I. J. Engineering and Manufacturing, **2021, 5, 41-47**
Published Online October 2021 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijem.2021.05.04

Online Signature Verification Using Fully Connected Deep Neural Networks

Snehal Reddy Yelmati
Department of Computer Science and Engineering, MVSR Engineering College, Hyderabad, Telangana, India
E-mail: snehalyelmati@gmail.com

Jayasree Hanumantha Rao
Professor, Department of Computer Science and Engineering, MVSR Engineering College, Hyderabad, Telangana, India
E-mail: jayasree_cse@mvsrec.edu.in

Received: 08 May 2021; Accepted: 26 July 2021; Published: 08 October 2021

Abstract: Biometric systems have been used in a wide range of applications. In this paper, we have introduced an online signature verification system using deep neural network models. The proposed system is designed to be used in a production environment and has accuracies on par with the state-of-the-art signature verification methods. It authenticates much faster than most of the existing signature verification systems (less than 2 seconds). To achieve better accuracies and faster training times, a feature vector with 42 features, both static and dynamic, is obtained from the signature sample. This feature vector is fed into the user identification model, which predicts the identity of the user with about 99% accuracy and based on this prediction, the user authentication model predicts if the signature is genuine or forged for that recognized user, with about 98% accuracy. The best possible accuracy achieved by the proposed system for 40 users is 97.5% and EER about 2%. The dataset from the Signature Verification Competition 2004 (SVC2004) was used to assess the performance of the proposed system. The results show that the proposed system competes with and even outperforms existing methods.

Index Terms: Online signature verification, Deep learning, Neural networks, SVC2004.

1. Introduction

Signature verification is the most widely used authentication system. There are two types of signature verification systems, namely offline and online. Offline systems use the hard copy of the signature to compare and authenticate the signature sample. On the other hand, the online signature verification system uses the data extracted from the original signature and the data extracted from the signature sample to compare the two signatures. This signature data extraction is done using a digitizing tablet or pressure-sensitive tablet. Online signature verification system has an edge over offline signature verification because there can be a lot more features like the pressure at different points or the angle of stylus which are very hard to duplicate and are very specific to a person.

The extracted data includes features like the X and Y coordinates, timestamps, button status (if the stylus is in contact with the tablet or not), azimuth, etc. All this data helps in verifying the signatures and detecting forgery. But this data by itself is not enough to identify or authenticate signatures accurately, we have analyzed the data from digitized tablet and generated about 40 features which can be used to verify signatures with better accuracy and precision (refer to Appendix 5.1 for the features).

There are three types of forgeries, random forgery, skilled forgery, and unskilled forgery. The dataset used is the First Signature Verification Competition 2004 (SVC2004) dataset which has forged signatures that come under skilled forgery. Any biometric system relies on the presumption that individuals are physically and behaviorally distinctive in several ways [1]. Even signatures of the same user have some minute differences. Making the signature verification system tolerant to these differences increases the total accuracy of the system.

There are some good signature verification systems that classify signatures accurately but most of them can't be used in real-time as they take a lot of time to verify or authenticate signatures. The goal of this research is to find a way to classify signatures accurately, taking the least time possible, that way this system can be applied in real-time signature verification systems that can be deployed in production and be used to authenticate users without any hassle.

2. Literature Survey

Most of the signature verification systems use the Dynamic Time Warping technique (or DTW)[2, 3, 4, 5, 6] or Hidden Markov Models (or HMM)[7, 8, 9]. Some of them use neural networks or recurrent neural networks to authenticate signatures [10, 11, 12]. Most of these signature verification systems were built for the most accuracy possible. The two main goals of our signature verification system are, the accuracy of the signature verification system should be on par with the state-of-the-art signature verification systems and it should be feasible to be used in a production environment. So, we choose neural networks for signature verification as they have faster prediction times and provide accuracy better than most of the available signature verification systems.

3. Methodology

For the development and evaluation of the machine learning model we've used the SVC2004 dataset. There are a total of 40 unique users with 40 signature samples each, we have set aside some signature samples of every user to make sure unseen data is used to test and validate the machine learning model that developed (please refer to the section 5.2 for more details). We have used Keras and Tensorflow 2.0 for training and testing the machine learning models; Python libraries Pandas, Matplotlib and Seaborn to analyze and visualize data.

4. Proposed System

1. Dataset

Dataset from First Signature Verification Competition 2004 (SVC2004) [13, 14] was used for training and testing the performance of the deep neural network models. This dataset comprises 40 signatures of 40 users each i.e., 1600 signature samples. Each genuine/forgery signature is stored in a separate text file. The file names are in the format "USERx_y.txt", where x (1...40) indicates the user and y (1...40) indicates one signature instance of the corresponding user, with the first 20 (1...20) representing genuine signatures and the rest (21...40) representing skilled forgeries provided by the other users. The basic stages of the proposed signature verification system are depicted in Fig 1.

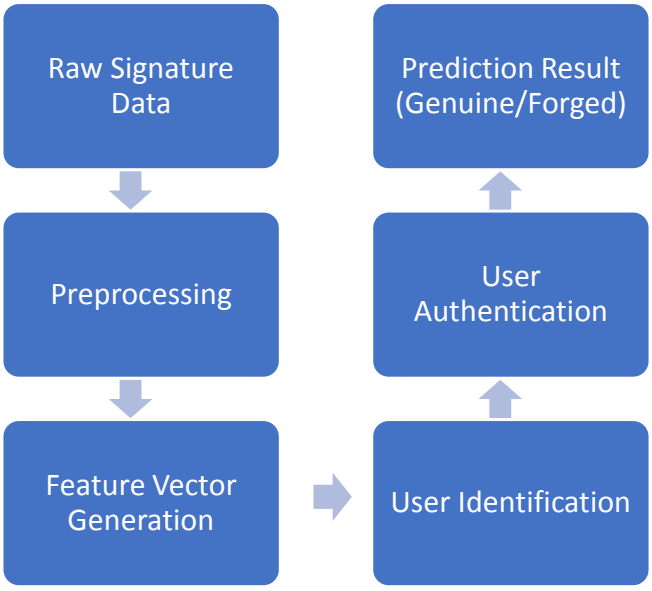

Fig. 1. Proposed system

2. Preprocessing

Preprocessing data is one of the most important aspects of neural network modeling. In this phase, both the training and testing datasets are preprocessed to remove some known anomalies in the data and format the data in the desired format. For instance, in the SVC2004 dataset, in some signature data files, there were some tuples where there are two distinct X and Y coordinates with the same timestamps. This is an anomaly because the tip of the stylus cannot be on two different points at the same time. Tuples like these had to be preprocessed; otherwise, some features like the velocity and acceleration would have been undefined. To handle this issue we have modified the timestamp of the second tuple to the average of the timestamps of the above and below tuples [15]. In addition to this, attributes like the

standard deviation of X, the standard deviation of Y, acceleration, velocity, the standard deviation of velocity, etc. are added to the 7 attributes of the SVC2004 dataset. Features like altitude and azimuth (aka pen orientation) were excluded from the feature set as they harmed the accuracy[13].

3. Feature vector generation and normalization

In this phase, a feature vector is generated based on the preprocessed signature data from the previous step. From 13 attributes in the preprocessed signature data, 42 static and dynamic features like average velocity, pen up to pen down ratio, maximum pressure, range of pressure, a variance of X velocity, number of local minima in the X direction, etc.[16] is generated (refer to the appendix for the full list of features generated). Using these 42 features a feature vector is generated and normalized (-3, 3) with min-max normalization. Normalization is useful in this case because the features generated have drastically different ranges and this can lead to longer training and testing times as the gradient descent can oscillate back and forth and take a long time to reach the global minimum [17].

4. User identification

The normalized feature vector obtained in the previous step is fed into the neural network to predict the id of a user to whom the signature belongs to. This is done using the user identification model. This model has 9 layers in total, 6 hidden layers, one dropout layer, one input layer, and one output/SoftMax layer. All the layers have rectified linear (or ReLu, refer to equation 1) activation function except the output layer. Refer to Fig 2 for additional details about the model's architecture.

$$R(z) = \max(0, z) \tag{1}$$

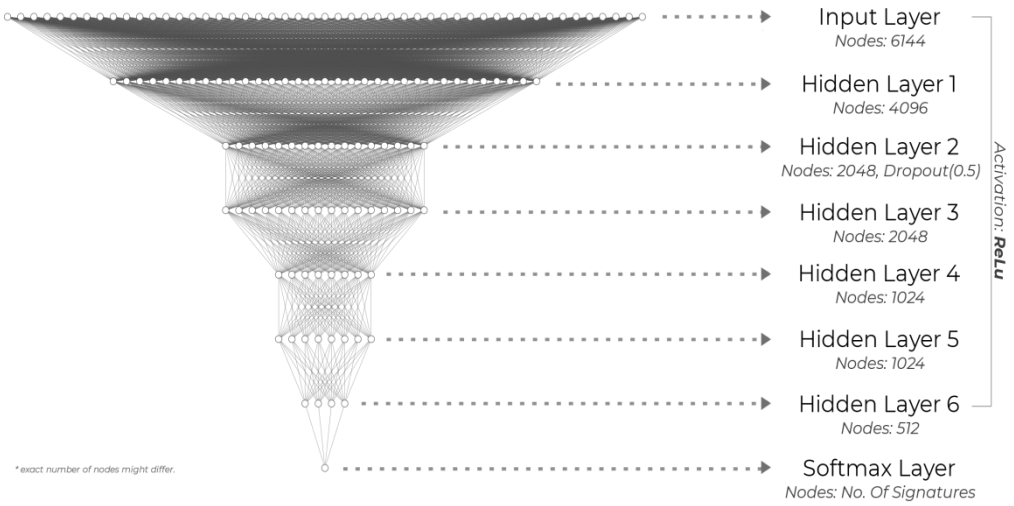

Fig. 2. User identification model

This model was tuned manually and the hyperparameters that proved to be the most useful were,

- Number of layers and Number of neurons – refer to Fig 1
- Optimizer – RMSprop
  - Rho – 0.9
- Epochs – 75
- Learning rate – used a dynamic learning rate
  - Initial learning rate – 10^{-4}
  - Patience – 3
  - Factor by which the learning rate was reduced – 0.25
  - Minimum learning rate – 10^{-6}

5. User authentication model

Based on the user id generated by the user identification model second model, user authentication model is selected. Every user has a unique second model, which is trained on his/her genuine and forged signature samples. This model is responsible for classifying the signature sample, genuine or forged. This model has four layers in total; there are two hidden layers, one input layer, and one output layer. All the layers except the output layer have the rectified linear (or ReLu) activation function, the output layer has the sigmoid function (refer to equation 2) as the activation function. This could be due to the complexity of the individual signature, for instance, if the signature is very easy to

mimic then a skilled forger could easily forge the signature, hence reducing the accuracy of the model. This was the most optimal architecture we have found in our testing after handling the bias and variance tradeoff. Refer to Fig 3 for additional details about the model's architecture.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2}$$

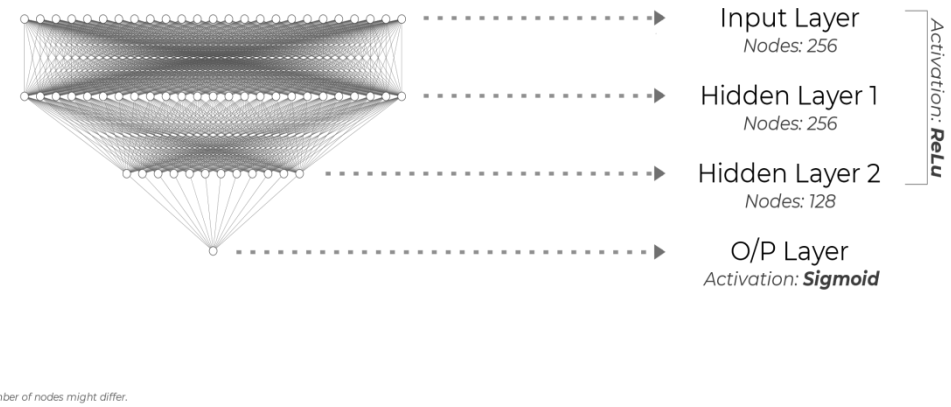

Fig. 3. User authentication model.

This model was manually tuned and the hyperparameters that proved to be most useful were,

- Number of layers and Number of neurons – refer to Figure 3.2
- Optimizer – ADAM
  - Beta 1 – 0.9
  - Beta 2 – 0.999
- Epochs – 100
- Learning rate – 10^{-4}

6. Summary

This proposed system uses a two-step signature verification process. The first step is to identify the owner of the signature and the second step is to classify the signature if it is a genuine sample or forged. When compared to signature verification systems [15] with a similar process, we have achieved better accuracy and error rates with less number of iterations. Our proposed system has accuracy on par with the state-of-the-art signature verification systems like these [4, 11, 18] but has faster prediction times than most of the signature verification systems. For instance prediction times for [11, 19] and our proposed system are 7.5s, 4s and <2s.

5. Performance Evaluation

1. Experimental setup

For training and testing relatively bigger machine learning models require computers with ample amount of Video RAM (VRAM), hardware specifications of the system we have used for this research,

- CPU specifications: 4 cores of Intel(R) Xeon(R) CPU @ 2.30GHz with 16 GB RAM
- GPU specifications: NVIDIA Tesla P100 as the GPU with 16GB VRAM

2. Training and testing

For the first model, the user identification model, we have used a 90:10 split of training and testing data. For the second model user authentication model, we have used an 80:20 split of training and testing data. Both the models have been trained for 75 and 100 epochs each respectively. These numbers of epochs proved to be enough for training with some reasonable margin for consistency. We have used only the publicly available SVC2004 dataset which has 40 signatures of 40 users, 1600 signatures in total unlike most of the other signature verification systems where they have used a much larger dataset like MYCT[20] or BioSecure Signature Subcorpus[20].

3. Results with comparison

Usually in most signature verification systems, when a signature has to be verified, it is compared with some genuine samples using some transformations and feature extraction. If the similarity metric yields an acceptable value, then the signature sample is said to genuine otherwise forged.

One major difference between most signature verification systems versus our proposed system is that when a signature sample is fed into the signature verification system, our proposed system can automatically detect the correct user authentication model (as every user has to have a unique model to authenticate the signature sample), and verify its authenticity (for the SVC2004 dataset the maximum users we could test was 40 users). The best accuracy obtained on unseen data of 40 users in the publicly available SVC2004 dataset is 97.5%. Time taken to verify signatures are under 2 seconds which is desirable for real-time systems that can be deployed to production. The two goals for our system, high accuracy and minimal time taken to authenticate are achieved.

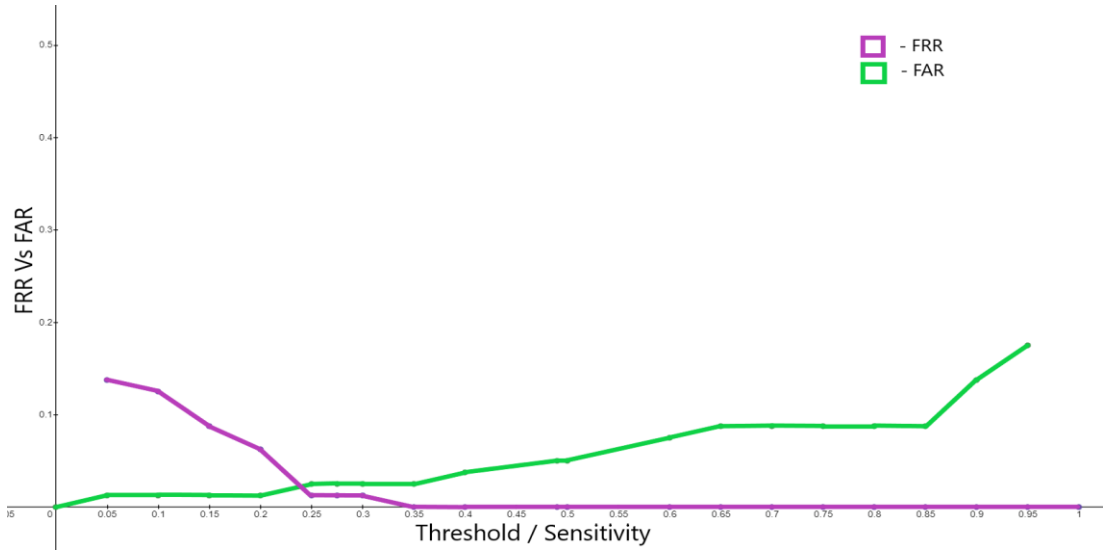

Fig. 4. FRR Vs FAR Graph

EER or equal error rate describes the point where the False Reject Rate (or FRR) and the False Acceptance Rate (or FAR). EER is also known as Crossover Error Rate (or CER). The equal error rate calculated for this system is about 2% (±0.25%). This is much better when compared with existing neural network approaches like [12] and [10]. Table 1 below gives a comparison of EER %

Table 1. EER values when compared with other state-of-the-art signature verification systems

Author	Model	EER%
[8]	HMM	6.90
[13]	DTW	5.50
[9]	HMM	4.83
[21]	DTW	3.38
[11]	MFCC+NN	3.0
[4]	DTW	2.8
[6]	DTW	2.73
[10]	RNN+LNPS	2.37
[22]	EMD	2.13
Proposed system	FCNN	2% (±0.25%)

6. Conclusion

There are not a lot of signature verification systems that use only the SVC2004 dataset with plain deep neural networks for signature verification. We have achieved about 2% (±0.25%) EER which is a very acceptable equal error rate in the signature verification domain. Accuracy of the system is on par with the state-of-the-art signature verification systems on this dataset with faster prediction times, which is a much better option for signature verification systems in a real-time production environment. This can be used as a standalone system or an extension to an existing signature verification system.

Appendix

1. List of features generated

S.no	Feature	S.no	Feature
1	Sample Points (S)	23	Average of Y Velocity
2	Average of X	24	Maximum of X Velocity
3	Average of Y	25	Maximum of Y Velocity
4	Average of Standard Deviation of X	26	X velocity with positive Samples
5	Average of Standard Deviation of Y	27	Y velocity with positive Samples
6	Average Velocity	28	Variance X Velocity
7	Average Acceleration	29	Variance Y Velocity
8	Average Standard Deviation of Velocity	30	Standard deviation X velocity
9	Average Standard Deviation of Acceleration	31	Standard deviation Y velocity
10	Pen Down	32	Median X Velocity
11	Pen Up	33	Median Y Velocity
12	Pen Ratio	34	Correlation of X and Y Velocity
13	Signature Width (W)	35	Mean of X acceleration
14	Signature Height (H)	36	Mean of Y acceleration
15	Width to Height ratio	37	Variance X Acceleration
16	Total Signature Duration	38	Variance Y Acceleration
17	Maximum Pressure	39	Standard deviation X Acceleration
18	Range of Pressure	40	Standard deviation Y Acceleration
19	Sample Points to Signature Width	41	X Local Minima
20	Mean Pressure	42	Y Local Minima
21	Pressure Variance		
22	Average of X Velocity		

References

[1] National Research Council, *Biometric Recognition*. Washington, D.C.: National Academies Press, 2010.

[2] O. Miguel-Hurtado, L. Mengibar-Pozo, M. G. Lorenz, and J. Liu-Jimenez, "On-line signature verification by dynamic time warping and Gaussian mixture models," *Proc. - Int. Carnahan Conf. Secur. Technol.*, no. November, pp. 23–29, 2007, doi: 10.1109/CCST.2007.4373463.

[3] N. N. Liu and Y. H. Wang, "Fusion of global and local information for an on-line signature verification system," *Proc. 7th Int. Conf. Mach. Learn. Cybern. ICMLC*, vol. 1, no. July, pp. 57–61, 2008, doi: 10.1109/ICMLC.2008.4620378.

[4] A. Kholmatov and B. Yanikoglu, "Identity authentication using improved online signature verification method," *Pattern Recognit. Lett.*, vol. 26, no. 15, pp. 2400–2408, Nov. 2005, doi: 10.1016/j.patrec.2005.04.017.

[5] L. Nanni, E. Maiorana, A. Lumini, and P. Campisi, "Combining local, regional and global matchers for a template protected on-line signature verification system," *Expert Syst. Appl.*, vol. 37, no. 5, pp. 3676–3684, May 2010, doi: 10.1016/j.eswa.2009.10.023.

[6] A. Sharma and S. Sundaram, "An enhanced contextual DTW based system for online signature verification using Vector Quantization," *Pattern Recognit. Lett.*, vol. 84, pp. 22–28, 2016, doi: 10.1016/j.patrec.2016.07.015.

[7] J. Fierrez-Aguilar, L. Nanni, J. Lopez-Peñalba, J. Ortega-Garcia, and D. Maltoni, "An On-Line Signature Verification System Based on Fusion of Local and Global Information," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3087, 2005, pp. 523–532.

[8] J. Fierrez, J. Ortega-Garcia, D. Ramos, and J. Gonzalez-Rodriguez, "HMM-based on-line signature verification: Feature extraction and signature modeling," *Pattern Recognit. Lett.*, vol. 28, no. 16, pp. 2325–2334, 2007, doi: 10.1016/j.patrec.2007.07.012.

[9] B. L. Van, S. Garcia-Salicetti, and B. Dorizzi, "On using the Viterbi path along with HMM likelihood information for online signature verification," *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 37, no. 5, pp. 1237–1247, 2007, doi: 10.1109/TSMCB.2007.895323.

[10] S. Lai, L. Jin, and W. Yang, "Online Signature Verification Using Recurrent Neural Network and Length-Normalized Path Signature Descriptor," *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, vol. 1, no. 1, pp. 400–405, 2017, doi: 10.1109/ICDAR.2017.73.

[11] A. Fallah, M. Jamaati, and A. Soleamani, "A new online signature verification system based on combining Mellin transform, MFCC and neural network," *Digit. Signal Process. A Rev. J.*, vol. 21, no. 2, pp. 404–416, 2011, doi: 10.1016/j.dsp.2010.09.004.

[12] S. Meshoul and M. Batouche, "A novel approach for online signature verification using fisher based probabilistic neural network," *Proc. - IEEE Symp. Comput. Commun.*, pp. 314–319, 2010, doi: 10.1109/ISCC.2010.5546760.

[13] D. Y. Yeung *et al.*, "SVC2004: First international signature verification competition," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3072, pp. 16–22, 2004, doi: 10.1007/978-3-540-25948-0_3.
[14] "SVC2004 Competition Public Dataset," 2004. https://www.cse.ust.hk/svc2004/download.html.
[15] S. Bhatia, P. Bhatia, D. Nagpal, and S. Nayak, "Online Signature Forgery Prevention," *Int. J. Comput. Appl.*, vol. 75, no. 13, pp. 21–29, Aug. 2013, doi: 10.5120/13172-0849.
[16] K. S. Manjunatha, S. Manjunath, D. S. Guru, and M. T. Somashekara, "Online signature verification based on writer dependent features and classifiers," *Pattern Recognit. Lett.*, vol. 80, pp. 129–136, 2016, doi: 10.1016/j.patrec.2016.06.016.
[17] U. Jaitley, "Why Data Normalization is necessary for Machine Learning models," 2018. https://medium.com/@urvashilluniya/why-data-normalization-is-necessary-for-machine-learning-models-681b65a05029 (accessed Aug. 12, 2020).
[18] D. Muramatsu and T. Matsumoto, "Effectiveness of pen pressure, azimuth, and altitude features for online signature verification," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4642 LNCS, pp. 503–512, 2007, doi: 10.1007/978-3-540-74549-5_53.
[19] M. Adamski and K. Saeed, "Online signature classification and its verification system," *Proc. - 7th Comput. Inf. Syst. Ind. Manag. Appl. CISIM 2008*, no. 1, pp. 189–194, 2008, doi: 10.1109/CISIM.2008.38.
[20] S. Garcia-Salicetti *et al.*, "Online Handwritten Signature Verification," in *Guide to Biometric Reference Systems and Performance Evaluation*, London: Springer London, 2009, pp. 125–165.
[21] J. M. Pascual-Gaspar, V. Cardeñoso-Payo, and C. E. Vivaracho-Pascual, "Practical On-Line Signature Verification," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5558 LNCS, 2009, pp. 1180–1189.
[22] T. Hafs, L. Bennacer, A. Nait-Ali, M. Boughazi, and A. Bouzid-Daho, "Online signature verification approach using Mellin transform and empirical mode decomposition," in *2017 2nd International Conference on Bio-engineering for Smart Technologies (BioSMART)*, Aug. 2017, pp. 1–6, doi: 10.1109/BIOSMART.2017.8095311.

Authors' Profiles

Snehal Reddy Yelmati has received his Bachelor's degree in Computer Science and Engineering from Osmania University, Hyderabad, and Telangana, India in 2020. He is currently working as a software engineer in India. He has many certifications and specializations from reputed institutions in the field of Machine Learning like Stanford University and Deeplearning.ai.

Jayasree Hanumantha Rao is a Professor at MVSR Engineering College, with 19 years of experience in teaching. Her areas of interests include Network Security, Cryptography, Data Mining and Machine Learning. She has good number of publications in international journals and conferences.

How to cite this paper: Snehal Reddy Yelmati, Jayasree Hanumantha Rao, " Online Signature Verification Using Fully Connected Deep Neural Networks ", International Journal of Engineering and Manufacturing (IJEM), Vol.11, No.5, pp. 41-47, 2021. DOI: 10.5815/ijem.2021.05.04