. However, traditional wavelet transforms cannot effectively encode edge and corner [16] and we need x-let transforms which were introduced much later. Among these transforms, shearlet transform is a newly proposed multi-scale and multi-directional transform [42]. Thanks to the special properties of shearlets, they have become an efficient tool for detection and analysis of edges and corners in images [16, 24, 34]. Shearlet transform has been used in the signature verification/recognition method proposed in this paper.

Recently, deep learning algorithms and Convolutional Neural Network (CNN) models have received special attention in signature verification /recognition systems [11, 29, 35]. Convolutional neural network models are used to obtain a representation of input images at multiple levels [5] and need a large dataset to be able to tune their many parameters. This process suffers from high computational complexity and it is very time consuming. The concept of transfer learning proposed by Perkins and Salomon enables us to use a smaller dataset and utilize the potential of the convolutional models trained on large reference datasets [52]. We prove that using a combination of two powerful tools, including the shearlet transform and transfer learning, can improve the performance of a signature verification/recognition system.

The rest of the paper is organized as follows: In Sec. 2, a brief overview of traditional wavelet transforms, Shearlet transforms, machine learning, deep learning, convolutional neural network, pre-trained models and transfer learning along with a literature review are presented. Sec. 3 describes the proposed methodology along with the experimental results for offline handwritten signature verification and signature recognition. Finally, Sec. 4 is dedicated to our conclusion and future works.

2. Basic Concepts and Review of Literature

2.1. **Wavelet Transform and Its Limitations.** The applicability of the theory of wavelets, especially in signal analysis and processing, is well known to those applied mathematicians and computer scientists whose fields of research deal with univariate and multivariate signals having singularities including isotropic and anisotropic features [42]. This theory provides a very effective methodology for analyzing isotropic features (point singularities) in univariate signals [70]. The reason is that wavelet coefficients around the point discontinuity are considerably larger than the coefficients pertaining to smooth regions [58]. The Continuous Wavelet Transform (CWT) of a signal $f \in L^2(\mathbb{R})$ is defined by

[70]

$$W_{a,b}(f) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(t)\psi^* \left(\frac{t-b}{a} \right) dt, \tag{2.1}$$

where a and b are respectively the scaling and translation parameters. As seen in Eq. (2.1), wavelets are generated using scaling and translation of a generating "mother wavelet" ψ. However, in applications we need to use Discrete Wavelet Transforms (DWT); discretization is carried out by discretizing the scaling and translation parameters [70].

Although wavelets are good tools when dealing with univariate functions, they fail to detect geometric features in multivariate cases [42]. Several approaches have been proposed for extending the benefits of wavelets to higher dimensions. These approaches are have two major objectives: to optimally encoding anisotropic features and dealing with continuous and discrete domain in a unified manner [42]. Again, there are two prevalent ways for extending wavelets to higher dimensions: using tensor products of wavelets in lower dimensions (separable wavelets) and directly defining wavelets in higher dimensions [12].

We first consider two dimensional separable wavelets. Applying such wavelets to an image generates four different subbands including one subband of low-frequency information (approximation coefficients) and three subbands of high-frequency information (detail coefficients). However even these detail coefficients describe the features of the image in only three directions: horizontal, vertical and diagonal. This description contains very little information and is not of much use for an optimally encoding of the image [42]. In order to deal with this deficiency in higher dimensional separable wavelets, other x-lets such as complex wavelet transforms [37, 38], curvelets [8, 9], contourlets [15] and shearlets [45] have been proposed.

Complex wavelets perform much better than traditional wavelets, in many applications. However, they still do not provide an optimally encoding of the image [42]. This is because the complex wavelet transform is not a true extension of wavelets [42]. Curvelets are defined on the continuous domain and cannot be efficiently discretized [70]. Contourlets are constructed using a combination of a multiscale and a directional filter bank and fewer directional features which lead to artifacts in denoising and compression by contourlets [42].

The shearlet transform was introduced to maintain the advantages of wavelets in higher dimensions while overcoming the deficiencies of these x-lets. The advantages of shearlet transform over wavelets are discussed in the next section.

2.2. **The Shearlet Transform and Its Advantages Over the Wavelet Transform.** The shearlet transform was introduced by Guo, Kutyniok, Labate, Lim and Weiss in 2006 [34, 42]. As mentioned in Subsection 2.1, several approaches have been proposed for overcoming the limitations of wavelets when dealing with singularities in higher dimensions and especially for 2D signals. Among these, shearlets as a multi-scale and multi-directional transform stand out as an efficient tool for addressing anisotropic and directional information at different scales [7, 25, 76]. This popularity is due to a number of desirable properties of the shearlet transform. Some of which have been listed below [16]:

- A shearlet transform deals with continuous and discrete domains in a unified manner.
- Only a finite set of functions is needed to construct a shearlet system.
- The analyzing functions in a shearlet system usually have compact support in both space and frequency domains.
- Implementation of shearlets can be carried out using fast algorithms.
- Optimally encoding of anisotropic features can be obtained by shearlet transform.

For precise mathematical proofs of the above mentioned facts we refer to [16, 30, 42].

Since signature verification/recognition are among two dimensional image processing problems, we shall focus only on two dimensional shearlets, in this paper. All notations used are consistent with the notations in [30]. In order to define a continuous shearlet transform on $L^2(\mathbb{R}^2)$, we need an anisotropic dilation (or scaling) matrix (A_a), a shear matrix (S_s) and a translation parameter $(t \in \mathbb{R})$, as follows [30]:

$$A_a = \begin{bmatrix} a & 0 \\ 0 & \sqrt{a} \end{bmatrix}, \quad a \in \mathbb{R}^+, \qquad S_s = \begin{bmatrix} 1 & s \\ 0 & 1 \end{bmatrix}, \quad s \in \mathbb{R}. \tag{2.2}$$

Now, shearlets, $\psi_{a,s,t}$, are defined using the dilation, shearing and translation of a generating function $\psi \in L^2(\mathbb{R}^2)$, called "mother shearlet", as follows [30]:

$$\psi_{a,s,t} = a^{-\frac{3}{4}} \psi \left(A_a^{-1} S_s^{-1} (x - t) \right). \tag{2.3}$$

The scaling parameter measures the level of resolution, the directional parameter measures directionality and the translation parameter shows position. Therefore, shearlets are more efficient tool for encoding directional information such as edges and corners than traditional wavelets [76]. For example, Fig. 1, shows a dilated and sheared version of a shearlet in Fourier and time domains. The shearlet transform, $SH_\psi(f)$, of a

signal $f \in L^2(\mathbb{R})$ is defined as follows [30]:

$$SH_\psi f(a,s,t) = \langle f, \psi_{a,s,t} \rangle. \tag{2.4}$$

To the best of our knowledge, three toolboxes have been presented to

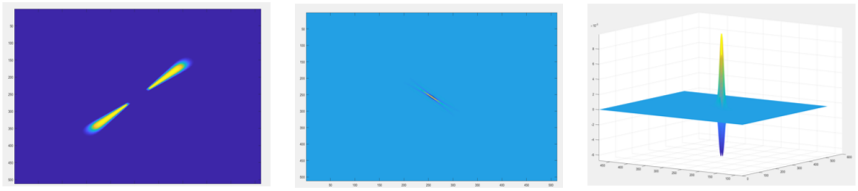

FIGURE 1. Shearlet in Fourier domain (a), time domain (b) and a 3D view of its time domain (c).

numerical implementation of shearlets, until today. These include the Local Shearlet Toolbox[3] [17], ShearLab[4] [43, 44] and the Fast Finite Shearlet Transform (FFST)[5] [30]. Among these, FFST has advantages such that discretization used in FFST is applied to all of the discrete grid, it works using a translation invariant shearlet [30] and the Fast Fourier Transform (FFT) is used to compute the shearlet coefficients. To implement our method we have used Python 3.6. So, the python version of FFST implementation has been used[6].

Consider a digital image, $f(x,y)$, of dimension $M \times N$ sampled on a discrete grid $\{(\frac{m_1}{M}, \frac{m_2}{N}); (m_1,m_2) \in \Omega\}$, where $\Omega := \{(m_1,m_2); m_1 = 0,\dots,M-1, m_2 = 0,\dots,N-1\}$. The following dilation, shearing and translation parameters are used in FFST for discretization on this grid.

$$\begin{array}{ll} a_j := 2^{-2j}, & j = 0,\dots,j_0-1, \\ s_{j,k} := k2^{-j}, & -2^j \le k \le 2^j, \\ t_m := \left(\frac{m_1}{M}, \frac{m_2}{N}\right), & m \in \Omega. \end{array} \tag{2.5}$$

Using the above discretization, the discrete shearlets are $\psi_{j,k,m}(x) = \psi_{a_j,s_{j,k},t_m}(x)$. These shearlets can decompose the function f on the entire Fourier domain tilling using one square and several cones as shown in Fig.2. The approximation of the function f on the square is obtained using a scaling function ϕ. Details of function f on the cones are computed using the generated shearlet ψ. Therefore, the discrete shearlet transform of f, is defined as follows:

[3] http://www.math.uh.edu/~dlabate/software.html
[4] http://www.shearlab.org
[5] https://www.mathematik.uni-kl.de/imagepro/forschung/software/ffst/
[6] https://github.com/grlee77/PyShearlets

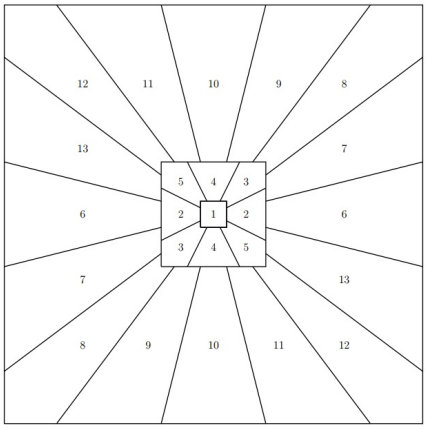

FIGURE 2. Frequency tiling with respective indices for η (figure from [30]).

$$SH(f)(\kappa,j,k,m) := \begin{cases} \langle f, \phi_m \rangle, & \kappa = 0, \\ \left\langle f, \psi^{\kappa}_{j,k,m} \right\rangle, & \kappa \in \{h, v\}, \\ \left\langle f, \psi^{h\times v}_{j,k,m} \right\rangle, & \kappa = h \times v, |k| = 2^j. \end{cases} \tag{2.6}$$

where, $j = 0, \ldots, j_0 - 1, -2^j + 1 \leq k \leq 2^j - 1$ and $m \in \Omega$.

Shearlets efficiently encode the directional details of an image. Moreover, if the generating function ψ has the admissibility property, the shearlet transform is invertible [30] and in this case a perfect reconstruction of the image is possible using its shearlet coefficients. This is true for shearlets used in FFST implementation [30]. The discrete shearlet system obtained in FFST provides a Parseval frame for $L^2(\mathbb{R}^2)$ using a ψ which is not defined directly. In other words, Eq. (2.7) is true and therefore the shearlet transform is invertible [42].

$$\|f\|_H^2 = \langle f, \psi_{a,s,t} \rangle. \tag{2.7}$$

In this paper, FFST is used to compute the shearlet transform of the signature image. It should be noted that based on FFST implementation [30], shearlet transform of a signature image of size $M \times N$, saved in a three dimensional matrix of size $M \times N \times \eta$. The third parameter, η, is proportional to the size of the image in hand and is related to the respective shear, one for lowpass (square in Fig. 2) and other numbers are related to the different shears and scales (cones in Fig. 2)[30].

The partition of the frequency domain used in this work has the same value of η as in [30] and is shown in Fig. 2. For example, for a signature image of size 512×512, the shearlet coefficients are saved in a matrix of

size $512 \times 512 \times 61$. Fig. 3, shows a signature image from the MCYT-75 signature dataset together with the respective shearlets and shearlet coefficients in Fourier domain for $\eta = 13$ and $\eta = 23$.

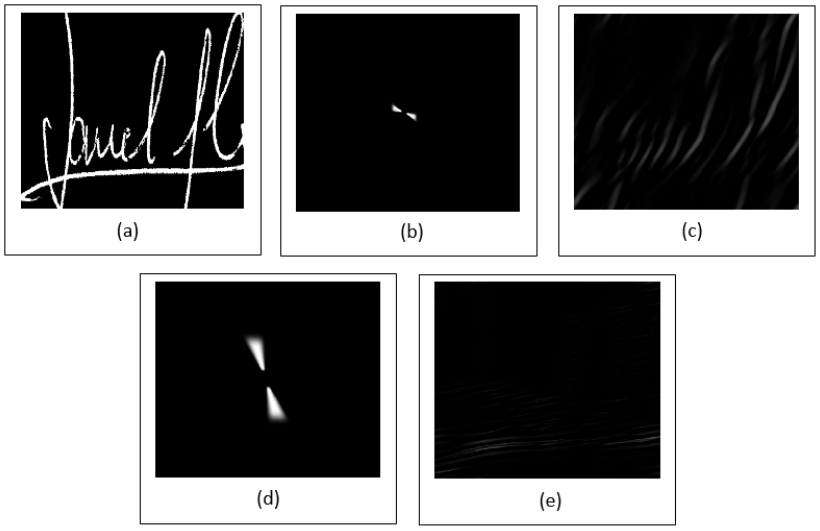

FIGURE 3. (a) A signature image from MCYT-75 signature dataset. (b) Two shearlets in Fourier domain related to $\eta = 13$ and, (d) $\eta = 23$. (c) Corresponding shearlet coefficients for $\eta = 13$ and (e) for $\eta = 23$.

In the following section, we present a brief review on machine learning concepts used in this paper with the literature review of the related signature verification/recognition.

2.3. **Machine Learning and Deep Learning.** Machine learning is a field of computer science which was first introduced in 1959 [61]. In this field of research, statistical techniques including neural networks are used in order to empower computers to learn and predict from data [39]. Signature verification and signature recognition are machine learning problems which try to learn from signature samples in order to decides and predict about the incoming signatures.

Learning in an automatic signature verification/recognition, may be writer-independent (WI, general learning) or Writer-dependent (WD, special learning) [69]. In the first case, WI, learning is based on a large population of signature samples related to all persons in the dataset, whereas in the second cases learning is based on the signature samples of each person, separately [69]. Although WD learning achieve good

results, for each user added to the system a classifier must be conducted which increases the complexity and cost of the system [18]. In order to reduce the complexity, WI learning attracts more researchers in recent years [23, 29, 41]. In this paper, we consider WI learning.

Deep learning is a branch of machine learning, which uses existing labeled data and powerful hardwares[7] [78]. In this process, learning is done as supervised (such as classification) or unsupervised (such as pattern analysis) manners [13]. Overviews of deep learning concepts, principles, different deep learning models and their developments may be found in [5, 6, 46, 62].

Toolboxes for deep learning include Torch[8], Theano[9], Keras[10], Caffe[11], CuDNN[12], TensorFlow[13], MXNet[14] and deeplearning4j[15]. TensorFlow developed by the Google Brain team within Googles AI (Artificial Intelligence) organization can efficiently perform numerical computations and is an open source software library [77]. Due to the flexible architecture of TensorFlow, computations can be easily performed using different platforms including CPUs, GPUs and TPUs[16]. In this paper, Keras which is simple and high-level model definition interface has been configured to use the TensorFlow backend[17].

2.4. **Convolutional Neural Network.** The Convolutional Neural Network (CNN) is one of the most successful class of deep learning models which has been used in visual tasks since the late 1980s [56]. As in traditional neural networks, convolutional neural networks are made of neurons with weights and biases (so called parameters) to be learned[18]. Compared to the neural networks, CNNs include more complex layers which empower the CNNs for complex tasks [36, 74]. The high number of layers leads to a large set of parameters to be tuned with large datasets. So, training process of a CNN requires a significant amount of computing power. To do this, as mentioned in Sec. 2.3, some platforms like GPU and TPU have been developed.

[7] http://deeplearning.net/

[8] http://torch.ch/docs/getting-started.html

[9] http://www.deeplearning.net/software/theano/index.html

[10] https://keras.io

[11] http://caffe.berkeleyvision.org/

[12] https://docs.nvidia.com/deeplearning/sdk/cudnn-install/index.html

[13] https://www.tensorflow.org/

[14] http://mxnet.io/index.html

[15] https://deeplearning4j.org/

[16] https://blog.algorithmia.com/hardware-for-machine-learning/

[17] https://blog.keras.io/keras-as-a-simplified-interface-to-tensorflow-tutorial.html

[18] http://cs231n.github.io/convolutional-networks/

Recently, CNNs attract the attention of the researchers in the field of signature verification/recognition. Khalajzadeh et al. [35] have proposed an offline signature verification method based on convolutional neural network. In this work, classification was carried out using the Multilayer Perceptron (MLP) network and a validity of 99.86% was reported for a private dataset of Persian signatures gathered by 22 writers. It should be noted that in this work only random forgery has been considered and did not considered skilled forgery. Cozzens et al. [11], used a convolutional neural network for signature verification. This work reported the verification rate of 83% on the SigComp2011 signature dataset. Soleimani [66] proposed a classification method named Deep Multitask Metric Learning (DMML) for offline signature verification. This work used Local Binary Pattern for feature vectors and reported Equal Error Rates of 17.45% and 13.44% on UTSig and MCYT-75 datasets, respectively.

In 2016, Hafemann et al. [29], proposed a WI feature learning for offline signature verification using deep convolutional neural network. This was evaluated on GPDS-160 dataset and reported the false rejection rate of 19.81% and false acceptance rate of 5.99%. In a follow-up [28], deep convolutional neural networks were used to analyze features learned for offline signature verification and then in [26], a method was proposed for learning representations directly from signature images using deep convolutional neural networks. The Equal Error Rates of 2.87%, 4.63% and 1.69% were reported on MCYT-75, CEDAR and GPDS-300 datasets, respectively.

2.5. **Pre-trained Models and Transfer Learning.** The objective of training a deep learning model is finding the appropriate weights for the network by using multiple forward and backward iterations. A large number of layers in such models results in a large number of parameters which have to be trained. Hence, the training task is really time consuming and needs a large dataset such as ImageNet. ImageNet created by Deng, Jia, et al. in 2009 [13, 40] and includes 14,197,122 images of more than 20,000 categories of fruits, animals, etc[19]. Several CNN models have been trained on ImageNet (known as pre-trained models) and the corresponding weights are available to the research community. There are several convolutional neural networks which have been trained on ImageNet such as VGG16, VGG19, InceptionV3, ResNet50 and Xception[20].

VGGNet was one of the famous models submitted to ILSVRC-2014[21] as 1st runner up for image classification task and won the localization

[19] `http://www.image-net.org/`

[20] `https://keras.io/applications/`

[21] ImageNet Large Scale Visual Recognition Challenge

task in ILSVRC 2014[22]. VGG16 is a CNN model proposed by K. Simonyan and A. Zisserman [65]. VGG16[23] 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 stucture with VGG16 but composed of 19 convolutional layers and is a deeper model than VGG16 [65]. It should be noted that the pre-trained models, VGG16 and VGG19, have been considered as strong baseline for many machine learning tasks [57], which used in this study.

In some applications, such as biomedical signal processing, creating a large dataset for optimizing the parameters of a CNN model is difficult and sometimes impossible [32, 73]. In such situations, the architecture and the parameters of a pre-trained model have been utilized by knowledge transfer or transfer learning. The first work on transfer learning in machine learning is dedicated to Lorien Pratt in 1993 [54]. A good survey on transfer learning has been gathered by Pan et al. [51]. In this survey, a formal definition of transfer learning has been presented. 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, respectively. Also, D_T and τ_T indicate target domain and target task, respectively. Transfer learning enables us to learn target conditional probability distribution $f_T(.) = P(Y_T|X_T)$ in D_T based on the knowledge gained from D_S and τ_S where $D_S \neq D_T$ or $\tau_S \neq \tau_T$ [51].

It should be noted that transfer learning is divided into some categories: inductive, transductive and unsupervised transfer learning [32]. The focus of this paper is on inductive transfer learning where there are labeled dataset for both source and target task. Several researches have been conducted related to the use of transfer learning in the field of signature verification/recognition. As mentioned in Sec. 2.4, Hafemann et al. [26], used deep CNN for signature verification and proposed two pre-trained models called SigNet (trained using only genuine signatures) and SigNet-F (trained with a subset of forgery samples in addition to genuine signatures) available to the research community[24]. To the best of our knowledge, these models are only CNN models which have been trained on offline handwritten signature datasets, until today. These models

[22]https://medium.com/coinmonks/paper-review-of-vggnet-1st-runner-up-of-ilsvlc-2014-image-classification-d02355543a11

[23]`https://www.pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/`

[24]`https://github.com/luizgh/sigver_wiwd`

have been trained on some popular offline signature datasets including GPDS-960 [72], CEDAR [33], MCYT-75 and Brazilian (PUC-PR) [22].

In this paper, because of our novelty to use shearlet subbands, in addition to SigNet and SigNet-F (trained on signature datasets), two other pre-trained models including VGG16 and VGG19 (trained on ImageNet dataset) have also been used for transfer learning. It worth noting that the explanation about the layers of these CNN models are not in the scope of this study. However, in order to show the layers, the architecture of VGG16 and SigNet have been shown in Fig. 4.

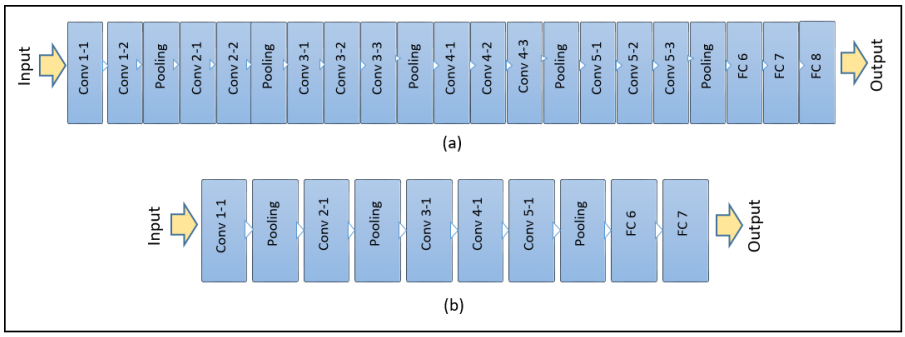

FIGURE 4. The architecture of (a) VGG16 and (b) SigNet models.

In the next section, we discuss the proposed methodology for handwritten signature verification and signature recognition based on the shearlet transform and transfer learning.

3. METHODOLOGY

A block diagram of the proposed method for offline signature verification is shown in Fig. 5. The procedure for the proposed signature recognition method is similar. In what follows, we describe this procedure in detail.

3.1. **Dataset.** The performance of the proposed method has been evaluated on three datasets including MCYT-75 dataset[25] as a large benchmark Latin offline signature dataset [20], UTSig (University of Tehran Persian Signature) dataset[26] as large and public Persian offline signature dataset [67] and FUM-PHSD (Ferdowsi University of Mashhad-Persian Handwritten Signature Database) [64], as another Persian offline signature dataset (with author's permission). These datasets have been commonly used by researchers as benchmark datasets in offline handwritten

[25] https://atvs.ii.uam.es/atvs/mcyt75so.html
[26] http://mlcm.ut.ac.ir/Datasets.html

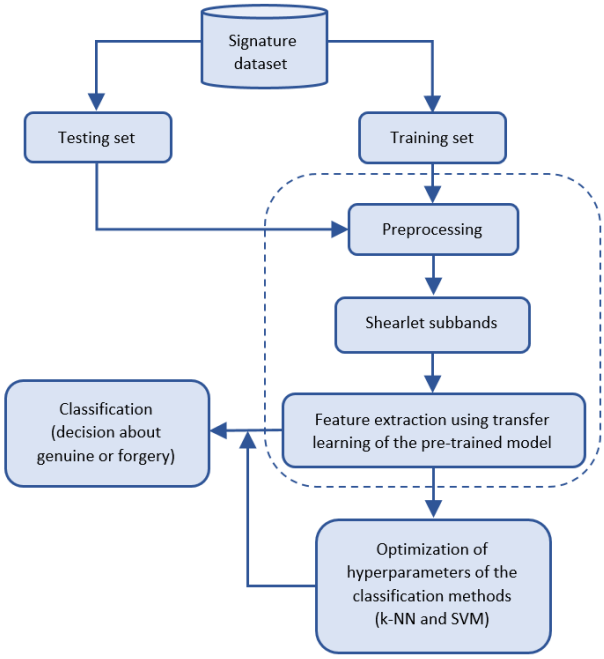

FIGURE 5. Block diagram of the proposed offline signature verification method.

signature field of research [27, 47]. The statistics of these datasets are displayed in Table 1 and samples from these datasets have been shown in Fig 6.

TABLE 1. The statistics of the datasets used in this work.

Dataset	#users	#genuine (per user)	#forgeries (per user)
MCYT-75	75	15	15 (skilled)
UTSig	115	27	6 (skilled) 36 (simple)
FUM-PHSD	20	20	10 (skilled)

3.2. **Preprocessing.** In order to enhance the quality of signature images, some preprocessing tasks have been conducted:

(i) Conversion of the image into the binary format using Otsu's algorithm [50],
(ii) Inversion the image so that 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,
(v) Normalization of size to 512×512.

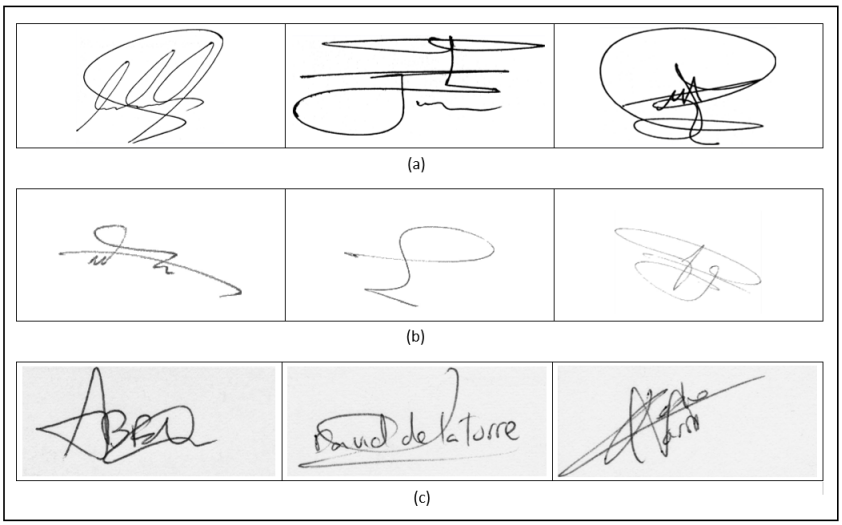

FIGURE 6. Three samples of Persian signatures from (a) UTSig, (b) FUM-PHSD and (c) Latin signatures from MCYT-75 signature datasets.

These preprocessing tasks have been conducted on the three datasets in Table 1. Because of the low quality of signature images in the FUM-PHSD dataset, as shown in Fig. 6(b), Gaussian filter removed some parts of signature image and so it was not used in the case of FUM-PHSD. Fig. 7 shows the output of these preprocessing tasks on a signature image from MCYT-75 dataset.

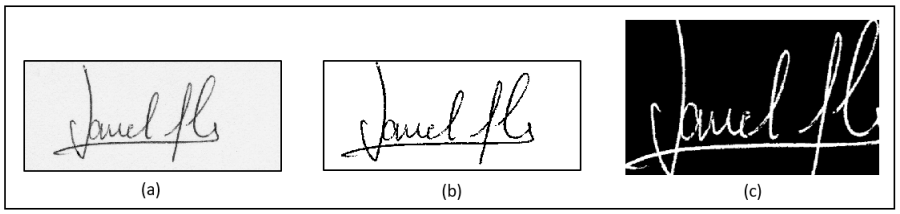

FIGURE 7. (a) A signature image from MCYT-75 signature dataset, (b) binarization, (c) background removal and inversion to zero-valued, noise removal, cropping and resizing the image.

3.3. **Proposed Signature Verification Method.** As mentioned in Sec. 2.2, shearlet high-pass subbands include details of the signature image in specific directions and scales (Fig. 3). Generally, signature samples in a dataset have the same overall shape and differences are in the details, especially in the case of skilled forgery. We also note that the performance of a signature verification/recognition system is directly related to the edge structures encoding tool and the shearlet transform performs well in this regard. This is our motivation to use shearlet transform in the presented method.

Here, using sum of the high-pass subbands of the shearlet transform of signature image, we obtain some approximations for the original signature image as follows:

- Shearlet subbands of four scales (for η=1 to 61) for each signature image have been computed using FFST.
- Corresponding to the indices of the frequency domain in Fig. 2, the following images have been computed and labeled corresponding to each of the original signature images: approximation of the original signature image (η=1), sum of the shearlet subbands in the first scale (η=2 to 5), in the second scale (η=6 to 13), in the third scale (η=14 to 29) and in the fourth scale (η=30 to 61), see Fig. 8.

The images of the sum of shearlet subbands contain useful information about signature images which is important for signature verification system to get the final decision for the query signature. In the presented method, as shown in Fig. 8, for each signature image, we create and save six images in addition to the original signature image and labeled as the original signature image. Therefore, the number of images in the dataset is seven times. Notably, the process of using these images in our method will be explained in Sec. 3.3.1.

In the following of the presented method, four pre-trained models including SigNet and SigNet-F (trained using offline signature datasets), VGG16 and VGG19 (trained using ImageNet dataset) have been used. Transfer learning enables us to use the architecture and parameters of these pre-trained models and act the model as a feature extractor in order to project the input signature images onto the feature space. This is done by performing forward-propagation of the signature images until the last layer, i.e. fully connected layer (FC in Fig. 4) which has the classification role in the source task [55]. In the following and in order to classify signature images in the feature space, two classification methods including k-Nearest Neighbors (k-NN) [2] and Support Vector Machine (SVM) with kernels: linear and polynomial (degree two) have

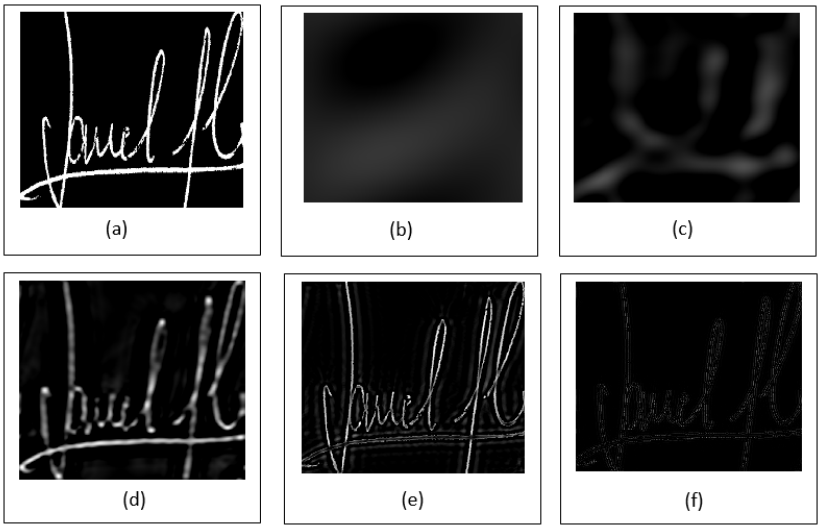

FIGURE 8. (a) A signature image from the MCYT-75 signature dataset, (b) shearlet subband in the lowest frequency band, (c) sum of the shearlet subbands in the first scale corresponding to indices 2 to 5 for η in FFST, (d) sum of the shearlet subbands in the second scale corresponding to indices 6 to 13 for η in FFST, (e) sum of the shearlet subbands in the third scale corresponding to indices 14 to 29 for η in FFST, (f) sum of the shearlet subbands in the fourth scale corresponding to indices 30 to 61 for η in FFST.

been used [10]. These classifications are easy to use and commonly used by researchers [26, 27, 48].

3.3.1. *Experimental Protocol for the Proposed Signature Verification Method.* One of the limitations to be account in real application of a signature verification system is that there are no skilled forgeries for each user enrolled to the system in real situations. Therefore, there are several approaches on how to use the skilled forgery samples available in the dataset for training of the system to empower the system for distinguishing between genuine signatures (positive class) and skilled forgeries (negative class). To the best of our knowledge, three strategies for training of a signature verification system have been used, until today. In the first strategy, training is done using a subset of both genuine signatures and skilled forgery samples of each user [14]. Second strategy tries to train with only a subset of genuine signatures without any forgery samples [23]. In this strategy, training was done using a subset of genuine signatures of the user (as positive class) and a subset of genuine signatures from other users (as random forgeries, negative class) [29, 48].

Finally, the third strategy assumes that the skilled forgery samples are available for only a small subset of users [26].

The first strategy utilizes the information of both genuine signatures and forgery samples available in the dataset for training the system. However, taking into consideration that in real situations such as banking systems, there is no forgery sample for every user enrolled to the system. Second strategy thoroughly considers the real situations for a signature verification system. Based on this strategy, in order to empower the system to better training for distinguishing between genuine and forgery classes, in addition to a subset of genuine signatures as positive class, a subset of signatures from other users are considered as random forgeries (negative class). The third strategy is more acceptable in order to gain the information contained in skilled forgery samples available in the dataset for the system to drive better distinguishing between genuine and skilled forgery classes. This is a reasonable assumption for real applications that there are skilled forgery samples for just a small subset of users [26].

In this work, training has been conducted based on the third strategy, as follows; In the case of UTSig and MCYT-75 datasets, two genuine signatures from every 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 set as testing set. As an example, training process for UTSig dataset, has been shown in Fig. 9. This process has also been conducted for FUM-PHSD dataset. Only difference is that due to the smaller number of users in FUM-PHSD (Table 1), skilled forgery samples of only 5 users considered as negative class in training set and the remaining process was the same. Based on this training process, the training data is balanced and classification does not need different weights for the positive and negative classes.

It should be noted that with each signature image contained in training set or testing set, all of their shearlet subbands (Sec. 3.3) have also been considered in the corresponding training set or testing set. Therefore, the images with the same label are only in the training set or in the testing set and not in both. It should be noted that optimization of the hyperparameters of the classification methods k-NN and SVM were done using the grid search on the training set. Also, in order to avoid the overfitting during the experiments, 3-fold cross validation has been used and the mean of the accuracy has been reported.

3.3.2. *Performance Evaluation of the Proposed Signature Verification Method.* Four well-known criteria including False Rejection Rate (FRR), False Acceptance Rate (FAR), Average Error Rate (AER) and Equal

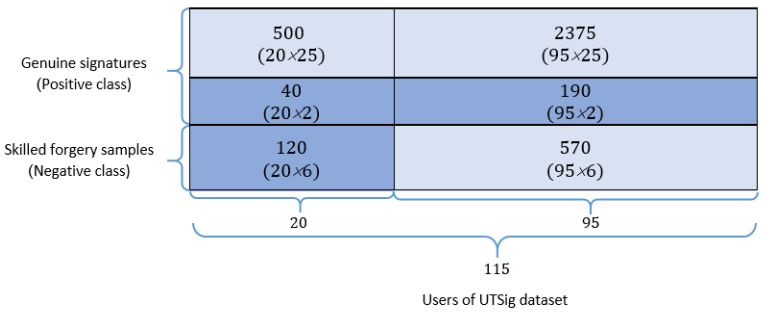

FIGURE 9. Separation of signature images of UTSig dataset into training set and testing set which are specified in dark blue and light blue, respectively. As in Table 1, there are 27 (=25+2) genuine signatures and 6 skilled forgery samples, per user in UTSig dataset.

Error Rate (EER) have been used in order to evaluate the performance of the proposed method. FRR is the fraction of the genuine signatures which have been falsely rejected, FAR is the fraction of skilled forgeries falsely accepted, AER is the average of FRR and FAR and EER is the error rate when FRR=FAR.

The presented signature verification method has been implemented using a system with Intel Core i7-7500U for CPU and 12 GB of RAM. The programming language Python 3.6 has been used. As mentioned in Sec. 3.3, considering shearlet subbands for each signature image in the dataset increases the number of images to seven times, which leads to the time consuming process of transfer learning. Therefore, google colab[27] which provides the free tesla K80 GPU and 13GB of RAM has been used to run the code.

3.3.3. *Signature Verification Results and Discussion.* Several experiments have been conducted using transfer learning on the pre-trained models: SigNet, SigNet-F, VGG16 and VGG19 with k-NN and SVM (linear and RBF kernels) as the classification methods. Also, the experimental results have been obtained in terms of FAR, FRR, AER and EER. In order to make simplified version of the results, the results have been completely shown and compared in the Appendix and just the obtained EERs using SVM (was consistently superior than k-NN) have been shown in Table 2.

[27] `https://colab.research.google.com/notebooks/welcome.ipynb`

TABLE 2. Obtained signature verification results using transfer learning on the pre-trained models.

Pre-trained model	Dataset	SVM (Linear)	SVM (Polynomial)
SigNet	UTSig	EER=7.11	EER=7.38
	FUM-PHSD	EER=3.38	EER=3.38
	MCYT-75	EER=3.00	EER=5.00
SigNet-F	UTSig	EER=5.00	EER=4.96
	FUM-PHSD	EER=4.11	EER=5.03
	MCYT-75	EER=2.98	EER=3.00
VGG16	UTSig	EER=8.41	EER=7.98
	FUM-PHSD	EER=5.06	EER=7.28
	MCYT-75	EER=3.93	EER=5.00
VGG19	UTSig	EER=8.38	EER=5.00
	FUM-PHSD	EER=5.59	EER=5.81
	MCYT-75	EER=4.11	EER=3.43

3.3.4. *Comparison of the Proposed Signature Verification Method with Literature.* In this section we compare the presented signature verification method with some prominent methods in literature. These comparisons have been shown in Table 3 for three datasets: UTSig, FUM-PHSD and MCYT-75.

Table 3 verifies that the proposed signature verification method outperforms the state-of-the-art for UTSig and FUM-PHSD dataset. It worth noting that, as shown in Table 3, the most of the presented methods in the literature are WD and, as mentioned in Sec. 2.3, higher accuracy can be obtained but WD approach has high complexity and cost. However, our presented signature verification method is a WI method. In the case of MCYT-75 signature dataset, the performance of the presented signature verification method is really satisfactory and close to the state-of-the-art reported by Hafemann et al. [26]. However, the presented method outperforms other prominent WI and WD signature verification methods in the literature.

3.4. **Proposed Signature Recognition Method.** In this section, an offline handwritten signature recognition method has been presented. Using the similar procedure of the presented signature verification method, in Sec. 3.3, the presented signature recognition method is based on transfer learning using the pre-trained models: SigNet, SigNet-F, VGG-16 and VGG19. These models have been used as feature extractors to map signature images into the feature space (similar to Sec.3.3). Also, in order to classify the query signature in the new space, k-Nearest Neighbors (k-NN) and Support Vector Machine (SVM) have been used.

TABLE 3. Comparison of the proposed signature verification method with literature on UTSig, FUM-PHSD and MCYT-75 datasets.

Dataset	Signature verification methods	Type	FRR(%)	FAR(%)	AER(%)	EER(%)
UTSig	Soleimani et al. [66]	WI+WD	18.96	16.15	-	17.45
	Soleimani et al. [67]	WD	32.50	32.43	-	32.46
	Soleimani et al. [68]	WD	9.15	27.74	-	15.83
	Narwade et al. [48]	WD	7.41	24.95	16.18	-
	Proposed method (SigNet-F, Polynomial SVM)	WI	0	17.47	8.73	4.96
FUM-PHSD	Sigari et al. [64]	WD	15	15	-	15
	Proposed method (SigNet, Polynomial SVM)	WI	14.29	11.79	13.04	3.38
MCYT-75	Soleimani et al. [66]	WI+WD	14.80	12.44	-	13.44
	Sharif et al. [63]	WD	3.67	6.67	5	-
	Azmi et al. [4]	-	6.67	12.44	9.56	-
	Zois et al. [79]	WD	4.96	17.21	3.45	-
	Ooi et al. [49]	WI	13.16	12.98	-	13.07
	Hafemann et al. [26]	WI	-	-	-	2.87
	Proposed method (SigNet-F, Linear SVM)	WI	4.70	9.46	7.08	2.98

3.4.1. *Experimental Protocol and the Performance Evaluation of the Proposed Signature Recognition Method.* Three datasets including UTSig, FUM-PHSD and MCYT-75 have been considered for experiments (Sec. 3.1). Notably, in the case of signature recognition, only genuine signatures from each datasets have been used. Similar to Sec. 3.3.1, 3-fold cross validation has been used and the mean accuracy has been reported.

It should be noted that the presented signature recognition method has been implemented using system with Intel Core i7-7500U for CPU and 12 GB of RAM. The programming language of Python 3.6 has been used.

3.4.2. *Signature Recognition Results and Discussion.* The performance of the presented signature recognition method has been evaluated using true recognition rate (the percentage of truly classified signatures) on the datasets. Table 4 shows the obtained signature recognition results using transfer learning on the pre-trained models: SigNet, SigNet-F, VGG16 and VGG19. In the following of this section, the performance of the presented signature recognition method based on the shearlet transform and using each of the pre-trained models have been compared and discussed. As shown in Table 4, VGG19, VGG16, SigNet and SigNet-F had the best performance for signature recognition, respectively. These performances have been compared using the following remarks.

- The pre-trained model SigNet had a little better performance than SigNet-F (Table 4): A possible explanation for this is as

TABLE 4. Signature recognition results using transfer learning on the pre-trained models in terms of true recognition rate (%).

Pre-trained Model	Dataset	k-NN (k=3)	SVM (Linear)	SVM (Polynomial)
SigNet	UTSig	72.52	84.50	68.03
	FUM-PHSD	88.75	92.68	86.96
	MCYT-75	61.27	79.94	62.79
SigNet-F	UTSig	72.79	82.80	75.51
	FUM-PHSD	88.57	94.64	85.54
	MCYT-75	61.59	77.59	78.86
VGG16	UTSig	94.22	94.71	95.12
	FUM-PHSD	98.93	99.29	99.64
	MCYT-75	91.17	89.78	91.56
VGG19	UTSig	93.65	94.33	95.24
	FUM-PHSD	98.57	99.64	99.64
	MCYT-75	90.73	90.29	91.56

follows; According to the same architecture used in the structure of SigNet and SigNet-F (refer to [26]), difference in their performance is due to their different ways of training which used a subset of forgery samples in addition to genuine signatures for SigNet-F. So, the parameters of SigNet-F have been better tuned for separation of genuine signatures from forgery samples. This is not necessary for the task of signature recognitio, i.e. separation between classes of different users and led to a little mistake in making decisions about different classes of users, as shown in the obtained experimental results for SigNet and SigNet-F.

- The pre-trained model VGG19 had better performance than VGG16 (Table 4): As discussed in Sec. 2.5, the pre-trained model VGG19 is a deeper CNN model than VGG16 [65]. Therefore VGG19 has a higher capacity for the classification tasks. This is truly shown in the obtained results for VGG16 and VGG19.
- The pre-trained model VGG19 had better performance than SigNet (Table 4): There are two facts for these results: at first it should be noted that the presented method is based on the shearlet subbands and the final recognition process has been conducted on the shearlet subband images. Some of these images, as shown in Fig. 8, are less similar to the original signature images (Fig. 7(b),(c) and (f)). On the other hand, the task of a signature recognition system is separation between classes of different users, which has been truly learned by the CNN model VGG19 pre-trained on ImageNet as a large dataset of images

from different categories. These are led to better performance for VGG19 than SigNet in the conducted experiments.

Therefore, the best performance of the presented signature recognition method has been gained using transfer learning on the pre-trained model VGG19 based on the shearlet subbands.

3.4.3. *Comparison of the Proposed Signature Recognition Method with Literature.* Several methods have been proposed for handwritten signature recognition, until today. However, none of them used CNN in their works and also these works have been evaluated using the private (rather than public) signature datasets and since we did not have access to these datasets, the comparison is not fair. To the best of our knowledge, just three proposed signature recognition methods (Table 5) have been implemented on the FUM-PHSD and MCYT-75 datasets used in this paper and no signature recognition method has been tested on the UTSig dataset, until today.

The comparison of the presented signature recognition method with similar works has been shown in Table 5. We see that our method performs quite well on the UTSig dataset. For FUM-PHSD dataset, true recognition rate has insignificant difference with complete precision (99.64%). Similar to the performance of the presented signature verification method on MCYT-75 (Sec. 3.3.4), the performance of the presented signature recognition method has a little difference with the state-of-the-art.

TABLE 5. Comparison of the proposed signature recognition method with literature on three datasets.

Method	UTSig	FUM-PHSD	MCYT-75
Fakhlai et al. [19]	-	98	-
Sigari et al. [53]	-	100	-
Hezil et al. [31]	-	-	97.3
Proposed method	95.24 (VGG19, Polynomial SVM)	99.64 (VGG16, Polynomial SVM) (VGG19, Linear SVM) (VGG19, Polynomial SVM)	91.56 (VGG16, Polynomial SVM) (VGG19, Polynomial SVM)

4. Conclusions and Future Work

In this paper, we have presented a methodology for offline handwritten signature verification and recognition based on the shearlet transform and transfer learning using the pre-trained models. Here, shearlet subbands containing useful information, about the important details of signature images including edges and corners, have been used to compute some images corresponding to each original signature image available in

the dataset. These images have been created using sum of the shearlet subbands. Experiments were conducted on UTSig (a large public offline Persian dataset), FUM-PHSD (another Persian offline signature dataset) and MCYT-75 (as a benchmark Latin offline signature dataset). The obtained experimental results for both signature verification and signature recognition showed the effectiveness of the presented methods in comparison with literature. Our proposed methods were also compared favorably with the state-of-the-art.

Although compared with literature, the presented method is really satisfactory, the performance of the pre-trained models will be tested on the datasets without considering shearlet subbands and compared the results with this work. Also, the presented method based on shearlet subbands will be checked using other Persian or Persian like (such as Arabic script) and Latin signature datasets, in the future. We shall investigate other ways for using the shearlet coefficients and other sparse transformations for handwritten signature verification and signature recognition.

Acknowledgment. The authors would like to thank the reviewers for their valuable suggestions and comments during the revision process.

References

1. I. Abroug and N.E. Ben Amara, *Off-line signature verification systems: recent advances*, Int. Conf. Image Process. Appl. Syst., (2014), pp. 1-6.
2. N.S. Altman, *An introduction to kernel and nearest-neighbor nonparametric regression*, Amer. Statist., 46 (1992), pp. 175-185.
3. M. Arathi and A. Govardhan, *An efficient offline signature verification system*, Int. J. Mach. Learn. Comput., 4 (2014), pp. 533-537.
4. A.N. Azmi, D. Nasien and F.S. Omar, *Biometric signature verification system based on freeman chain code and k-nearest neighbor*, Multimedia Tools Appl., 76 (2017), pp. 15341-15355.
5. Y. Bengio, *Learning deep architectures for AI*, Found. Trends Mach. Learn., 2 (2009), pp. 1-127.
6. Y. Bengio and A. Courville, *Deep learning of representations*, in Handbook on Neural Inf. Process., Springer, (2013), pp. 1-28.
7. T.A. Bubba, G. Kutyniok, M. Lassas, M. Marz, W. Samek, S. Siltanen and V. Srinivasan, *Learning the invisible: a hybrid deep learning-shearlet framework for limited angle computed tomography*, arXiv:1811.04602v1, 2018.
8. E.J. Candes and D.L. Donoho, *Continuous curvelet transform: I. Resolution of the wavefront set*, Appl. Comput. Harmon. Anal., 19 (2005), pp. 162-197.

9. E.J. Candes and D.L. Donoho, *Curvelets: a surprisingly effective nonadaptive representation for objects with edges*, Tech. Rep. Department of Statistics, Stanford University, 1999.
10. C. Cortes and V. Vapnik, *Support-vector networks*, Mach. Learn., 20 (1995), pp. 273-297.
11. B. Cozzens and R. Huang, *Signature verification using a convolutional neural network*, Las Vegas, Nevada, 2017.
12. I. Daubechies, *Ten lectures on wavelets*, Soc. Ind. Appl. Math., Philadelphia, 1992.
13. L. Deng and D. Yu, *Deep learning: methods and applications*, Found. Trends in Signal Process., 7 (2014), pp. 197-387.
14. S. Dey, A. Dutta, J. I. Toledo, S K. Ghosh, J. Llados and U. Pal, *SigNet: convolutional siamese network for writer independent offline signature verification*, CORR, abs/1707.02131, 2017.
15. M.N. Do and M. Vetterli, *The contourlet transform: an efficient directional multiresolution image representation*, IEEE Trans. Image Process., 14 (2005), pp. 2091-2106.
16. M.A. Duval-Poo, F. Odone and E. De Vito, *Edges and corners with shearlets*, IEEE Trans. Image Process., 24 (2015), pp. 3768-3780.
17. G. Easley, D. Labate and Wang-Q Lim, *Sparse directional image representations using the discrete shearlet transform*, J. Appl. Comput. Harmon. Anal., 25 (2008), pp. 25-46.
18. G. Eskander, R. Sabourin and E. Granger, *Hybrid writer-independent-writer-dependent ofine signature verication system*, IET Biometrics 2(4) (2013), pp. 169-181.
19. M. Fakhlai, H.R. Pourreza, R. Moarefdost and S. Shadroo, *Offline signature recognition based on contourlet transform*, Int. Conf. Mach. Learn. Comput., 2009.
20. 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 Biom. Authentication, (2004), pp. 295-306.
21. A. Foroozandeh, Y. Akbari, M.J. Jalili and J. Sadri, *A novel and practical system for verifying signatures on Persian handwritten bank checks*, Int. J. Pattern Recognit. Artif. Intell. (IJPRAI), 26 (2012), pp. 1-27.
22. C. Freitas et al., *Bases de dados de cheques bancarios brasileiros*, In XXVI Conferencia Latinoamericana de Informatica, 2000.
23. Y. Guerbai, Y. Chibani and B. Hadjadji, *The effective use of the one-class SVM classifier for handwritten signature verification based on writer-independent parameters*, PR, 48(1) (2015), pp. 103-113.

24. Y. Guo, U. Budak, A. Sengur and F. Smarandache, *A retinal vessel detection approach based on shearlet transform and indeterminacy filtering on fundus images*, J. Symmetry, 9 (2017).
25. K. Guo and D. Labate, *The construction of smooth parseval frames of shearlets*, Math. Model. Nat. Phenom., 8 (2013), pp. 82-105.
26. L.G. Hafemann, R. Sabourin and L.S. Oliveira, *Learning features for offline handwritten signature verification using deep convolutional neural networks*, Pattern Recognit., 70 (2017), pp. 163-176.
27. L.G. Hafemann, R. Sabourin and L.S. Oliveira, *Offline handwritten signature verification-literature review*, Int. Conf. Image Process. Theory Tool Appl., (2017), pp. 1-8.
28. 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), (2016), pp. 2989-2994.
29. L.G. Hafemann, R. Sabourin and L.S. Oliveira, *Writer-independent feature learning for Offline Signature Verification using Deep Convolutional Neural Networks*, Int. Joint Conf. Neural Netw., (2016), pp. 2576-2583.
30. S. Häuser and G. Steidl, *Fast Finite Shearlet Transform: a tutorial*, arXiv:1202.1773v1 [math.NA] 8 Feb 2012.
31. H. Hezil, R. Djemili and H. Bourouba, *Signature recognition using binary features and KNN*, Int. J. Biometrics, 10 (2018), pp. 1-15.
32. M. Kaboli, *A Review of transfer learning algorithms*, Research Report, Technische Universität München, 2017.
33. M.K. Kalera, S. Srihari and A. Xu, *Offline signature verification and identification using distance statistics*, Int. J. Pattern Recognit. AI., 18 (2004), pp. 1339-1360.
34. G. Kanghui, G. Kutyniok and D. Labate, *Sparse multidimensional representations using anisotropic dilation and shear operators*, Int. Conf. Interact. Wavelets and Splines, (2005), pp. 189-201.
35. H. Khalajzadeh, M. Mansouri and M. Teshnehlab, *Persian signature verification using convolutional neural networks*, Int. J. Eng. Res. Technol. 1 (2012), pp. 7-12.
36. M. Khan, A. Jamil, M. Irfan, R. Seungmin and B. Sung, *Convolutional neural networks based fire detection in surveillance videos*, IEEE Access, 6 (2018), pp. 18174-18183.
37. N. Kingsbury, *Complex wavelets for shift invariant analysis and filtering of signals*, Appl. Comput. Harmon. Anal., 10 (2001), pp. 234-253.
38. N. Kingsbury, *Image processing with complex wavelets*, Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci., 357 (1999), pp. 2543-2560.

39. J.R. Koza, F.H. Bennett, D. Andre and M.A. Keane, *Automated design of both the topology and sizing of analog electrical circuits using genetic programming*, J. Artificial Intelligence Des., (1996), pp. 151-170.
40. A. Krizhevsky, I. Sutskever and G.E. Hinton, *ImageNet classication with deep convolutional neural networks*, Adv. Neural Inf. Process. Syst., 25 (2012), pp. 1097-1105.
41. R. Kumar, L. Kundu, B. Chanda and J.D. Sharma, *A Writer-independent off-line signature verication system based on signature morphology*, Proc. Conf. Intell. Interactive Technol. Multimedia, New York, NY, USA, (2010), pp. 261-265.
42. G. Kutyniok and L. Demetrio, *Shearlets: multiscale analysis for multivariate data*, Springer New York: Birkhäuser, 2012.
43. G. Kutyniok, W.Q. Lim and R. Reisenhofer, *Shearlab 3D: faithful digital shearlet transforms based on compactly supported shearlets*, J. ACM Trans. Math. Softw., 42 (2016), pp. 1-42.
44. G. Kutyniok, M. Shahram and X. Zhuang, *ShearLab: a rational design of a digital parabolic scaling algorithm*, SIAM J. Imaging Sci., 5 (2012), pp. 1291-1332.
45. S. Mallat and S. Zhong, *Characterization of signals from multiscale edges*, IEEE Trans. Pattern Anal. Mach. Intell., 14 (1992), pp. 710-732.
46. S. Min, B. Lee and S. Yoon, *Deep learning in bioinformatics*, Briefings in Bioinformatics, 18 (2016), pp. 851-869.
47. R.A. Mohammed, R.M. Nabi, S.M.R. Mahmood and R.M. Nabi, *State-of-the-art in handwritten signature verification system*, Int. Conf. Comput. Sci. Comput. Intell., (2015), pp. 519-525.
48. P.N. Narwade, R.R. Sawant and S.V. Bonde, *Offline handwritten signature verification using cylindrical shape context*, 3D Res. 9 (2018), https://doi.org/10.1007/s13319-018-0200-0.
49. S.Y. Ooi, A.B. J. Teoh, Y.H. Pang and B.Y. Hiew, *Image-based handwritten signature verication using hybrid methods of discrete radon transform, principal component analysis and probabilistic neural network*, Appl. Soft Comput., 40 (2016), pp. 274-282.
50. N. Otsu, *A threshold selection method from gray-level histograms*, IEEE Trans. Sys. Man. Cyber., 9 (1979), pp. 62-66.
51. S.J. Pan and Q. Yang, *A survey on transfer learning*, IEEE Trans. Knowledge Data Eng., 22 (2010), pp. 1345-1359.
52. D.N. Perkins and G. Salomon, *Transfer of learning*, Int. Encyclopedia of Education, 2nd Edition, Oxford, England: Pergamon Press, 1992.

53. 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., (2009), pp. 670-673.
54. L.Y. Pratt, *Discriminability-based transfer between neural networks*, Advances in Neural Inf. Process. Syst. 5 (1993), pp. 204-211.
55. S. Rajaraman et al., *Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images*, (2018), PeerJ 6:e4568; DOI: 10.7717/peerj.4568.
56. W. Rawat and Z. Wang, *Deep convolutional neural networks for image classification: a comprehensive review*, Neural Comput., 29 (2017), pp. 2352-2449.
57. S. Razavian, H. Azizpour, J. Sullivan and S. Carlsson, *CNN features off-the-shelf: an astounding baseline for recognition*, IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), (2014), pp. 512-519.
58. J.D. Regele and O.V. Vasilyev, *An adaptive wavelet-collocation method for shock computations*, Int. J. Comput. Fluid Dyn., 23 (2009), pp. 503-518.
59. J. Sadri, M.J. Jalili, Y. Akbari and A. Foroozandeh, *Designing a new standard structure for improving automatic processing of Persian handwritten bank cheques*, Int. J. Pattern Anal. Appl. (PAA), 17 (2014), pp .849-862.
60. M.H. Saffar, M. Fayyaz, M. Sabokrou and M. Fathy, *Online signature verification using deep representation: a new descriptor*, Int. Comput. Vis. Pattern Recognit. (cs.CV), arXiv:1806.09986.
61. L. Samuel, *Some studies in machine learning using the game of checkers*, IBM J. Res. Dev., 3 (1959), pp. 210-229.
62. J. Schmidhuber, *Deep learning in neural networks: an overview*, Neural Networks, 61 (2015), pp. 85-117.
63. M. Sharif, M.A. Khan, M. Faisal, M. Yasmin and S.L. Fernandes, *A framework for offline signature verification system: best features selection approach*, Pattern Recognit. Lett., 2018.
64. M.H. Sigari, M.R. Pourshahabi and H. R. Pourreza, *Offline handwritten signature identification and verification using multiresolution Gabor wavelet*, Int. J. Biometrics and Bioinform. (IJBB), 5 (2011), pp. 234-248.
65. K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, arXiv 1409.1556, 2014.
66. A. Soleimani, B.N. Araabi and K. Fouladi, *Deep multitask metric learning for offline signature verification*, Pattern Recognit. Lett.,

80 (2016), pp. 84-90.
67. A. Soleimani, K. Fouladi and B.N. Araabi, *UTSig: a Persian offline signature database*, IET Biometrics, 6 (2017), pp. 1-8.
68. A. Soleimani, K. Fouladi and B.N. Araabi, *Persian offline signature verification based on curvature and gradient histograms*, 6th Int. Conf. Comput. Knowledge Eng., (2016), pp. 147-152.
69. H. Srinivasan, S.N. Srihari and M.J. Beal, *Machine learning for signature verification*, Computer Vision, Graphics and Image Processing, Springer Berlin Heidelberg, (2006), pp. 761-775.
70. J.L. Starck, F. Murtagh and J. Fadili, *Sparse image and signal processing: wavelets, curvelets, morphological diversity*, Cambridge University Press, 1st Edition, 2010.
71. S. Tayeb, M. Pirouz, B. Cozzens, R. Huang, M. Jay, K.Khembunjong and S. Paliskara, *Toward data quality analytics in signature verification using a convolutional neural network*, IEEE Int. Conf. Big Data, (2017), pp. 2644-2651.
72. J. Vargas, M. Ferrer, C. Travieso and J. Alonso, *Off-line handwritten signature GPDS-960 corpus*, Doc. Anal. Recognit., 9th Int. Conf., 2 (2007), pp. 764-768.
73. K. Weiss, T.M. Khoshgoftaar and D.D. Wang, *A survey of transfer learning*, J. Big Data, 3 (2016).
74. N. Xiaopeng, W. Zhiliang and P. Zhigeng, *Extreme learning machine based deep model for human activity recognition with wearable sensors*, Comput. Sci. Eng., 21 (2018), pp. 16-25.
75. M.E. Yahyatabar and J. Ghasemi, *Online signature verification using double-stage feature extraction modelled by dynamic feature stability experiment*, IET Biometrics, 6 (2017), pp. 393-401.
76. F.Yuan, L-M. Po, M. Liu, X, Xu, W. Jian, K. Wong and K. Cheung, *Shearlet based video fingerprint for content-based copy detection*, J. Signal Inf. Process., 7 (2016), pp. 84-97.
77. G. Zaccone, M.R. Karim and Menshawy, *Deep learning with TensorFlow, explore neural networks and build intelligent systems with Python*, Birmingham, England, Mumbai, India, Packt, 2017.
78. G. Zhong, L. Wang and J. Dong, *An overview on data representation learning: from traditional feature learning to recent deep learning* , J. Financ Data Sci., 2 (2016), pp. 265-278.
79. E.N. Zois, L. Alewijnse and G. Economou, *Offline signature verification and quality characterization using poset-oriented grid features*, Pattern Recognit., 54 (2016), pp. 162-177.

Appendix

Tables 6-9, show the signature verification results using transfer learning on the pre-trained models: SigNet, SigNet-F, VGG16 and VGG19, respectively. Also, Table 10, shows the best obtained EER. As shown in this table, the pre-trained models SigNet-F and SigNet had the best performance on all three datasets. In the following, the performance of four pre-trained models has been discussed, based on Tables 6-10.

Table 6. Obtained signature verification results using transfer learning on the pre-trained model SigNet.

Verification results using SigNet (%)	k-NN (k=3)	SVM (Linear)	SVM (Polynomial)
UTSig	FRR=4.02 FAR=10.70 AER=7.36 EER=9.53	FRR=6.40 FAR=13.48 AER=9.94 EER=7.11	FRR=0 FAR=17.471 AER=8.73 EER=7.38
FUM-PHSD	FRR=5.48 FAR=14.17 AER=9.82 EER=6.49	FRR=14.29 FAR=11.79 AER=13.04 EER=3.38	FRR=0.6 FAR=26.07 AER=26.67 EER=3.38
MCYT-75	FRR=12.09 FAR=18.29 AER=15.19 EER=4.96	FRR=20.82 FAR=19.56 AER=20.19 EER=3.00	FRR=18.25 FAR=18.63 AER=18.44 EER=5.00

Table 7. Obtained signature verification results using transfer learning on the pre-trained model SigNet-F.

Verification results using SigNet-F (%)	k-NN (k=3)	SVM (Linear)	SVM (Polynomial)
UTSig	FRR=4.20 FAR=10.95 AER=7.58 EER=9.33	FRR=7.02 FAR=14.04 AER=10.53 EER=5.00	FRR=0 FAR=17.47 AER=8.73 EER=4.96
FUM-PHSD	FRR=5.00 FAR=13.33 AER=9.17 EER=7.06	FRR=13.57 FAR=13.81 AER=13.69 EER=4.11	FRR=1.19 FAR=24.64 AER=12.92 EER=5.03
MCYT-75	FRR=12.19 FAR=18.60 AER=15.40 EER=4.91	FRR=4.70 FAR=9.46 AER=7.08 EER=2.98	FRR=12.59 FAR=17.97 AER=15.28 EER=3.00

- The pre-trained model SigNet-F had better performance than SigNet (Tables A and B): As mentioned in Sec. 2.5, two pre-trained models SigNet and SigNet-F have the same architecture and their difference is in the training process which has

TABLE 8. Obtained signature verification results using transfer learning on the pre-trained model VGG16.

Verification results using VGG16 (%)	k-NN (k=3)	SVM (Linear)	SVM (Polynomial)
UTSig	FRR=1.58 FAR=3.69 AER=2.63 EER=16.11	FRR=5.53 FAR=6.74 AER=6.14 EER=8.41	FRR=0.37 FAR=7.70 AER=4.03 EER=7.98
FUM-PHSD	FRR=7.73 FAR=1.07 AER=4.41 EER=8.62	FRR=6.19 FAR=8.09 AER=7.14 EER=5.06	FRR=0.71 FAR=9.76 AER=5.23 EER=7.28
MCYT-75	FRR=3.40 FAR=8.06 AER=5.73 EER=7.70	FRR=13.14 FAR=13.52 AER=13.33 EER=3.93	FRR=9.43 FAR=12.44 AER=10.94 EER=5.00

TABLE 9. Obtained signature verification results using transfer learning on the pre-trained model VGG19.

Verification results using VGG19 (%)	k-NN (k=3)	SVM (Linear)	SVM (Polynomial)
UTSig	FRR=1.42 FAR=3.83 AER=2.63 EER=13.08	FRR=5.75 FAR=7.51 AER=6.63 EER=8.38	FRR=0.46 FAR=6.95 AER=3.71 EER=5.00
FUM-PHSD	FRR=1.43 FAR=6.90 AER=4.16 EER=9.23	FRR=6.30 FAR=7.50 AER=6.90 EER=5.59	FRR=1.31 FAR=9.64 AER=5.48 EER=5.81
MCYT-75	FRR=3.33 FAR=8.09 AER=5.71 EER=7.64	FRR=10.67 FAR=13.56 AER=12.11 EER=4.11	FRR=8.73 FAR=10.89 AER=9.81 EER=3.43

TABLE 10. The best obtained signature verification results in terms of EER using transfer learning on the pre-trained models.

The best obtained EER (%)	SigNet	SigNet-F	VGG16	VGG19
UTSig	7.11 (Linear SVM)	4.96 (Polynomial SVM)	7.98 (Polynomial SVM)	5.00 (Polynomial SVM)
FUM-PHSD	3.38 (Linear SVM)	4.11 (Linear SVM)	5.06 (Linear SVM)	5.59 (Linear SVM)
MCYT-75	3.00 (Linear SVM)	2.98 (Linear SVM)	3.93 (Linear SVM)	3.43 (Polynomial SVM)

been conducted using only genuine signatures (for SigNet) or genuine signatures in addition to a subset of skilled forgery samples (for SigNet-F) [26]. Therefore, as shown in the first

two columns of Table E, the ability of SigNet-F for distinguishing between the positive (genuine signatures) and negative classes (forgery samples) is higher than SigNet which is due to the better tuning of the parameters of SigNet-F.

- The pre-trained model VGG19 had better performance than VGG16 (Tables C and D): As mentioned in Sec. 2.5, two pre-trained models VGG16 and VGG19 have similar architecture and their difference is in more convolutional layers used in VGG19 [65]. In other words, VGG19 is a deeper CNN model than VGG16. This architecture enables VGG19 to have better performance in image classification [65]. This is truly shown in our experiments as shown in two last columns of Table E.
- The pre-trained model SigNet-F had better performance than VGG19 (Tables B and D): Taking into consideration the pre-trained model SigNet-F has been trained using the offline handwritten signature datasets, however, VGG19 has been trained on ImageNet dataset, as a general large dataset, we expect the better performance of SigNet-F for signature verification than VGG19 which is shown using the obtained experimental results (see the second and the fourth columns of Table B).

As above discussion, the best performance of the presented signature verification method has been obtained using transfer learning on the pre-trained model SigNet-F based on the shearlet subbands.

---

[1]Department of Applied Mathematics, Faculty of Sciences and Modern Technology, Graduate University of Advanced Technology, Kerman, Iran.

E-mail address: at_foroozandeh@yahoo.com

[2] Department of Applied Mathematics, Faculty of Mathematics and Computer, Shahid Bahonar University of Kerman, Kerman, Iran and Mahani Mathematical Research Center, Shahid Bahonar University of Kerman, Kerman, Iran.

E-mail address: askari@uk.ac.ir

[3] Department of Biomedical Engineering, School of Advanced Technologies in Medicine, Isfahan University of Medical Sciences, Isfahan, Iran and Medical Image and Signal Processing Research Center, Isfahan University of Medical Sciences, Isfahan, Iran.

E-mail address: h_rabbani@med.mui.ac.ir