Handwriting features based detection of fake signatures

Anton Akusok
anton.akusok@arcada.fi
Arcada University of Applied Sciences
Helsinki, Finland

Leonardo Espinosa-Leal
leonardo.espinosaleal@arcada.fi
Arcada University of Applied Sciences
Helsinki, Finland

Kaj-Mikael Björk
kaj-mikael.bjork@hanken.fi
Hanken School of Economics
Helsinki, Finland

Renjie Hu
renjie-hu@uiowa.edu
The University of Iowa
Iowa City, IA

Amaury Lendasse
alendass@Central.UH.EDU
University of Houston
Houston, Texas, USA

Figure 1: Predicted probabilities of being true signatures for local features of two true and one fake (*right*) signatures.

ABSTRACT

Detection of fake signatures is a hard task. In this paper, we present a novel method for detecting trained forgeries using features extracted from sliding windows with different overlaps on a public available dataset of static images of signatures. Using a linear machine learning model named Extreme Learning Machine (ELM), our methodology achieves, in average, an Equal Error Rates (EER) of 2.31% for an overlap of 90%. In line with the state-of-the-art results available in the scientific literature.

CCS CONCEPTS

• **Security and privacy** → *Usability in security and privacy*; • **Computing methodologies** → **Biometrics**; • **Applied computing** → Evidence collection, storage and analysis.

KEYWORDS

Signature verification, neural networks, biometrics

ACM Reference Format:
Anton Akusok, Leonardo Espinosa-Leal, Kaj-Mikael Björk, Renjie Hu, and Amaury Lendasse. 2021. Handwriting features based detection of fake signatures. In *The 14th PErvasive Technologies Related to Assistive Environments Conference (PETRA 2021), June 29-July 2, 2021, Corfu, Greece.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3453892.3454003

PETRA 2021, June 29-July 2, 2021, Corfu, Greece

ACM ISBN 978-1-4503-8792-7/21/06…$15.00
https://doi.org/10.1145/3453892.3454003

1 INTRODUCTION

Despite the advances in digitalization, handwritten signatures are still used daily to verify biometric identity verification. In general, handwritten signatures are classified into two groups: online based on stroke trajectory and speed information captured by specialize devices [10, 13], and offline working with an image of a signature [6], for example, scanned from legal documents. Online methods generally achieve lower EER of 1.34%[13] on MCYT dataset [14] due to the extra information available, with offline ones reaching 2.87% EER on the same dataset. Automatic identity verification of signatures is an important field of research, that relies on image processing methods like Local Binary Patterns [9] and Histogram of Oriented Gradients [16], or various Deep Learning-based methods [5] like Siamese Networks [12]. Forgeries are also classified into two groups (trained or untrained) depending on if the forger knows the original signature [6].

This work investigates signature verification based on local handwriting features; in two different settings. First one is a traditional approach with 30 labeled examples from each user: 15 true and 15 fake ones. The second setting investigates the possibility of general signature verification independent of a particular user. The assumption is that a user learns to write own true signature automatically by muscle memory at a very high writing speed; this will produce different kind of strokes than a relatively slow copying of another person's signature while consciously controlling the movements of a writing hand.

2 METHODOLOGY

Local image patches-based approach learns the difference in handwriting between a person's own signature and an attempted forge, looking only at local image features independent of the overall picture. The high computational requirements of such approach

are handled by Extreme Learning Machine [7] (ELM), with its particular implementation[1] [2] that combines the speed of a previous implementation [1] with the flexibility of modeling tools in Scikit-Learn [15] framework.

2.1 Local signature features

This work is based on a classical MCYT-75 [14] dataset containing signatures of 75 people, with each user submitting 15 samples of own true signature, accompanied by 15 forgeries of that signature by other users. Signatures are presented as high-resolution images of size 850x360 pixels.

Local feature vectors are extracted by sliding a square window across the signature image, and extracting 1024 general purpose image features from the contents of a sliding window with the pre-trained *Inception21k* [8] network that is successfully applied across various domains [3, 11]. Sliding windows use three different sizes: 64, 128 and 256 pixels; and overlap by a different percentage of a window size: by 10%, 50% or 90%. Number of sliding windows depends on the overlap percentage, increasing as the overlap between nearby windows increases. One example signature has 120 windows of 256 pixels size, 1080 windows of 128 pixels size, and 5000 windows of 64 pixels size.

2.2 Signature verification with image patches

Signature verification separates true signatures from fake ones, by looking at localized image features depicting patches of lines. Image patches are smaller than the global signature shape, so the method does not rely on the look of a whole signature.

First, feature vectors for all image patches are extracted by inference on a pre-trained Inception21k [8] convolutional neural network model. These feature vectors are used to optimize an ELM model that learns the likelihood of an image patch to come from a genuine signature. Likelihood of a whole signature to be true one is an average of the likelihoods of its image patches.

2.3 User-agnostic signature verification

An interesting research task is investigating the user-independent signature verification performance. This is performed by combining local image samples from multiple users together, so the model is forced to learn shared features that would separate true signatures from forged ones.

The model is pre-trained with data from 50 users, and the evaluation procedure runs on 5 new users at a time taking one signature per user for validation. The experiment then follows two different approaches: supervised and unsupervised. In a supervised approach, the model is updated with all the signatures of the new users except the validation one before computing its predictions – ELM models support online updating with new incoming data [1]. In an unsupervised approach, the validation predictions are computed without the model seeing any data from the corresponding users. Experiments are repeated multiple times to obtain reliable results.

[1] https://github.com/akusok/scikit-elm

Table 1: Signature verification performance with local image features sampled by sliding windows at different overlaps.

Overlap	10%	50%	90%
EER	7.11%	3.56%	2.31%

3 EXPERIMENTAL RESULTS

All experiments utilize the same optimal model structure with 10,000 hidden neurons and L2 regularization parameter $\alpha = 0.8$ found on local image sample classification. Results for full signatures are averages of their local image patches.

3.1 Optimal model parameters

The new implementation of ELM in Scikit-Learn compatible format allows for extensive parameter validation. In particular, *RandomizedSearchCV()*[2] method accepts ranges of values for multiple model parameters, then runs a given number of cross-validation experiments sampling the given parameters randomly from the specified ranges. Given a large number of random trials (10,000 in this paper), an effect of every parameter is visible by averaging out performance metrics of all other parameters. Although not a precise performance measure, it allows for fast rejection of poor parameter choices.

An exhaustive search of Extreme Learning Machine model parameters found hyperbolic tangent neurons to be the best performing ones. A detailed selection of the number of neurons and the L2 regularization parameter α (that depend on each other) is done on local image feature classification task, again with randomized parameter selection and cross-validation. The results are printed on logarithmic axes on Figure 2, showing the optimal performance regions. Predicted probabilities of local image features can be printed over the signature to visualize the results - as shown on Figure 1 at the beginning of the paper.

3.2 Signature verification performance

Signature verification is done on per-user basis, separating true signatures of a user from fake ones. All 30 samples are split into 10 groups for 10-fold cross validation, where an ELM model is trained on local features from 9 groups, and predicts probabilities of being a true signature for local features of the 10th group. Predicted probability for the whole signature is an average of the predicted probabilities of its local image patches.

Equal error rate (EER) is a balanced metric combining False Acceptance Rate and False Rejection Rate - more precisely, a point where the two rates are equal. EER results are shown in Table 1. Performance improves with denser image features sampling (higher sliding window overlap levels).

A comparison result from [12] also based on local image structure analysis achieves its best results of 2.45% EER with significantly more complicated methodology.

3.3 User-agnostic signatures verification

User-agnostic signature verification is a seemingly impossible task that pools together local image patches of all the signatures in the

[2] https://scikit-learn.org/stable/modules/grid_search.html#randomized-parameter-search

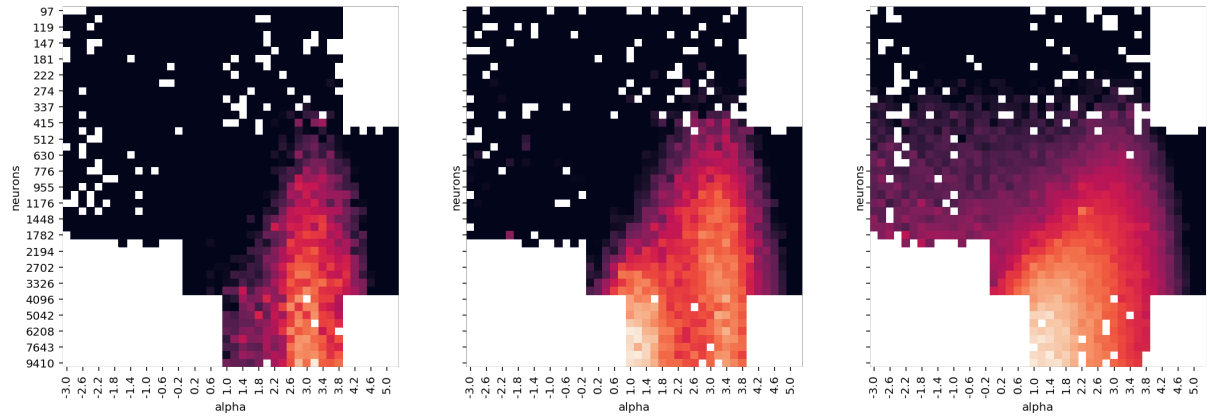

Figure 2: Local image feature classification accuracy, for 10%/50%/90% overlap. Higher overlap levels introduce a new optimal parameter zone around $\alpha = 1.0$. Highest accuracy on the figure is around 62%. The plot shows combined randomized search results over 3 different parameter ranges.

dataset, only labeling them as coming from a valid or a forged signature. One signature per user is used for validation. Patches from other signatures of the same user are included in the training dataset in *supervised* setup, and excluded in the *unsupervised* one - meaning that in an unsupervised case, the model needs to verify a signature of a completely unknown user.

ROC curves of the results with AUC and EER scores are presented on Figure 3. Interestingly, the supervised results when the model learns on the signatures of a user (all except the one held for validation) are not overwhelmingly better than the unsupervised results when the model sees no training samples of the users in the validation set. This hints that the model manages to learn useful *truthfulness* indicators from the way users write, independently of the shape of a particular signature.

4 CONCLUSIONS

This paper proposes an offline signature verification methodology that reaches state-of-the-art results while being based on off-the-shelf building blocks such as a pre-trained general purpose convolutional neural network and a Scikit-Learn compatible classifier. Signature analysis bases fully on local image patches sampled with sliding windows of different sizes, with denser sampling improving the results at the expense of higher computational requirements. Interestingly, a general-purpose CNN model outperformed custom-build models [5] applied to whole signature images, suggesting that local image patches provide extra information for the discriminative model. Online signature verification provides even more information about the signature writing process, leading to better results [13] if a suitable capture device is available.

User-independent signature verification is an interesting finding, especially the feasibility of unsupervised verification of signatures from users never before seen by a model. It suggests that features extracted from image patches by a general purpose CNN include information about writing pattern difference between a user writing own signature, and trying to forge another one's. The continuation research will focus on improving the unsupervised performance at larger signature datasets like GPDSS10000 [4].

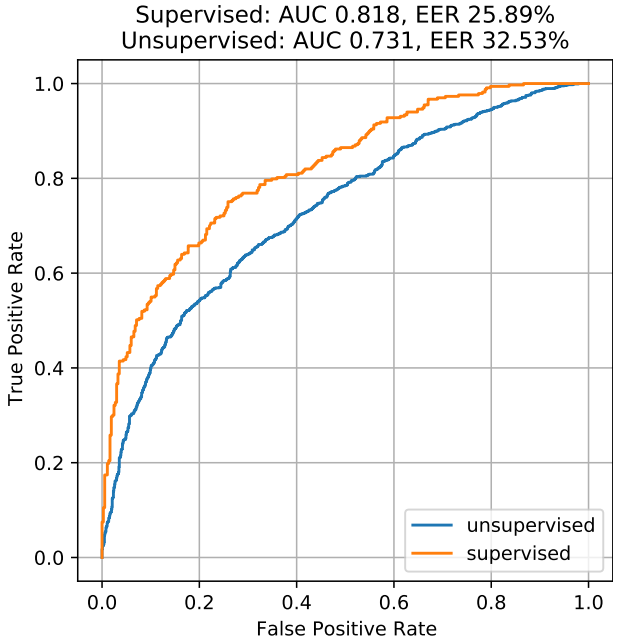

Figure 3: ROC curves for supervised and unsupervised signature classification.

5 ACKNOWLEDGMENTS

The authors wish to acknowledge Risklab at Arcada UAS for the provided GPU computational resources. The authors also wish to acknowledge CSC – IT Center for Science, Finland, for additional computational resources.

REFERENCES

[1] Anton Akusok, Kaj-Mikael Björk, Yoan Miche, and Amaury Lendasse. 2015. High-performance extreme learning machines: a complete toolbox for big data applications. *IEEE Access* 3 (2015), 1011–1025.

[2] Anton Akusok, Leonardo Espinosa Leal, Kaj-Mikael Björk, and Amaury Lendasse. 2021. Scikit-ELM: An Extreme Learning Machine Toolbox for Dynamic and

Scalable Learning. In *Proceedings of ELM2019*, Jiuwen Cao, Chi Man Vong, Yoan Miche, and Amaury Lendasse (Eds.). Springer International Publishing, Cham, 69–78.

[3] Leonardo Espinosa-Leal, Anton Akusok, Amaury Lendasse, and Kaj-Mikael Björk. 2021. Website Classification from Webpage Renders. In *Proceedings of ELM2019*, Jiuwen Cao, Chi Man Vong, Yoan Miche, and Amaury Lendasse (Eds.). Springer International Publishing, Cham, 41–50.

[4] M. A. Ferrer, M. Diaz, C. Carmona-Duarte, and A. Morales. 2017. A Behavioral Handwriting Model for Static and Dynamic Signature Synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 6 (2017), 1041–1053.

[5] Luiz G. Hafemann, Robert Sabourin, and Luiz S. Oliveira. 2017. Learning features for offline handwritten signature verification using deep convolutional neural networks. *Pattern Recognition* 70 (2017), 163–176.

[6] Luiz G Hafemann, Robert Sabourin, and Luiz S Oliveira. 2017. Offline handwritten signature verification—Literature review. In *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE, 1–8.

[7] G. Huang, H. Zhou, X. Ding, and R. Zhang. 2012. Extreme Learning Machine for Regression and Multiclass Classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 42, 2 (April 2012), 513–529.

[8] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).

[9] Edson JR Justino, Abdenain El Yacoubi, Flavio Bortolozzi, and Robert Sabourin. 2000. An off-line signature verification system using HMM and graphometric features. In *Proc. of the 4th international workshop on document analysis systems*. Citeseer, 211–222.

[10] S. Lai, L. Jin, and W. Yang. 2017. Online Signature Verification Using Recurrent Neural Network and Length-Normalized Path Signature Descriptor. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Vol. 01. 400–405.

[11] Leonardo Espinosa Leal, Kaj-Mikael Björk, Amaury Lendasse, and Anton Akusok. 2018. A Web Page Classifier Library Based on Random Image Content Analysis Using Deep Learning. In *Proceedings of the 11th PErvasive Technologies Related to Assistive Environments Conference* (Corfu, Greece) *(PETRA '18)*. Association for Computing Machinery, New York, NY, USA, 13–16.

[12] Li Liu, Linlin Huang, Fei Yin, and Youbin Chen. 2018. Off-Line Signature Verification Using a Region Based Metric Learning Network. In *Pattern Recognition and Computer Vision*, Jian-Huang Lai, Cheng-Lin Liu, Xilin Chen, Jie Zhou, Tieniu Tan, Nanning Zheng, and Hongbin Zha (Eds.). Springer International Publishing, Cham, 74–86.

[13] M. Okawa. 2019. Template Matching Using Time-Series Averaging and DTW With Dependent Warping for Online Signature Verification. *IEEE Access* 7 (2019), 81010–81019.

[14] J. Ortega-Garcia. 2003. MCYT baseline corpus: a bimodal biometric database. *IEE Proceedings - Vision, Image and Signal Processing* 150 (December 2003), 395–401(6). Issue 6.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[16] Bailing Zhang. 2010. Off-line signature verification and identification by pyramid histogram of oriented gradients. *International Journal of Intelligent Computing and Cybernetics* (2010).