OFFLINE SIGNATURE VERIFICATION USING MACHINE LEARNING

Burak Mert Gönültaş
040140003
Advisor : Assoc. Prof. Hazım Kemal Ekenel

What is signature verification ?

What is offline signature verification ?

- Online verification

  Speed, pen pressure, directions, pen stroke length etc. dynamic features.

  Not very common.

  Solved with high accuracy.

- Offline verification

  Only image.

  Widely used and accepted.

  Not fully solved yet.

  Generalization problems due to scanning quality, scan noise, lighting etc.

MOTIVATION

Provide a technique to detect handwritten signature forgeries without expert analysis and in a much shorter timespan.

Useful against forgeries in official and financial documents.

EARLIER WORKS

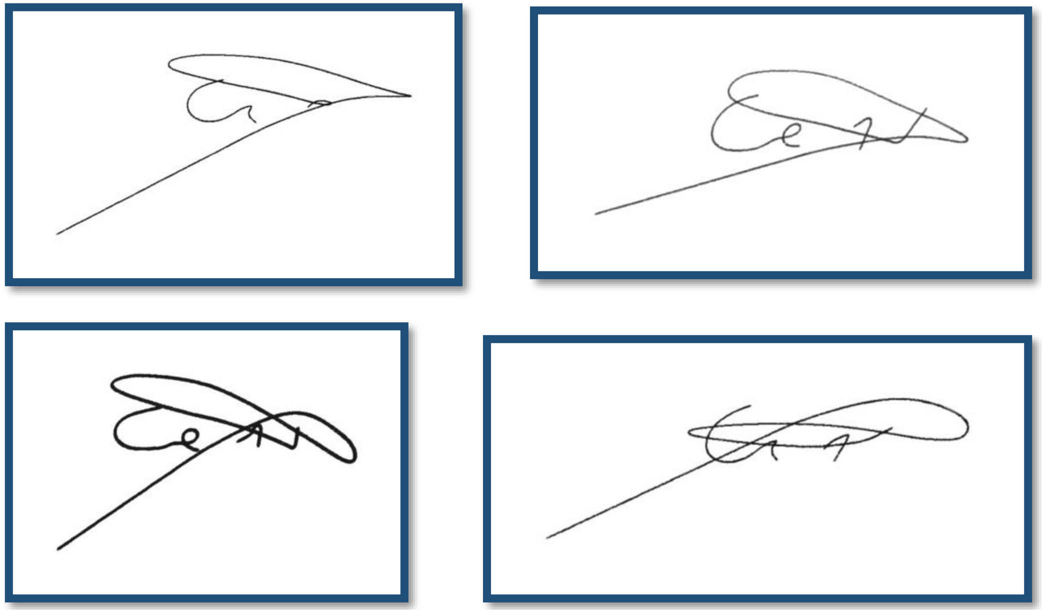
- Hand-engineered features
- Using forgeries during training
- CNNs
- Siamese Networks
- Metric Learning

FINAL GOAL

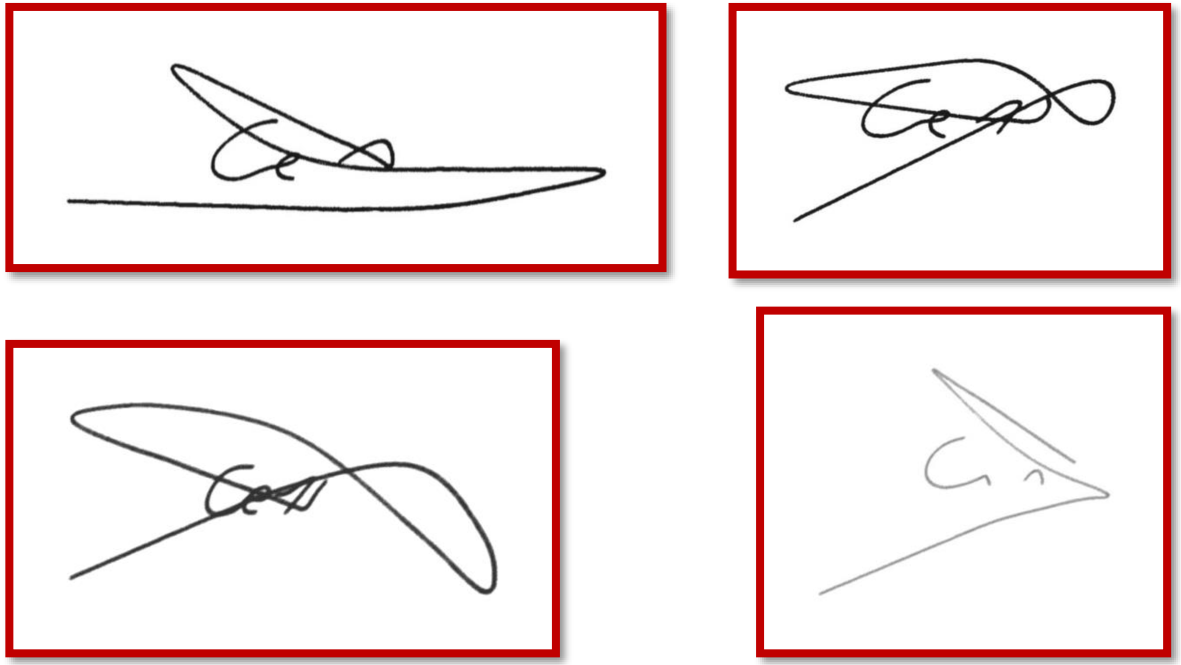
- Do not use forged signatures during training.
- Only learn discriminatory features from genuine signatures.
- Use as little genuine samples as possible.
- Accept new users with little training and adjustment overhead.

Genuines vs Forgeries
Synthetic Examples

Genuine Signatures

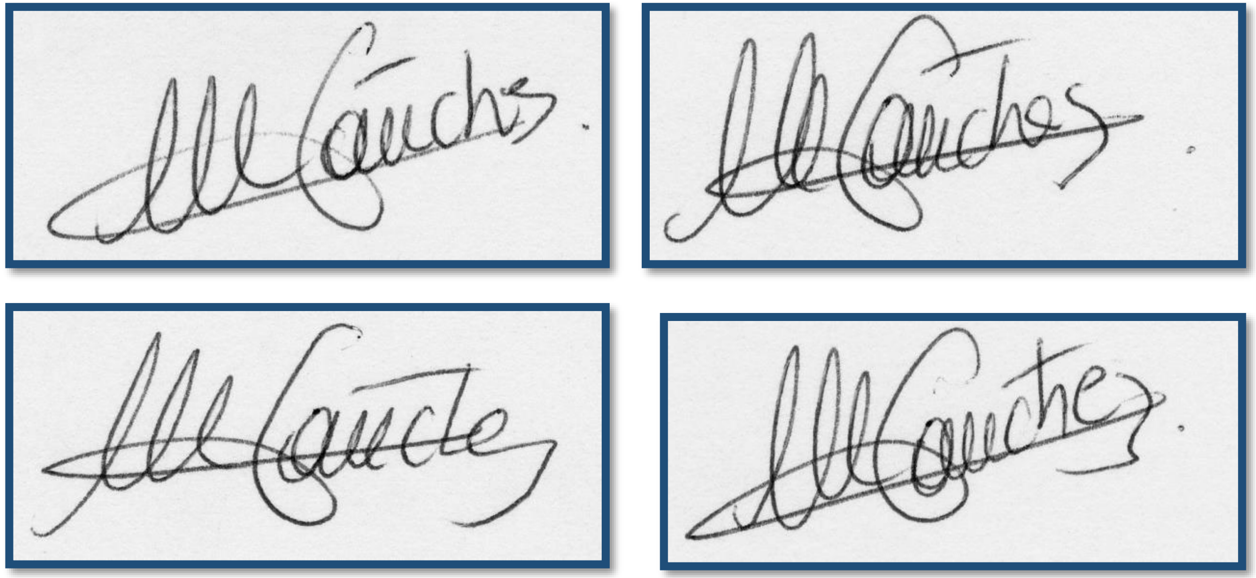


Forgeries



Genuines vs Forgeries
Biometric Examples

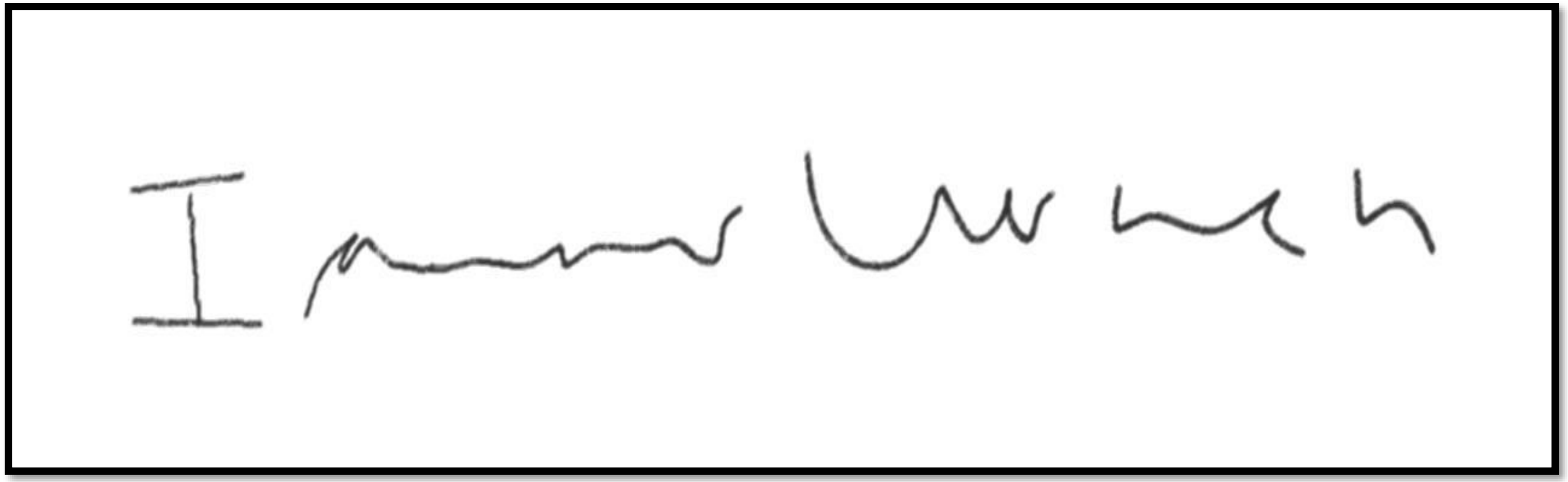
Genuine Signatures



Forgeries

Preprocessing

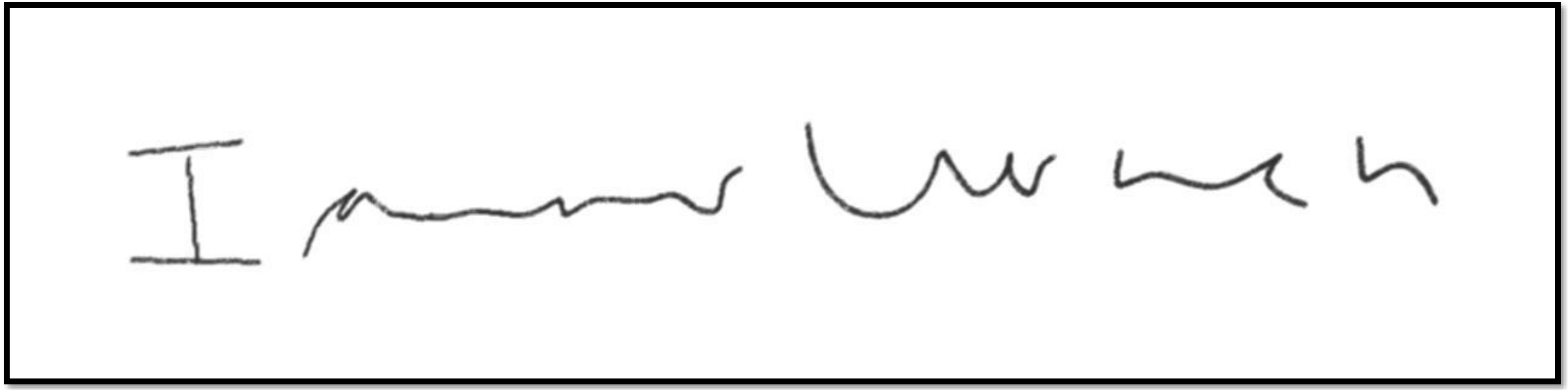
- Normalization, noise filtering, standardized in the literature.

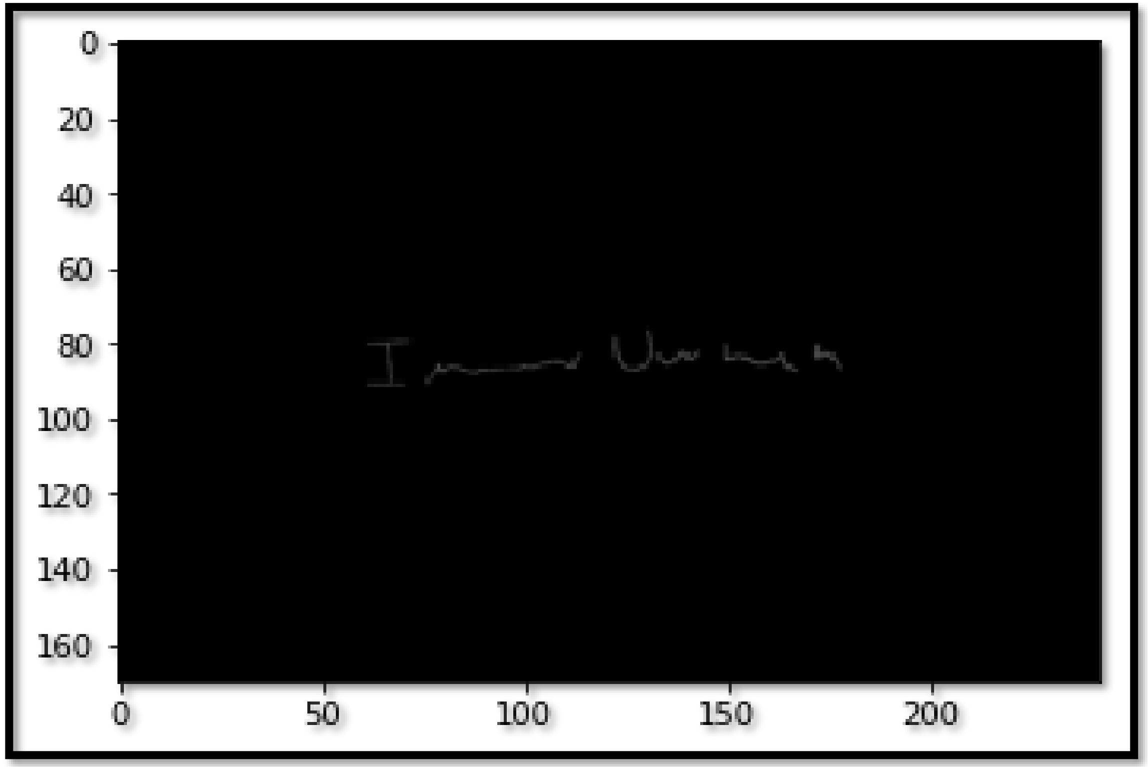


Example genuine signature

Preprocessing Steps

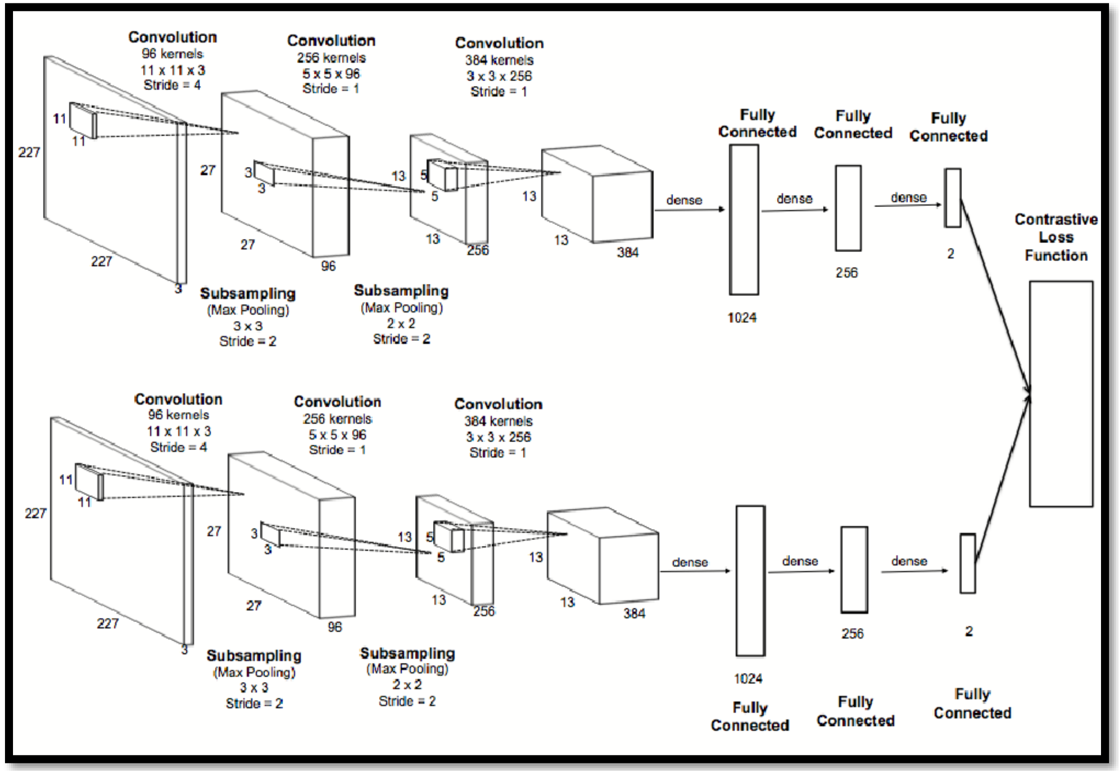
- Size normalization, center to max size and white pad. (1338x2973 for GPDS-S-4000)
- OTSU Thresholding, background set to white(255), foreground is left as-is.
- Inversion
- Resize to 170x242.
- CNN input is 150x220, data augmentation.




0
20
40
60
80
100
120
140
160
0
50
100
150
200

Initial Proposal: Siamese Networks

- Low accuracy without forged signatures.
- Difficult to train and combine with low number of samples.
- Around 66% accuracy, confirming the academic literature.



Example Siamese architecture

CNN Architecture

- Very similar to AlexNet.
- 2048 FC units instead 4096.
- Various custom loss functions for different CNN training procedures. (including skilled forgeries during writer-independent feature learning)

Layer	Size	Parameters
Input	1x150x220	
Convolution	96x11x11	stride = 4
Max. Pool	96x3x3	stride = 2
Convolution	256x5x5	stride = 1, pad = 2
Max. Pool	256x3x3	stride = 2
Convolution	384x3x3	stride = 1, pad = 1
Convolution	384x3x3	stride = 1, pad =1
Convolution	256x3x3	stride = 1, pad =1
Max. Pool	256x3x3	stride = 2
Fully Connected	2048	
Fully Connected	2048	

Main Idea

- Train CNNs to differentiate between handwritten user signatures (user-user discriminator)
- Use a less complex classifier (SVM with rbf kernel) to classify using learned features. (genuine-forgery discriminator)
- No need to re-train CNN (for new users)
- Generalized, multi-to-one comparison version of Siamese Networks

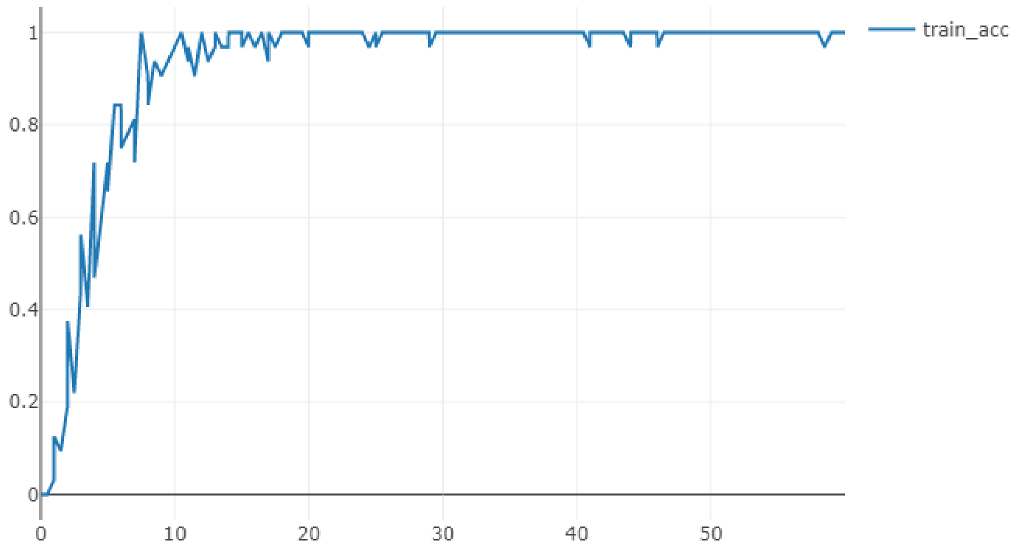
Available Datasets

- GPDS-S 4000
- MCYT-75
- GPDS (extracted features) (881 users)
- MCYT-75 (extracted features)
- CEDAR (extracted features) (55 users)

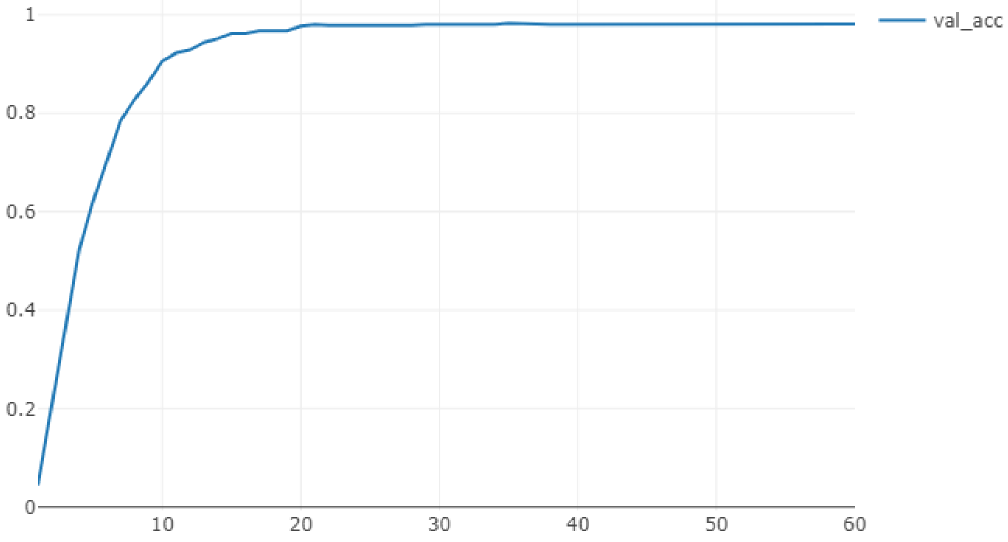
CNN Experimental Protocol

- CNN trained on 1000 users from GPDS-S-4000 (feature learning)
- 10 – 2 training - validation split (split 12 signatures)
- Multiclass user-user classification problem
- %99.8 validation accuracy
- Also forgery based feature learning tested (extra neuron, loss)
- Tested the generalization ability of the learned features from GPDS-S-4000 on MCYT-75

60 epochs training & validation accuracy plots 32 batch size

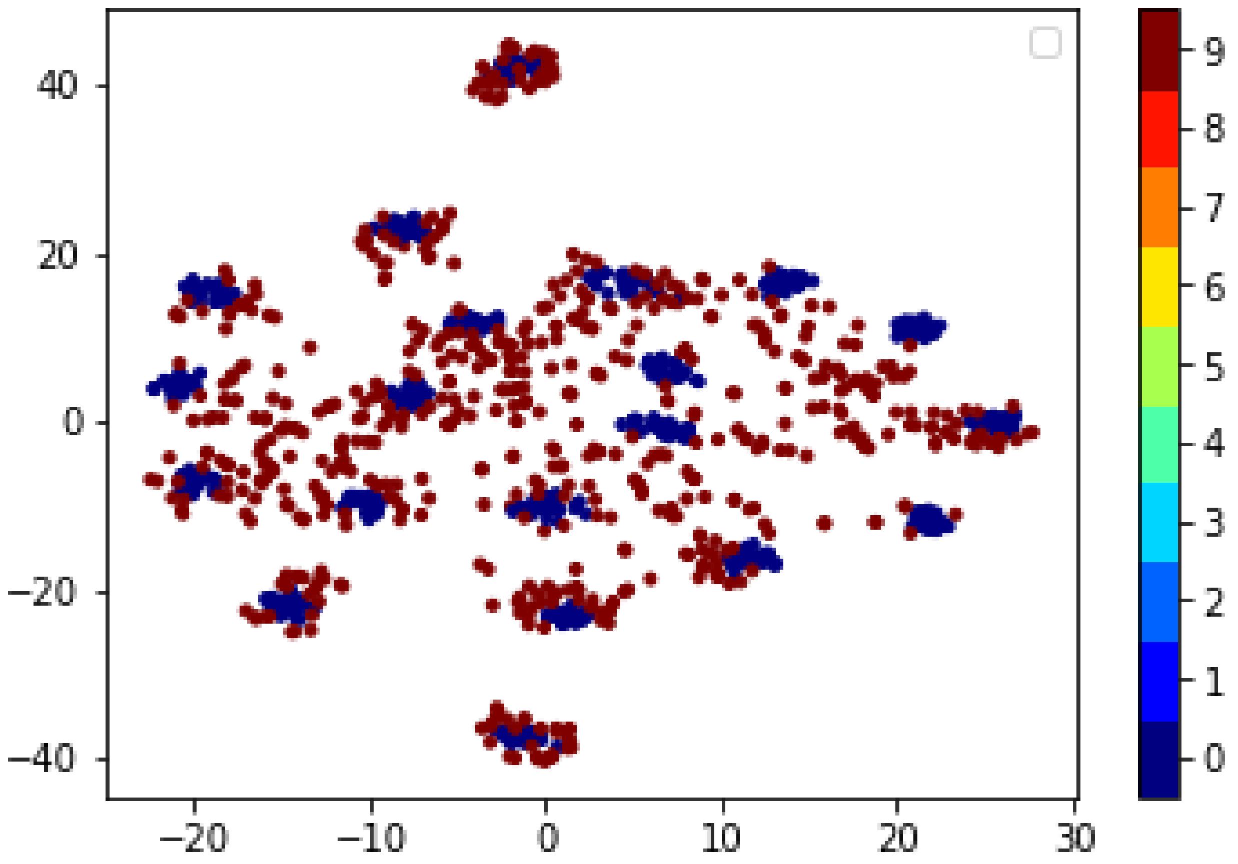


Training accuracy.



Validation accuracy.

CNN Dimensionality Reduction (t-SNE)



Randomly sampled 20 user signatures, blues are genuine, reds are forged signatures.

SVM Experimental Protocol (Feature Verification)

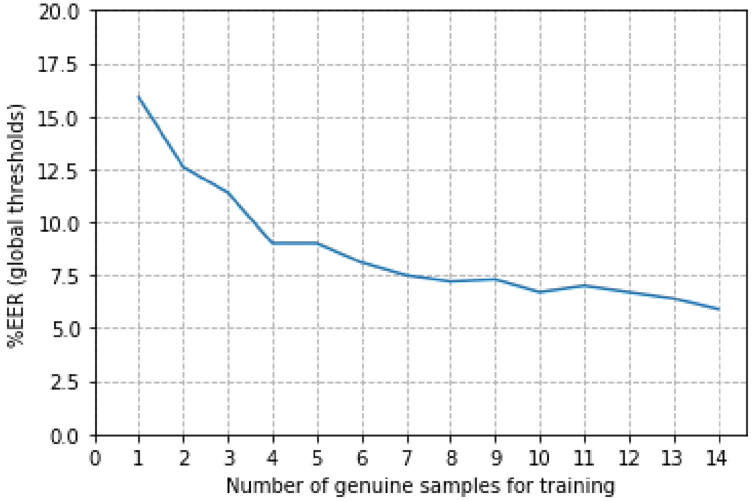
- Binary SVM classifiers for each user.
- From GPDS-S-4000, 200 users are used for feature validation and SVM hyperparameter tuning. (also decision threshold for GPDS-S 4000)
- Tuned models are tested on the rest.

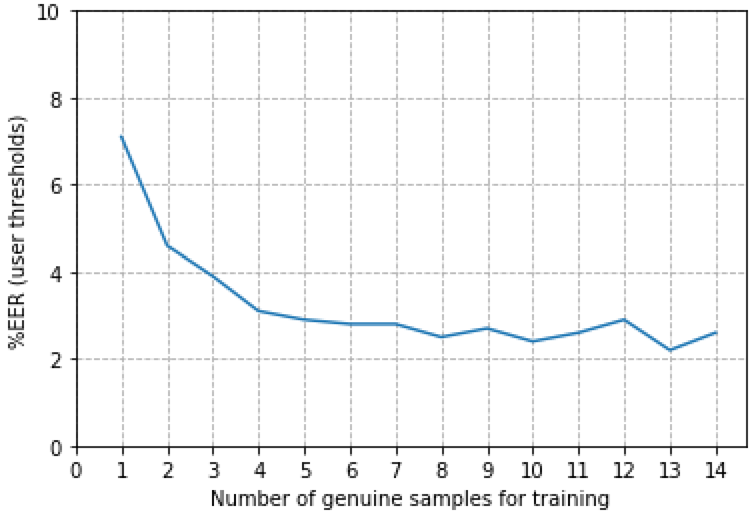
Dataset	Training Split		Testing Split
	Genuine	Random Forg.	
GPDS-Synthetic 4000	n ∈ {1,…,12}	1200 x 14	(10 genuine, 10 forgery)
MCYT- 75	n ∈ {1,…,10}	74 x 15	(5 genuine, 15 forgery)
GPDS	n ∈ {1,…,12}	581 x 14	(10 genuine, 10 forgery)
CEDAR	n ∈ {1,…,12}	54 x 12	(10 genuine, 10 forgery)

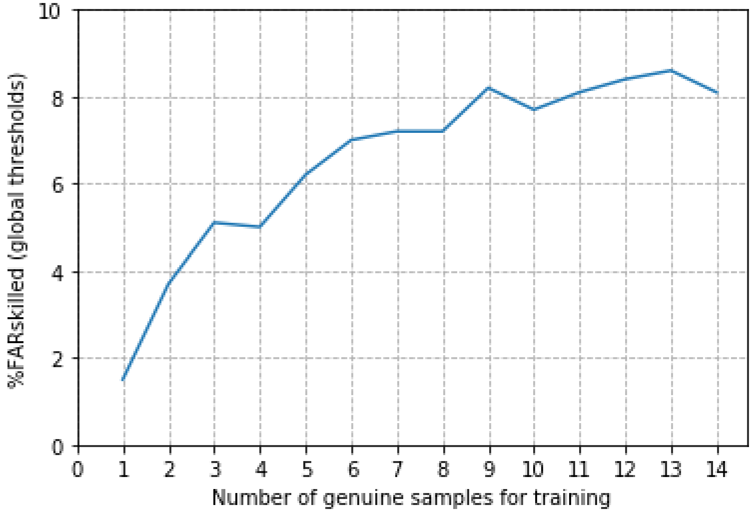
Evaluation Metrics

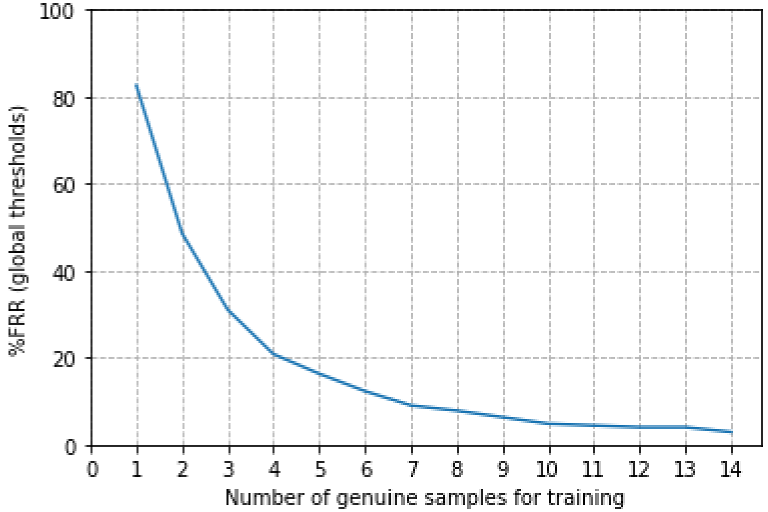
EER_{global} ,EER_{user} , FRR, $FAR_{skilled}$

Effect of Training Sample Count (GPDS-S 4000)









Performance on Datasets

Reference	EER_{global}(%)
Dutta et al. [30]	26.33
Dey et al.[25]	22.24
Soleimani et al. [31]	13.30
Ferrer et al.,[32]	16.44
Serdouk et al.,[33]	16.68
Zhang et al.[34]	14.79
Proposed	**6.96**

Performance on GPDS-S 4000, using 10 genuine 10 skilled forgeries. FRR = 7.17%, FARskilled = 6.83%.

Reference	EER_{user} (%)
Gilperez et al.[37]	6.44
Vargas et al.[36]	7.08
Ooi et al. [35]	9.87
Soleimani et al.,[31]	9.86
Hafemann et al.[14]	**2.87**
Proposed	4.00

Performance on MCYT-75, using 10 genuine 10 skilled forgery samples.

Performance using pre-extracted Features

Reference	# genuine samples	EER_{user} (%)
Hu and Chen[38]	10	7.66
Guerbai et al.[39]	12	15.07
Serdouk et al.[40]	16	12.52
Soleimani et al.[31]	10	20.94
Yilmaz.[41]	12	6.97
Hafemann et al.[14]	10	1.69
Proposed	**10**	**1.33**

Performance on GPDS using pre-extracted features for the proposal.

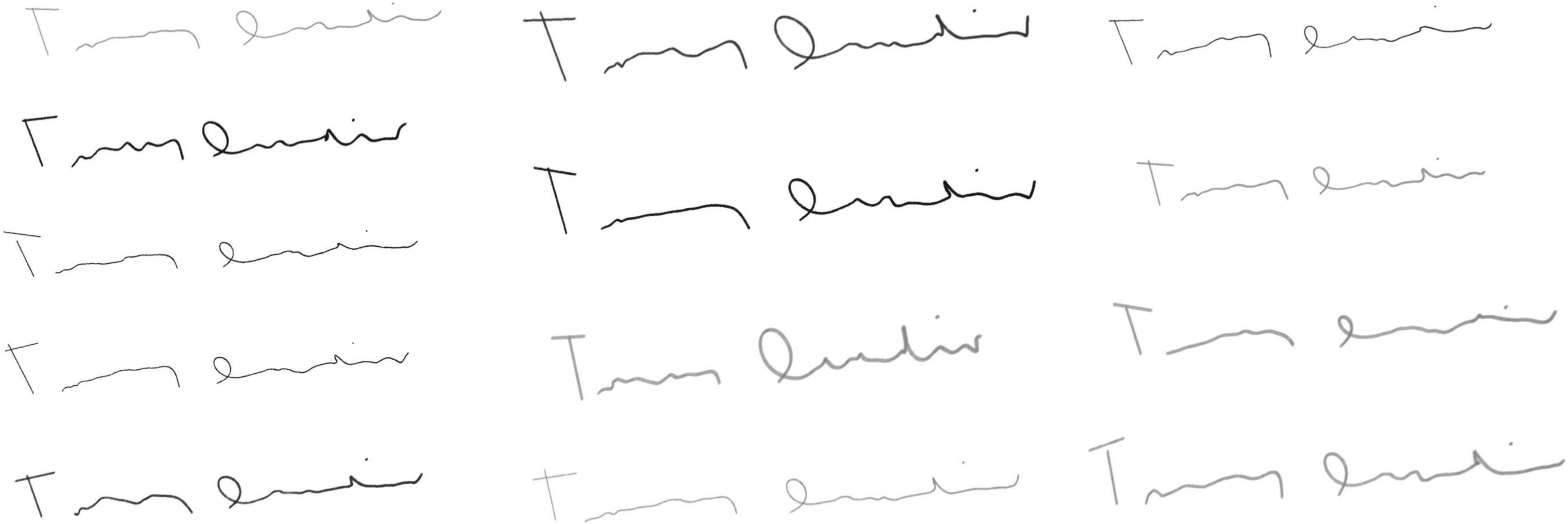
Reference	EER_{user} (%)
Gilperez et al.[37]	6.44
Vargas et al.[36]	7.08
Ooi et al. [35]	9.87
Soleimani et al.,[31]	9.86
Hafemann et al.[14]	**2.87**
Proposed	4.00
Proposed(pre-extracted)	3.14

Performance on MCYT-75, 10 samples each, included using pre-extracted features for the proposal.

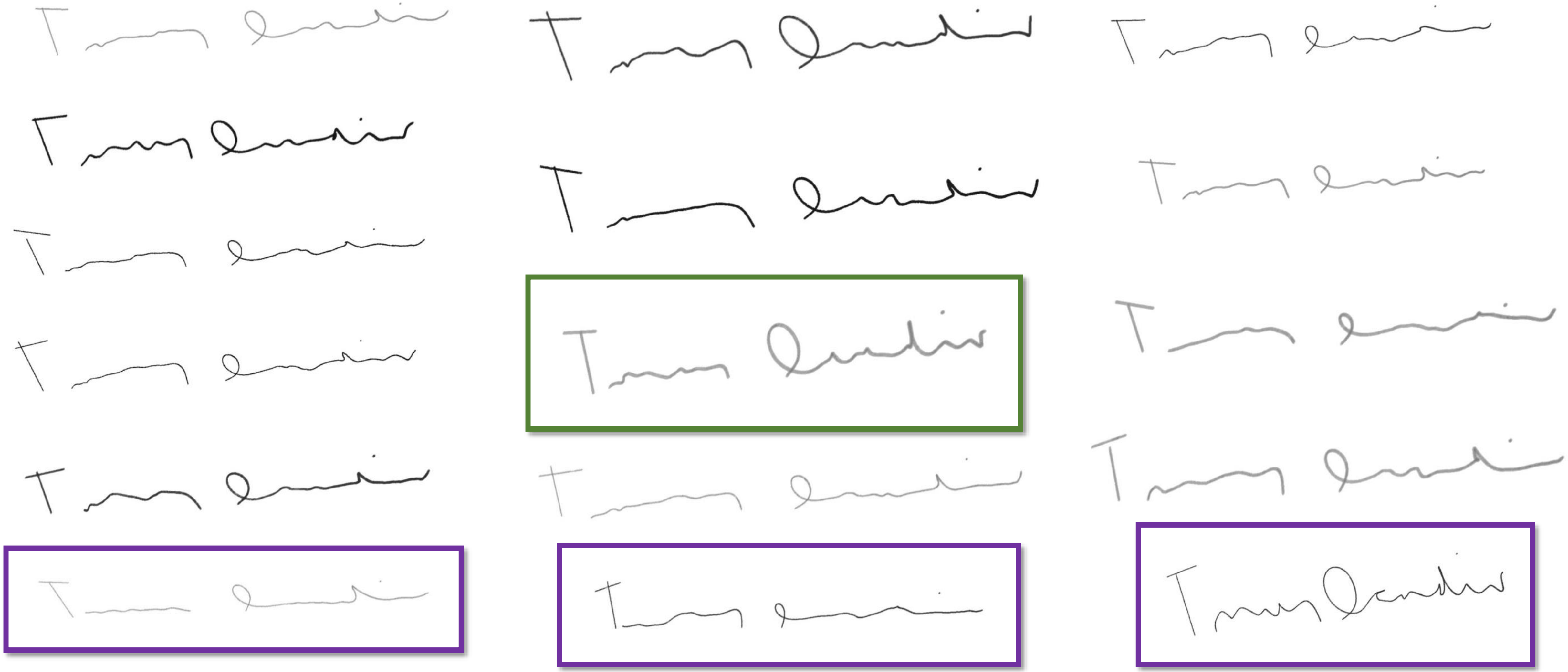
Reference	# genuine samples	EER_{user} (%)
Chen and Srihari	16	7.9
Bharatri and Shekar	12	15.07
Guerbai et al.	12	5.6
Hafemann et al.[14]	12	4.63
Proposed	**12**	**2.87**

Performance on CEDAR using pre-extracted features for the proposal.

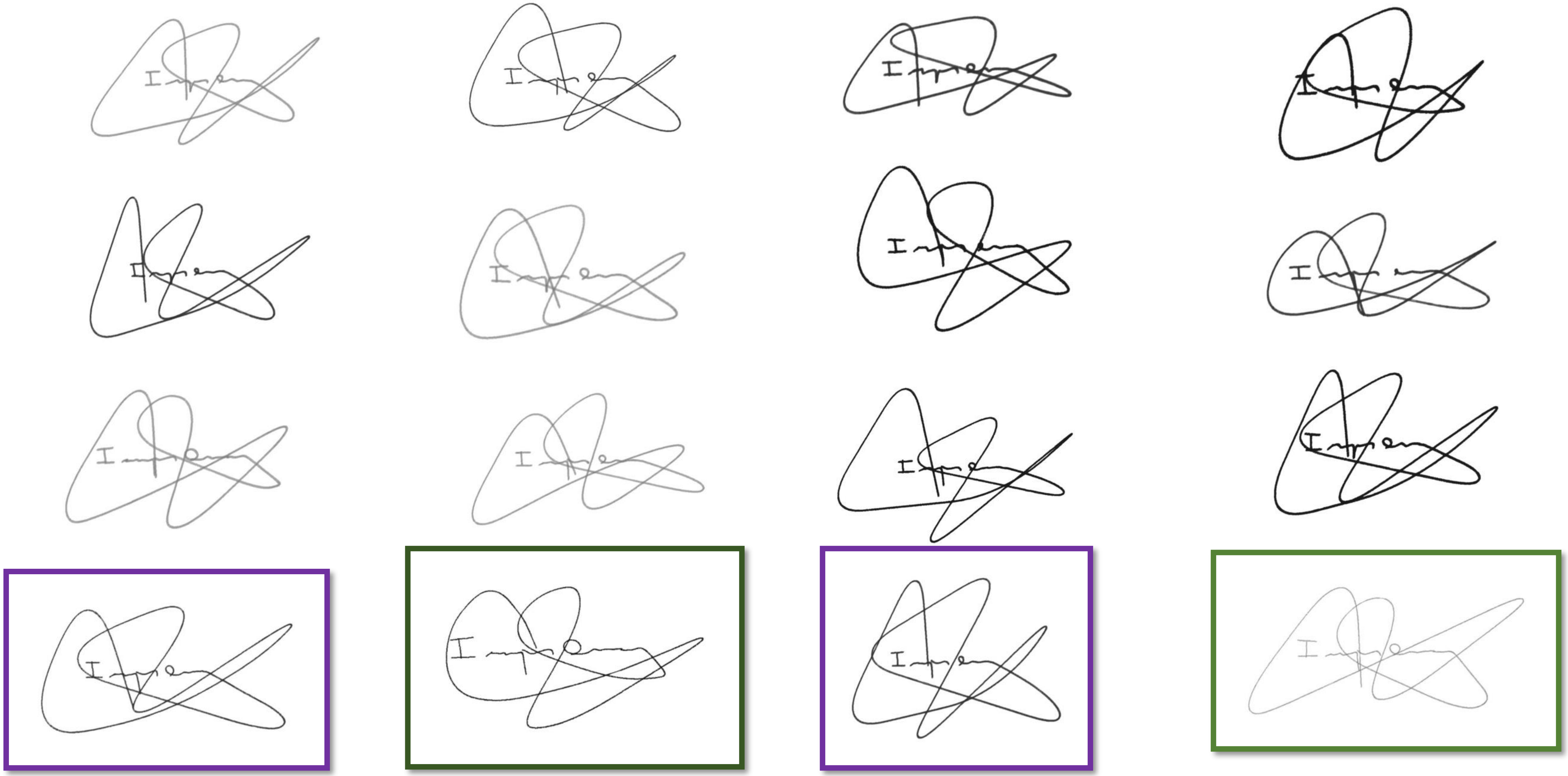
False Acceptance Example



False Acceptance Example

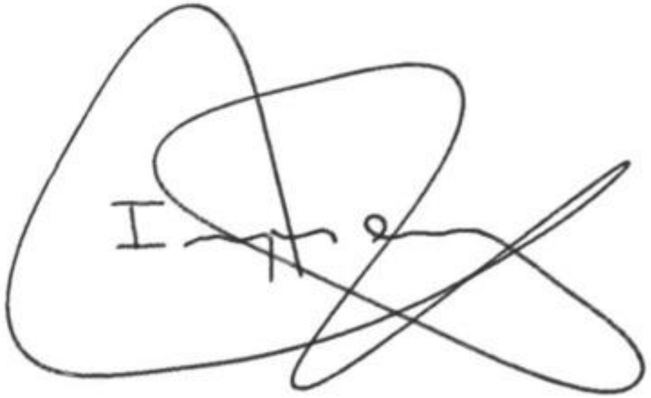


False Rejection Example

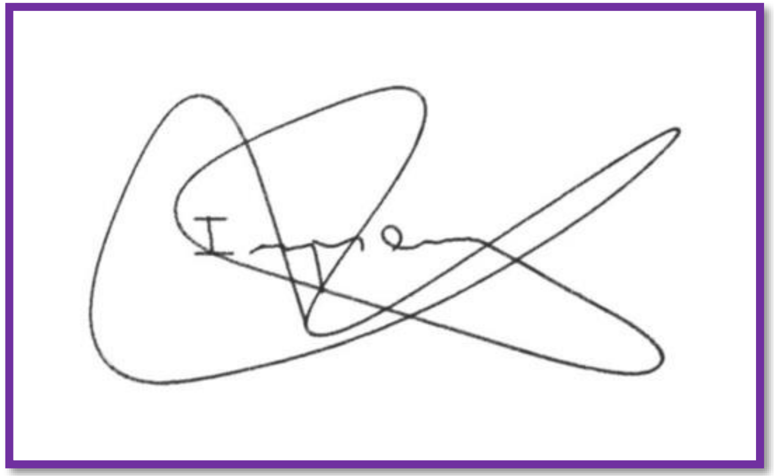


Closer Look

Training Examples



Falsely Rejected



Comparison vs State of the Art

- No superior CNN architecture
- SVM hyperparameters better tuned ca. +%0.5 score
- Loss function was changed for better forgery feature learning (generalized poorly)
- Tested decision tree and MLP based classifiers, SVM was best
- However neural architecture search can possibly find a better architecture since MLP scores were close
- Surpassed the state of the art on the synthetic new dataset. (GPDS-S 4000)

Conclusion & Future Work

- CNNs are strong feature extractors
- But deeper CNNs lose their generalization capability
- Learnable forgery features do not generalize (CNN side)
- Correct architecture is critically important
- Synthetic features generalize to biometric data
- But not as good as biometric-biometric generalization capability yet
- GAN might be very useful in the future for generating forgeries

Q & A

https://github.com/gonultasbu/sigver_v2

References

[1] H. Baltzakis and N. Papamarkos. A new signature verification technique based on a two-stage neural network classifier. *Engineering applications of Artificial intelligence*, 14(1):95–103, February 2001.

[2] A. El-Yacoubi, E. J. R. Justino, R. Sabourin, and F. Bortolozzi. Offline signature verification using HMMs and cross-validation. In *Neural Networks for Signal Processing X, 2000. Proceedings of the 2000 IEEE Signal Processing Society Workshop*, volume 2. IEEE, 2000.

[3] Luiz S. Oliveira, Edson Justino, Cinthia Freitas, and Robert Sabourin. The graphology applied to signature verification. In *12th Conference of the International Graphonomics Society*, pages 286–290, 2005.

[4] R. Sabourin and Jean-Pierre Drouhard. Off-line signature verification using directional PDF and neural networks. In *International Conference on Pattern Recognition*, pages 321–325, August 1992.

[5] J.-P. Drouhard, Robert Sabourin, and Mario Godbout. A neural network approach to off-line signature verification using directional PDF. *Pattern Recognition*, 29(3):415–424, 1996.

[6] Dominique Rivard, Eric Granger, and Robert Sabourin. Multi-feature extraction and selection in writer-independent off-line signature verification. *Int. Journal on Doc. Analysis and Recognition*, 16(1), 2013.

[7] Bailing Zhang. Off-line signature verification and identification by pyramid histogram of oriented gradients. *International Journal of Intelligent Computing and Cybernetics*, 3(4):611–630, 2010.

[8] R. Sabourin and G. Genest. An extended-shadow-code based approach for off-line signature verification. I. Evaluation of the bar mask definition. In *Int. Conference on Pattern Recognition*, October 1994.

[9] Dominique Rivard, Eric Granger, and Robert Sabourin. Multi-feature extraction and selection in writer-independent off-line signature verification. *Int. Journal on Doc. Analysis and Recognition*, 16(1), 2013.

[10] G.S. Eskander, R. Sabourin, and E. Granger. Hybrid writer-independentwriter-dependent offline signature verification system. *IET Biometrics*, 2(4):169–181, December 2013.

[11] Yoshua Bengio. Learning Deep Architectures for AI. *Found. Trends Mach. Learn.*, 2(1):1–127, January 2009.

[12] Luiz G. Hafemann, Robert Sabourin, and Luiz S. Oliveira. Analyzing features learned for offline signature verification using Deep CNNs. In *International Conference on Pattern Recognition*, pages 2989–2994, 2016.

[13] Luiz G. Hafemann, Robert Sabourin, and Luiz S. Oliveira. Writer independent feature learning for Offline Signature Verification using Deep Convolutional Neural Networks. In *International Joint Conference on Neural Networks*, pages 2576–2583, July 2016.

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

[15] H. Rantzsch, H. Yang, and C. Meinel. Signature embedding: Writer independent offline signature verification with deep metric learning. In *Advances in Visual Computing*. Springer, 2016.

[16] Z. Zhang, X. Liu, and Y. Cui. Multi-phase offline signature verification system using deep convolutional generative adversarial networks. In *2016 9th International Symposium on Computational Intelligence and Design (ISCID)*, volume 02, pages 103–107, 2016.

[17] M. A. Ferrer, M. Diaz-Cabrera, A. Morales, (2015), "Static Signature Synthesis: A Neuromotor Inspired Approach for Biometrics", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.37, n.3, pp. 667-680.

[18] M. A. Ferrer, M. Diaz-Cabrera, A. Morales, "Synthetic Off-Line Signature Image Generation", 6th IAPR International Conference on Biometrics, Madrid, 4-7 June 2013, pp. 1 - 7. doi: 10.1109/ICB.2013.6612969

[19] Ortega-Garcia, J., Fierrez-Aguilar, J., Simon, D., Gonzalez, J., Faundez-Zanuy, M., Espinosa, V., ... & Escudero, D. (2003). MCYT baseline corpus: a bimodal biometric database. *IEE Proceedings-Vision, Image and Signal Processing*, *150*(6), 395-401.

[20] M. Blumenstein, Miguel A. Ferrer, J.F. Vargas, \93 The 4NSigComp2010 off-line signature verification competition: Scenario 2 \94, in proceedings of 12th International Conference on Frontiers in Handwriting Recognition, ISSBN: 978-0-7695-4221-8, pp. 721-726, Kolkata, India, 16-18 November 2010.

[21] M. K. Kalera, S. Srihari, A. Xu, Offline signature verification and identification using distance statistics, International Journal of Pattern Recognition and Artificial Intelligence 18 (07) (2004) 1339–1360. doi:10.1142/S0218001404003630.

[22] N. Otsu, A threshold selection method from gray-level histograms, IEEE Transactions on Systems, Man, and 33 Cybernetics 9 (1) 62–66. doi:10.1109/TSMC.1979.4310076.

[23] Nair, Vinod, and Geoffrey E. Hinton. "Rectified linear units improve restricted boltzmann machines." *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010.

[24] S. Chopra, R. Hadsell, Y. LeCun, Learning a similarity metric discriminatively,

with application to face verification, in: CVPR, 2005, pp. 539–546.

[25] Dey, S., Dutta, A., Toledo, J. I., Ghosh, S. K., Lladós, J., & Pal, U. (2017). Signet: Convolutional siamese network for writer independent offline signature verification. *arXiv preprint arXiv:1707.02131*.

[26] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

[27] S. Ioffe, C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate

Shift, in: Proceedings of The 32nd International Conference on Machine Learning, 2015, pp. 448–456.

[28] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273-297.

[29] Gonultas, B. M. (n.d.). Sigver_v2. Retrieved May 19, 2019, from github.com/gonultasbu/sigver_v2

[30] A. Dutta, U. Pal, J. Llad´os, Compact correlated features for writer independent signature verification, in: ICPR, 2016, pp. 3411–3416.

[31] Soleimani, A., Araabi, B. N., & Fouladi, K. (2016). Deep Multitask Metric Learning for Offline Signature

Verification. Pattern Recognition Letters, 80, 84-90. doi: 10.1016/j.patrec.2016.05.023

[32] Ferrer, M. A., Diaz-Cabrera, M., & Morales, A. (2015). Static signature synthesis: A neuromotor inspired approach for biometrics. *IEEE Transactions on pattern analysis and machine intelligence, 37*(3), 667-680.

[33] Serdouk, Y., Nemmour, H., & Chibani, Y. (2017). Handwritten signature verification using the quad-tree histogram of templates and a Support Vector-based artificial immune classification. *Image and Vision Computing, 66*, 26-35.

[34] Zhang, Z., Liu, X., & Cui, Y. (2016). *Multi-phase Offline Signature Verification System Using Deep Convolutional Generative Adversarial Networks.* Paper presented at the Computational Intelligence and Design (ISCID), 2016 9th International Symposium on.

[35] S. Y. Ooi, A. B. J. Teoh, Y. H. Pang, B. Y. Hiew, Image-based handwritten signature verification using hybrid methods of discrete radon transform, principal component analysis and probabilistic neural network 40 274–282. doi:10.1016/j.asoc.2015.11.039.

[36] J. F. Vargas, M. A. Ferrer, C. M. Travieso, J. B. Alonso, Off-line signature verification based on grey level information using texture features, Pattern Recognition 44 (2) (2011) 375–385. doi: 10.1016/j.patcog. 2010.07.028.

[37] A. Gilperez, F. Alonso-Fernandez, S. Pecharroman, J. Fierrez, J. Ortega-Garcia, Off-line signature verification using contour features, in: 11th International Conference on Frontiers in Handwriting Recognition, Montreal, Quebec-Canada, August 19-21, 2008, CENPARMI, Concordia University, 2008.

[38] J. Hu, Y. Chen, Offline Signature Verification Using Real Adaboost Classifier Combination of Pseudo-dynamic Features, in: Document Analysis and Recognition, 12th International Conference on, 2013, pp. 1345–1349. doi:10.1109/ICDAR.2013.272.

[39] Y. Guerbai, Y. Chibani, B. Hadjadji, The effective use of the one-class SVM classifier for handwritten signature verification based on writer-independent parameters, Pattern Recognition 48 (1) (2015) 103–113. doi:10. 1016/j.patcog.2014.07.016.

[40] Y. Serdouk, H. Nemmour, Y. Chibani, New gradient features for off-line handwritten signature verification, in: 2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA), 2015, pp. 1–4. doi:10.1109/INISTA.2015.7276751.

[41] M. B. Yilmaz, B. Yanikoglu, Score level fusion of classifiers in off-line signature verification, Information Fusion 32, Part B (2016) 109–119. doi: 10.1016/j.inffus.2016.02.003.

[42] Python Software Foundation. Python Language Reference, version 2.7. Available at http://www.python.org

[43] Travis E, Oliphant. **A guide to NumPy**, USA: Trelgol Publishing, (2006).

[44] Fernando Pérez and Brian E. Granger. IPython: A System for Interactive Scientific Computing, Computing in Science & Engineering, 9, 21-29 (2007), DOI:10.1109/MCSE.2007.53 (publisher link)

[45] John D. Hunter. **Matplotlib: A 2D Graphics Environment**, Computing in Science & Engineering, **9**, 90-95 (2007),DOI:10.1109/MCSE.2007.55 (publisher link)

[46] Wes McKinney. **Data Structures for Statistical Computing in Python**, Proceedings of the 9th Python in Science Conference, 51-56 (2010) (publisher link)

[47] Travis E. Oliphant. **Python for Scientific Computing**, Computing in Science & Engineering, **9**, 10-20 (2007),DOI:10.1109/MCSE.2007.58 (publisher link)

[48] Stefan Behnel, Robert Bradshaw, Craig Citro, Lisandro Dalcin, Dag Sverre Seljebotn and Kurt Smith. **Cython: The Best of Both Worlds**, Computing in Science and Engineering, **13**, 31-39 (2011), DOI:10.1109/MCSE.2010.118 (publisher link)

[49] Paszke, A., Gross, S., Chintala, S., & Chanan, G. (2017). Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration. *PyTorch: Tensors and dynamic neural networks in Python with strong GPU acceleration*, *6*.