PERSIVAL: Persian Signature Verification using Deep Learning Siamese Network

Yalda Afshar, Jamal Ghasemi

Department of Engineering and Technology, University of Mazandaran
yalda.afshar@gmail.com

***Abstract*—Signature verification is widely used for identity verification. Each person has a unique signature in terms of the shape and the amount of pressure they put on the pen which results in parts of the signature to be darker, lighter, or even create disjointed parts. Many researchers have developed machine learning techniques to automatically verify the genuinity of the signatures from scanned documents, i.e., offline signature verification. However, these techniques are usually trained using English signatures and therefore are not suited for Persian signatures. Persian signatures are different from other nation signatures, since people usually do not use text in it and they draw a shape as their signature. Even if they use text in their signature, it is in Persian alphabets, which is very different than English characters. In this paper, we have focused on Persian signatures and have proposed PERSIVAL. PERSIVAL is designed as a Siamese network and uses deep Convolutional neural networks (CNNs). PERSIVAL is trained on a set of Persion signatures consists of genuine and forgery signatures from a general population. Through the training process, PERSIVAL learns the features of genuine and forgery signatures that are independent from the users, i.e., writers. This is known as writer-independent signature verification which is generally a harder problem than writer-dependant solutions. PERSIVAL has an accuracy of $90\%$for offline writer-independent signature verification and outperforms the state-of-the-art SVM-based approach for Persian signature verification technique by 19%.**

***Index terms*— Deep Learning, Convolutional Neural Network (CNN), Siamese Network, Signature Verification**

I. Introduction

Although we are in the age of information technology and the wide use of electronic devices over paper, the vast majority of the financial transactions and legal contracts are finalized through handwritten signatures. As such, many of the fraudulent transactions or contracts are done through signature forgeries. Although, if implemented correctly, handwritten signatures can act as an authentication factor in Multi Factor Authentication (MFA). Therefore, it is paramount that we study and utilized novel solutions for detecting and preventing signature forgery detection.

To achieve the goal of detecting signature forgeries, we employ Machine Learning (ML) techniques as they have a great capacity for detecting common patterns of a genuine signature and through that, identify the fraudulent signatures. One of the biggest challenges of a Machine Learning-based solution is providing an adequate dataset, since the signatures' shape has a great impact on the system's efficiency.

There are two ways to investigate the signature verification problem: 1) manual which means comparing two signatures by looking at them. This way has been proven to be an imprecise method and skilled forgeries or even some simple ones cannot be detected. 2) automated: this method depends on the algorithms and systems we use since there is no human intervention involved. ML-based techniques can learn a pattern and when exposed to a new dataset, they can give an accurate result. Moreover, it sometimes surpasses human's ability in object recognition.

One can study the problem of signature verification in two settings.

- Writer Dependent (WD) setting: In the WD setting, the idea is to train the system with a sample signature of all the users in the system such that the system can learn the individual traits of each writer.
- Writer Independent (WI) setting: In the WI setting, the system is trained with a smaller subset of signatures with the goal of identifying the general characteristics of genuine vs. forgery signatures. WI is a more powerful setting since it can be used to verify the identity of users that the system has not seen during the training.

Moreover, this problem can be broken into two categories based on the type of input data.

- Online signatures: were the user signs on an electronic device. Such signatures can record a wealth of information about the signature such as speed and pressure, in addition to the visual aspects [1].
- Offline signatures: were the user draws on a piece of paper and the signature is then scanned and entered to the system. Such signatures have far less information compared to their online counterparts. This is also referred to as static signatures.

In this paper, we focus on offline WI approach of detecting signature forgeries. The best previous work on static Persian signature verification, followed a WD approach and achieved an accuracy of 71% using custom feature selection [2]. The best previous work on English, Hindi, and Bengali signature verification, followed a WI approach achieved high accuracy (*e.g.*, 100% on CEDAR dataset [3], 86% on Bengali dataset, and 84.64% on Hindi dataset [4]).

In this paper we combine the two works and present PERSIVAL, **PER**sian **SI**gnature **V**erific**A**tion using Deep

Learning Siamese Network, for a static WI approach for Persian signatures with 90%accuracy.

In the rest of the paper, we first describe our testing methodology in section III. Next, we describe our experiment setup in section IV. Finally, sections V and VI, we present our findings and draw a conclusion.

II. Related work

Researches in signature verification field have shown that signatures' shape which has a great contribution to the system's efficiency is different depends on the culture. Comparing branch-points and end-points of Persian signatures of UTSig dataset with Dutch, Spanish, Chinese and Japanese ones [2] (UTSig) determines that Persian signatures have fewer branch-points and endpoints than the others. This emphasize on the importance of choosing practical features that represent images.

In recent works, texture features are more popular, for instance Local Binary Patterns (LBP) [5], [6] and Gray-Level Co-occurrence Matrix (GLCM) [7] and directional-based features such as Histogram of Oriented Gradients (HOG) [5]. However, no feature has been proven to have a great impact on signature verification and researchers usually use a variety of features in these researches. Such feature extractions are necessary for classic classification algorithms such as SVM in which feature extraction phase and finding good representations for images is a time consuming task.

In contrast, deep learning algorithms do not require such a phase and can identify the features on their own. Therefore, the time consuming feature extraction phase can be omitted. In [8], Hafemann, *et al*., proposed a new technique for offline handwritten signature verification using convolutional neural network (CNN). They proposed a two phase algorithm. In the first phase, they trained a CNN to learn the characterizations of a forgery signature in the Writer Independent setting. In the second phase, they trained an SVM for each of the users in Writer Dependent setting. They achieved an accuracy of 98.26%.

In SigNet, [4], Dey *et al*., used Siamese network which is an end to end training method on English and Hindi datasets. Their proposed network were tested on various benchmark datasets such as CEDAR [3], GPDS [9], in addition to their Hindi and Bengali datasets and beat the best previous accuracies. In comparison to [8], SigNet is in WI setting and requires less training data due to the use of one-shot learning.

We use SigNet's Siamese network in this paper. A Siamese network [10] is a neural net that is composed of two subnets sharing the same weights. Siamese networks receive two inputs and computes the output vectors by passing them to the two subnets. Next, it computes the distance between the two output vectors as a measure of how similar or dissimilar they are together. Therefore, in the this work, the network is fed with two samples of genuine signatures as a positive sample and a sample of genuine and a sample of forgery signature as a negative sample. The network uses a contrastive loss function [11]. This function rewards the network when the distance of the output vectors of the two subnets is close to 0 for positive samples and when the distance of the subnets is close to 1 for negative samples.

III. Methodology

In this paper, we use a refined version of UTSig dataset, which is a rich dataset of educated male signatures. This dataset contains 115 users who are 90% right handed, for each user there is 27 genuine samples and 42 skilled forgeries in the dataset. There are also 6 opposite-hand signatures which we did not use. Moreover, the forgers were 40% female.

We started by preprocessing the images by removing the background, bringing each pixel value to the [0,1] range, inverting the image such that the background has a value of 0, and placing them all on a 155x220 canvas. Next, we manually went through the signatures and filtered the outliers (the users whose genuine signatures were inconsistent). See Figure 1 for an example of such signatures. At the end, from 115 users, we were left with 70 users. We split those 70 users into 3 categories, training (70%), validation (15%), and testing (15%) users. We emphasise that even in this reduced state, the UTSig datasets still has more signature samples than one of the famous benchmark datasets, CEDAR, which has the signatures of 55 users.

To prepare the input of the the Siamese network, we need to create *pairs* of positive and negative samples. Therefore, we need to choose pairs of genuine signatures (`GEN-GEN`) and feed it to the network as positive samples and pairs of genuine-forgery (`GEN-FRG`) samples and feed them to the network as negative samples.

To create `GEN-GEN` pairs, for each of the 70 users, we select a combination of 2 out of 27 to get a total of 70 * 351 `GEN-GEN` pairs. We matched all genuine signatures of a user with all 42 forgery signatures of the same user to get a total of 70 * 27 * 42 = 70 * 1,134 `GEN-FRG` pairs. To create a balanced dataset, we randomly selected 350 positive samples and 350 negative samples for each user.

During the training phase, we used the pairs of the users that were chosen for training and validation. Similarly, during the test phase, we used the pairs of the users that were chosen for testing.

We ran the experiments 10 times and took the average accuracy. Moreover, in each iteration of experiment we shuffled the training, validation, and test users and for each user we shuffled the selected pairs.

We used the network of SigNet and adjusted the Dropout rates to prevent overfitting. Moreover, in SigNet's implementation of the Siamese network, the authors search for the best threshold value that would minimize the contrastive loss of test inputs after the prediction phase. In contrast, we chose the fixed threshold value of 0.5 for both training and test samples.

The base network and its parameters can be found in listing 1. More detailed explanations can be found in [4].

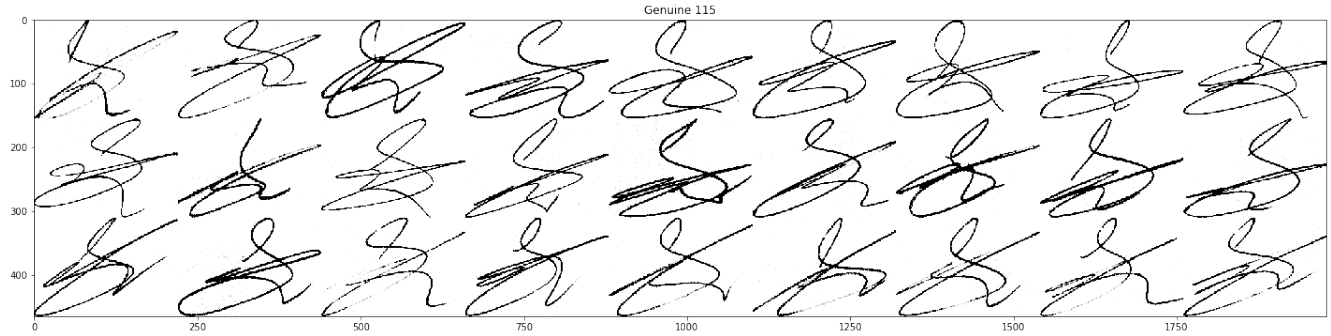

Fig. 1: "Genuine signatures of user #115 (an example of a filtered user)"

IV. Experiment setup

We used Jupyter notebook on Google Colab with GPU processing enabled. The source code is available upon request. We experimented with various parameters to prevent over-fitting of the network and tracked both the overall accuracy of the network and the accuracy of false positives (*i.e.*identifying a `GEN-FRG` as a `GEN-GEN`) as we believe this to be the most important metric when it comes to the security of the solution.

V. Results

Table I shows the overall accuracy of the model on test data. Each row corresponds to a test user. Each row contains the counts of True Positive (TP), True Negative (TN), False Positive (FP), and False Negatives (FN) predictions of the network for that user. Moreover, the False Acceptance Rate (FAR) and False Rejection Rates (FRR) are calculated for each user. Overall, the table shows we have an accuracy of 100% for the 8 of the test users and a total accuracy of 90%.

User#	TP	TN	FP	FN	FAR	FRR
2	276	276	0	0	0.00	0.00
5	276	276	0	0	0.00	0.00
9	276	276	0	0	0.00	0.00
15	127	242	33	150	0.12	0.54
26	276	276	0	0	0.00	0.00
27	276	276	0	0	0.00	0.00
33	276	276	0	0	0.00	0.00
41	276	276	0	0	0.00	0.00
50	276	276	0	0	0.00	0.00

TABLE I: Accuracy results for test users

On the other hand, running the model on the users that we had initially filtered out, shows that the model tends to detect the vast majority of the pairs as negative samples (high FRR). This strengthens our initial argument that the genuine signatures of these sets of users were of low quality. Refer to table II for the reported accuracy of some of these users.

It is worth mentioning that previous work on Persian signatures using the same dataset have achieved an accuracy of 71% without filtering any of the signatures. We believe that this is due the fact that our approach is a Writer Independent approach, while the work of [2] is a Writer Dependent approach. Therefore, in their work, they were able to capture the essence of the signature of all the users. In contrast, in our solution, the network is focused on learning what makes a signature genuine regardless of the user. This difference, could explain the difference in the filtering of the dataset.

User#	TP	TN	FP	FN	FAR	FRR
1	16	267	9	260	0.03	0.94
23	13	268	8	263	0.03	0.95
32	27	269	7	249	0.03	0.90
36	9	272	4	267	0.01	0.97
56	11	265	11	265	0.04	0.96
57	21	264	12	255	0.04	0.92
61	41	258	18	235	0.07	0.85
63	52	260	16	224	0.06	0.81
64	49	256	20	227	0.07	0.82
70	53	253	23	223	0.08	0.81
77	30	261	15	246	0.05	0.89
78	42	253	23	234	0.08	0.85
85	68	238	38	208	0.14	0.75
96	17	271	5	259	0.02	0.94
98	16	258	18	260	0.07	0.94
105	55	237	39	221	0.14	0.80
108	14	264	12	262	0.04	0.95
110	44	258	18	232	0.07	0.84
115	11	272	4	265	0.01	0.96

TABLE II: Accuracy results for filtered users

VI. Conclusions and future work

In this work we investigated the effectiveness of the CNN-based Siamese network for verifying the identity using static Persian signatures in a Writer Independent setting. Our work showed that given a proper dataset, the existing solutions can achieve high accuracy, 90%, on Persian signatures. One interesting future work is to combine this approach with a two phase approach of [8], where in the first phase, the network learns the essence of a forgery in Writer Independent mode and in the second phase, a customize network is trained for each user in the Writer Dependent mode.

References

[1] M. E. Yahyatabar and J. Ghasemi, "Online signature verification using double-stage feature extraction modelled by dynamic feature stability experiment," *IET Biometrics*, vol. 6, no. 6, pp. 393–401, 2017.

[2] A. Soleimani, K. Fouladi, and B. N. Araabi, "Utsig: A persian offline signature dataset," *IET Biometrics*, vol. 6, p. 1–8, Jan 2017.

Listing 1: Base Network

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1_1 (Conv2D)             (None, 37, 53, 96)        11712
_________________________________________________________________
batch_normalization_12 (Batc (None, 37, 53, 96)        148
_________________________________________________________________
max_pooling2d_18 (MaxPooling (None, 18, 26, 96)        0
_________________________________________________________________
zero_padding2d_18 (ZeroPaddi (None, 22, 30, 96)        0
_________________________________________________________________
conv2_1 (Conv2D)             (None, 18, 26, 256)       614656
_________________________________________________________________
batch_normalization_13 (Batc (None, 18, 26, 256)       72
_________________________________________________________________
max_pooling2d_19 (MaxPooling (None, 8, 12, 256)        0
_________________________________________________________________
dropout_18 (Dropout)         (None, 8, 12, 256)        0
_________________________________________________________________
zero_padding2d_19 (ZeroPaddi (None, 10, 14, 256)       0
_________________________________________________________________
conv3_1 (Conv2D)             (None, 8, 12, 348)        802140
_________________________________________________________________
zero_padding2d_20 (ZeroPaddi (None, 10, 14, 348)       0
_________________________________________________________________
conv3_2 (Conv2D)             (None, 8, 12, 256)        802048
_________________________________________________________________
max_pooling2d_20 (MaxPooling (None, 3, 5, 256)         0
_________________________________________________________________
dropout_19 (Dropout)         (None, 3, 5, 256)         0
_________________________________________________________________
flatten (Flatten)            (None, 3840)              0
_________________________________________________________________
dense_12 (Dense)             (None, 1024)              3933184
_________________________________________________________________
dropout_20 (Dropout)         (None, 1024)              0
_________________________________________________________________
dense_13 (Dense)             (None, 128)               131200
=================================================================
Total params: 6,295,160
Trainable params: 6,295,050
Non-trainable params: 110
```


[3] D. Impedovo and G. Pirlo, "Automatic signature verification: The state of the art," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 5, pp. 609–635, 2008.

[4] 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," 2017.

[5] M. B. Yılmaz and B. Yanıkoğlu, "Score level fusion of classifiers in off-line signature verification," *Information Fusion*, vol. 32, pp. 109–119, 2016.

[6] N. Zhu, M. Qin, and Y. Yin, "Recaptured image detection based on convolutional neural networks with local binary patterns coding," in *Fourth International Workshop on Pattern Recognition*, vol. 11198, p. 1119804, International Society for Optics and Photonics, 2019.

[7] X. Zhang, J. Cui, W. Wang, and C. Lin, "A study for texture feature extraction of high-resolution satellite images based on a direction measure and gray level co-occurrence matrix fusion algorithm," *Sensors*, vol. 17, no. 7, p. 1474, 2017.

[8] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Learning features for offline handwritten signature verification using deep convolutional neural networks," *Pattern Recognition*, vol. 70, p. 163–176, Oct 2017.

[9] M. E. Munich and P. Perona, "Visual identification by signature tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 200–217, 2003.

[10] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2, Lille, 2015.

[11] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 539–546, IEEE, 2005.