electronics

MDPI

Article

An Automated Method for Biometric Handwritten Signature Authentication Employing Neural Networks

Mariusz Kurowski *, Andrzej Sroczyński, Georgis Bogdanis and Andrzej Czyżewski

ETI Faculty, Multimedia Systems Department, Gdańsk University of Technology, 80-233 Gdańsk, Poland; a.sr@multimed.org (A.S.); georgis@microsystem.com.pl (G.B.); ac@pg.edu.pl (A.C.)
* Correspondence: markuro@multimed.org

Abstract: Handwriting biometrics applications in e-Security and e-Health are addressed in the course of the conducted research. An automated analysis method for the dynamic electronic representation of handwritten signature authentication was researched. The developed algorithms are based on the dynamic analysis of electronically handwritten signatures employing neural networks. The signatures were acquired with the use of the designed electronic pen described in the paper. The triplet loss method was used to train a neural network suitable for writer-invariant signature verification. For each signature, the same neural network calculates a fixed-length latent space representation. The hand-corrected dataset containing 10,622 signatures was used in order to train and evaluate the proposed neural network. After learning, the network was tested and evaluated based on a comparison with the results found in the literature. The use of the triplet loss algorithm to teach the neural network to generate embeddings has proven to give good results in aggregating similar signatures and separating them from signatures representing different people.

Keywords: biometrics; deep learning; electronic pen; feature extraction; neural networks; online handwritten signature verification

check for updates

Citation: Kurowski, M.; Sroczyński, A.; Bogdanis, G.; Czyżewski, A. An Automated Method for Biometric Handwritten Signature Authentication Employing Neural Networks. *Electronics* **2021**, *10*, 456. https://doi.org/10.3390/electronics10040456

Academic Editor: Marcos Faundez-Zanuy

Received: 8 December 2020
Accepted: 9 February 2021
Published: 12 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The development of biometric verification methods is one of the significant trends in current scientific research. Meanwhile, currently used methods of biometry are known for many problems that limit the scope of their application. Consequently, numerous research teams around the world are looking for new or improved approaches to the acquisition, processing, interpretation, and protection of biometrical data. Similar projects are also carried out at the Gdansk University of Technology in cooperation with the largest Polish bank, particularly those that assume a multimodal approach to biometric authentication. A simplified flowchart of the system visible in Figure 1 shows how each signature is converted from raw signals to a high-dimensional feature representation and how the verification process is organized.

As opposed to the static ones, the handwritten signature's dynamic parameters describe the process of creating the signature, which enables its fuller and more stable representation. The group of dynamic parameters includes the duration of creating a signature, the pen pressure on the surface, the tilt of the pen, and others. These parameters are variable during the signature creation, making it possible to extract the individual features. Thanks to the continuous recording of the signature's parameters, it is much less likely to impersonate another person than using the static (graphical) signature representation. The dynamic signature can serve as a biometric modality that uses the writer for recognition purposes—their individual anatomical and behavioral characteristics. Dynamic signature devices should not be confused with electronic signature capture systems used to capture the signature's graphic image, which is common in locations where merchants are capturing signatures for transaction authorizations.

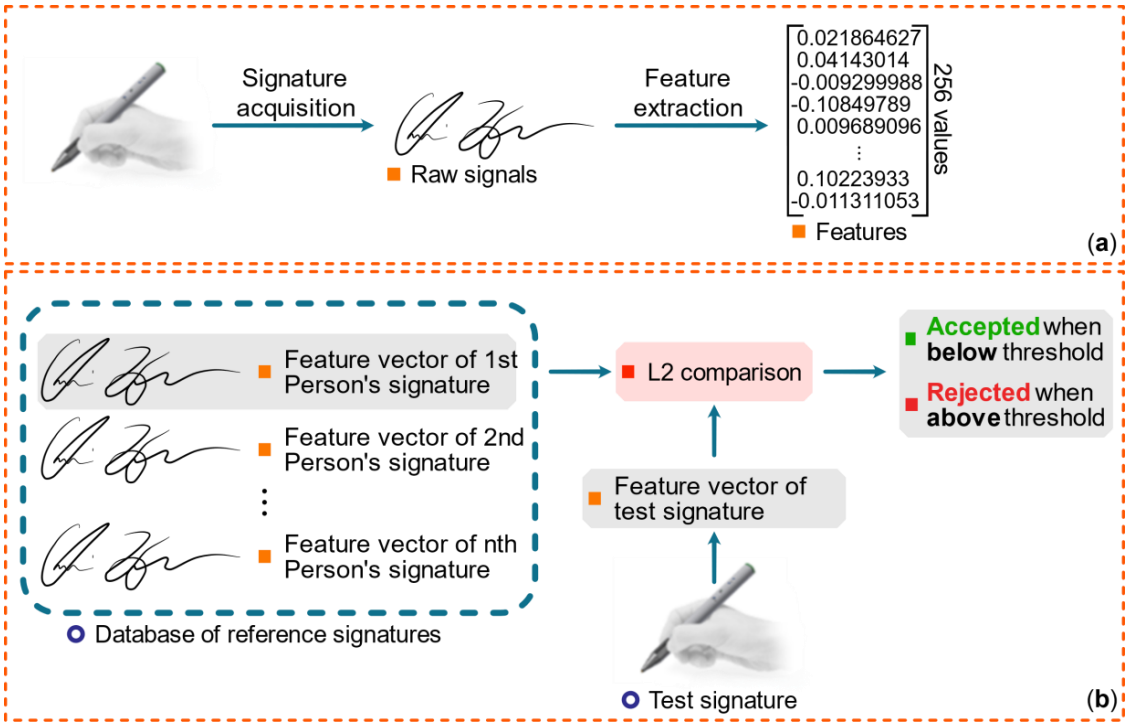

Figure 1. Diagram (**a**) shows the way each signature is processed to obtain its high-dimensional representation. Diagram (**b**) illustrates the verification process, where the signature provided by the client, the test signature, and the reference signature stored in the database are compared.

The primary task is to develop efficient algorithms and identity verification methods based on dynamic analysis of electronically handwritten signatures. In the course of the work, methods of acquiring individual characteristics of signatures were developed and tested for their suitability in the verification process. First, the signatures were obtained using the device designed at the Gdańsk University of Technology (GUT), namely the experimental electronic pen. A large-scale biometric experiment was organized in a major Polish bank using an earlier pen prototype [1,2]. Still, so far, no neural networks have been researched in this context (the scope of hitherto experiments conducted by GUT was limited to overall signature verification using the Dynamic Time Warping algorithm [3,4]). Meanwhile, as it has been known for a long time, neural network methods can help in the detection of many unique characteristics of the signature author, which in turn, is an additional confirmation of identity [5–7].

It is worth adding that forensic science experts commonly use the grapho-analytic analysis (different from "graphological"). Still, its principles have not yet been reflected adequately in the algorithmic analysis of signatures created with digital styluses or machine learning methods. The handwritten signature still remains a legally required confirmation of many types of contracts and transactions. Meanwhile, dynamic signature recognition research does not focus on static or geometric characteristics (i.e., how the signature looks like), but the interest is in dynamic features (how the signature was created). Dynamic signatures use multiple attributes in the analysis of an individual's handwriting. Most of them are active rather than static or geometric, although the latter representations can also be included in the study. Principles of analysis of handwritten signatures by human investigators are standardized in the field of grapho-analytics and forensics. They depend on handwritten text features as slant, commencement and terminating strokes, size or style of words, spacing, pressure, frequency of words, letter proportions, and many others [8]. It is also important to note that in the field of forensics, what separates an untrained person from a skilled investigator is the rate of false positives and the number of cases for which only an inconclusive opinion about the authors' identity may be expressed [9]. Many

systems help to analyze a handwritten text [10–14], which utilizes such experts' features. There are also methods, based on signature analysis, that help diagnose a variety of illnesses, such as Parkinson's and Alzheimer's disease [15–18]. However, the last-mentioned topics are beyond the scope of this paper.

Meanwhile, common dynamic characteristics include velocity, acceleration, timing, pressure, and direction of the signature strokes. Some dynamic signature recognition algorithms incorporate learning functions to account for natural changes or drifts that occur in an individual's signature over time [19]. The characteristics used for dynamic signature recognition are almost impossible to replicate. Unlike a graphical image of the signature, which can be reproduced by a trained human forger, a computer manipulation, or a photocopy, dynamic characteristics are complicated, so they are unique to the individual's handwriting style.

There are many approaches to the classification of a handwritten text, i.e., signature-based person authentication [11]. Examples of such methods are dynamic time warping, Gaussian mixture models, fuzzy modeling, and many others. Meanwhile, deep learning is one of the more promising techniques employed for signature-based verification, which allows for obtaining satisfactory high true-positive and low false-positive rates of detection. Applications of deep learning are mainly based on convolutional, recurrent, and generative adversarial neural networks.

We developed an algorithm based on a convolutional neural network trained with the triplet loss method. This network is then utilized to analyze dynamic features obtained from the biometric pen's sensors, as well as static features (i.e., the shape of the signature), and output a fixed-length representation that could be used to compare and group together signatures. In comparison to analyzing only the shape of a signature, this method has an advantage. It is no longer sufficient to provide a signature with a convincingly similar shape as the way a particular user handles the device while signing contributes substantially to the network's final output.

Many other methods, some of which we discuss briefly, also rely on the computation of signatures' representations using a deep neural network and involve performing the comparison using an additional neural network [20,21]. They are usually composed of multiple feed-forward layers. The approach we used has been previously used to perform verification for face images [22]. We show that it can also be successfully used for the modality of dynamic signatures.

An overview for the rest of the article is as follows: In Section 2, we describe the biometric pen, a novel device designed specifically for biometric verification, and the data acquired by its many sensors. Section 3 details the method we use to extract features from signatures to determine their similarity. Section 3.1 contains information about the dataset we gathered and used for training a deep convolutional neural network to extract signatures' features. Section 3.2 contains the implementation details of our neural network architecture, and Section 3.3 concerns the training process. Finally, in Section 4, we present quantitative results of our work and discuss them in Section 5, comparing them to other authors' methods.

2. Biometric Pen

The dynamic biometric signature is well embedded in law at the international level; there are two corresponding documents: ISO/IEC FCD 19794-7: Information technology—biometric data interchange formats, Part 7: Signature; and ISO/IEC WD 19794-11: Information technology—biometric data interchange formats, Part 11: Signature Processed Dynamic Data. In this document, one can find that a valid biometric signature is a data series—a collection of points comprising of timestamp, 2D position, velocity, acceleration, pen pressure, and pen angle. However, the abovementioned general requirements do not define signal analysis methods that should be used to perform the signature verification (recognition). Meanwhile, proper recording and signature verification require the use of compatible devices, allowing for the registration of biometric data with a sufficient time

resolution and with enough pressure levels. These requirements are described in ISO/IEC FDIS 19794-7 standards for a biometric signature.

An electronic pen was developed at the Gdańsk University of Technology to satisfy the above demands. Components of the device and its interaction with a tablet can be seen in Figure 2.

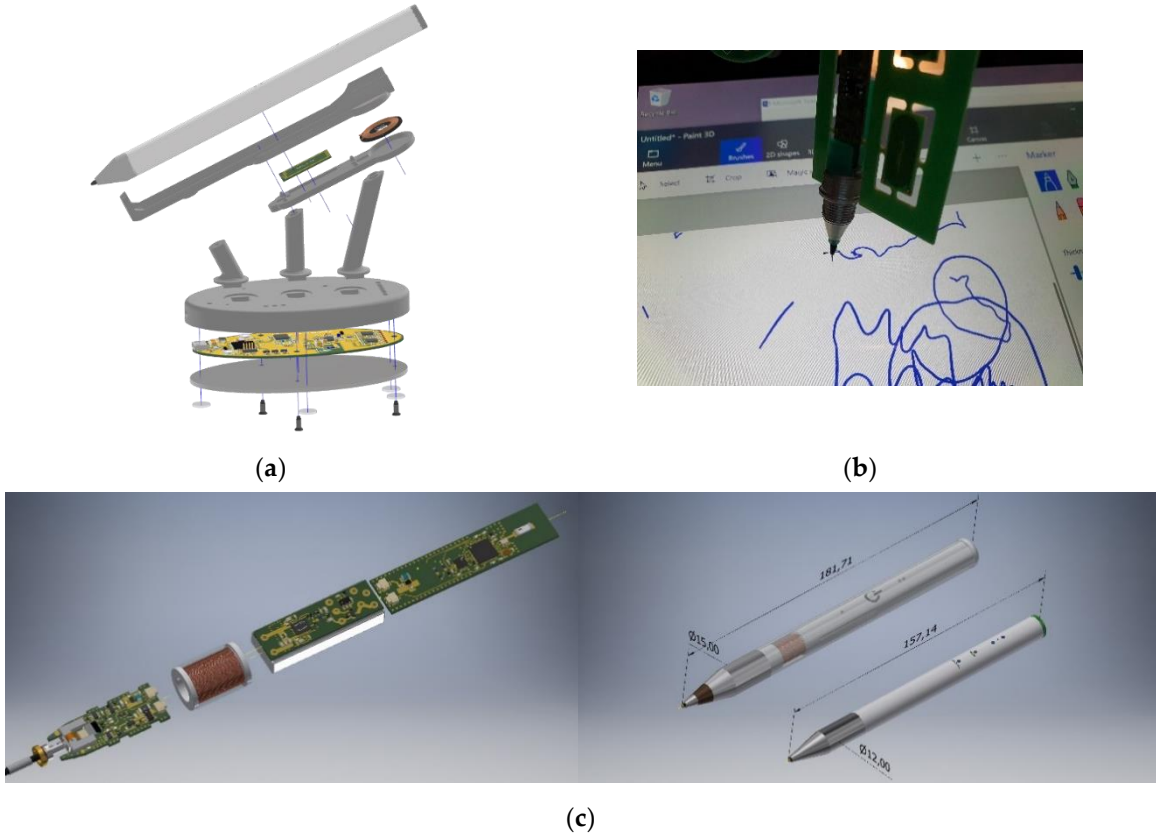

Figure 2. Construction details of the pen and its wireless charger (**a**); the pen prototype while interacting with a tablet screen (**b**); developed biometric pen for data handwritten signature recognition (**c**).

The pen has the following resources and properties:

- works with a variety of computer and mobile phone screens
- has a pressure sensor
- 6-axis gyroscope-accelerometer
- 3-axis inclinometer
- 2 MEMS (Microelectromechanical systems) microphones
- miniature speaker
- BLE (Bluetooth Low Energy) radio interface
- specialized module with built-in Bluetooth antenna
- built-in rechargeable battery
- wireless charging
- over-the-air software update
- passive NFC (Near-Field Communication) tag for pen authorization in the system (pairing)
- active mode and status detected by an accelerometer (no external buttons)
- signaling lights illuminated through the casing (no holes in the casing)
- USB-2.0 (Universal Serial Bus) interface for system communication and power supply
- cordless pen charger
- NFC transceiver to communicate with the pen (authorization data in BLE)

Each signature collected via the biometric pen is represented as a data structure containing samples sent from the device's sensors and recorded screen coordinates of the pen tip. Data from sensors are acquired with a constant sampling rate of 50 S/s. Pen tip coordinates are recorded as Windows cursor coordinates with a mean sampling rate of 80 S/s. The exact amount of coordinate samples may vary even among signatures that have the same duration because samples are not collected if the pen tip is not touching the tablet surface or not changing position. Table 1 shows the biometric pen's data to the receiving computer and recorded by the computer.

Table 1. Data received from the biometric pen and captured from screen coordinates. One sample sent from the pen contains 7 values, while the sample recorded from the screen contains 2 values.

Sample Sent from Sensors	Screen Coordinate
Accelerometer x component	x coordinate of the pen tip
Accelerometer y component	y coordinate of the pen tip
Accelerometer z component	-
Gyroscope x component	-
Gyroscope y component	-
Gyroscope z component	-
Pen pressure	-

A typical signature created in around 6.6 s could contain 230 samples received from the biometric device and 388 samples recorded as screen coordinates. Depending on a person and circumstances, a signature might take a varying amount of time to complete, thus producing a different number of samples for both data streams each time. Figure 3 demonstrates example data acquired for a single signature.

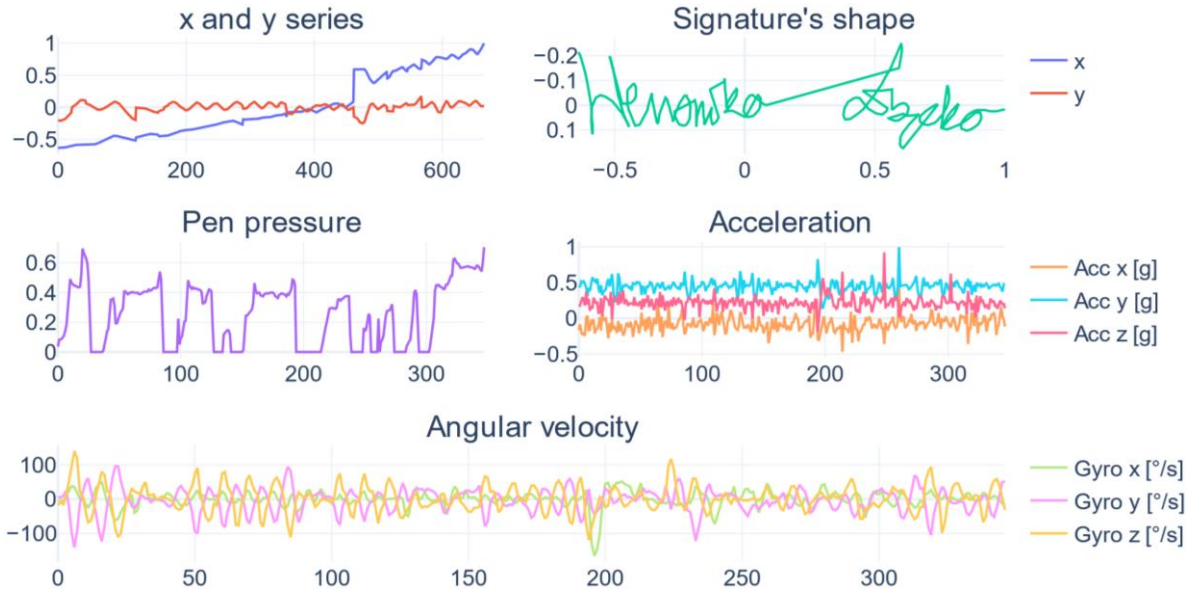

Figure 3. Example of raw signals' values for subsequent samples for a single signature.

3. Methods

The algorithm presented in this section utilizes a convolutional neural network to extract meaningful features from signatures supplied for biometric authentication. In a typical usage scenario, one would keep a database of users with one or more signatures assigned to each of them. Whenever a need for authentication arises, the system would ask a person claiming to be a registered user to provide their signature (a test signature). The neural network would then process this signature to obtain a fixed-length 256-dimensional

embedding. Afterward, the system sends it to the authentication server. The server would fetch the signature assigned for that user (already in embedding form) from the database. Having both the test embedding and the reference embedding, one can calculate the Euclidean distance between them. As the network is designed to constrain an embedding onto a unit hypersphere, such a distance is bound to lie anywhere in a $\langle 0, 2 \rangle$ interval. The server would authenticate the user successfully, should the calculated distance be lower than a predefined threshold. It should be noted that the method we describe in this work assumes each registered user has just one signature associated with them in the database, and a single test signature is used for comparison. Our method, however, may also be extended to make use of an additional number of signatures stored in the database, which can contribute to higher recognition (authentication) rates.

As the method we develop is designed for user authentication, not for classification, we are only concerned with finding a function mapping signature samples to a new embedding space. Points corresponding to the same person should lie close together, and different people's points should be far apart. This way, it is easy to determine if two arbitrarily chosen signatures are similar, i.e., correspond to the same person.

In the triplet loss algorithm, a neural network is optimized using triplets. Each triplet consists of two similar samples (and therefore should lie together in the embedding space) plus one sample that is not (should be pushed away). The network is trained to ensure that every triplet does not violate the triplet loss condition. The triplet loss formula is defined as follows:

$$\mathcal{L} = max(0, \parallel A - P \parallel_2 - \parallel A - N \parallel_2 + \alpha) \quad (1)$$

where A (anchor) and P (positive) represent signatures that should be close in the embedding space, and N (negative) stands for a signature that is different from both A and P. It should be noted that A, P, and N are feature vectors extracted by the neural network, not raw signatures. The loss function is designed to ensure that for every triplet, the distance between A and N is greater than the distance between A and P by at least the margin α. Figure 4 demonstrates the relationship between triplet elements and shows various triplet types.

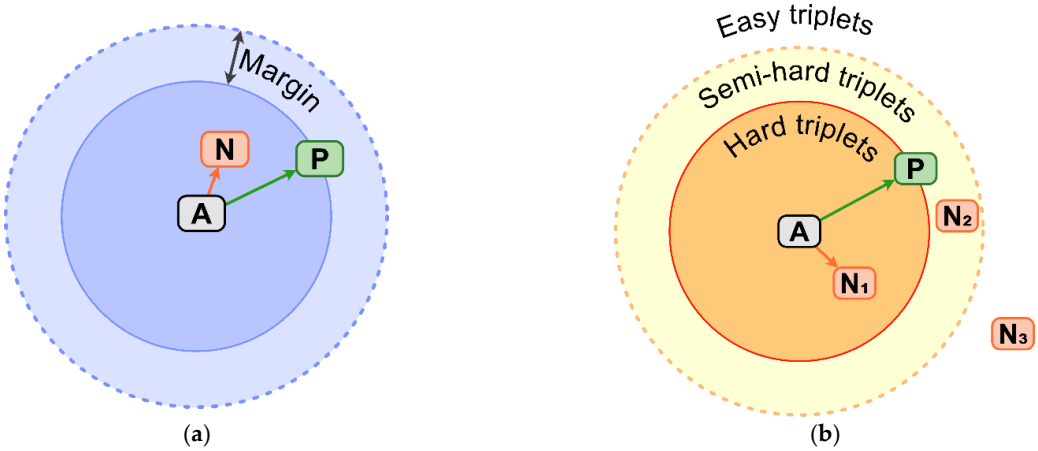

Figure 4. Diagram (**a**) demonstrates a triplet violating the triplet loss condition. A (anchor) and P (positive) represent feature vectors of similar signatures, and N (negative) represents feature vectors of a signature different from both A and P. The triplet loss algorithm aims to ensure that the distance from A to N (AN distance) is greater than the AP distance for every triplet, at least by the margin. Diagram (**b**) illustrates three possible types of triplets, depending on the distance between A and N points.

Several other deep metric learning methods have been proposed that depend on similar principles. The Contrastive Loss [23] algorithm uses pairs, instead of triplets, to group together similar samples and push away samples belonging to other classes. There are also methods utilizing not pairs or triplets but a greater number of points per training

example, such as Quadruple Loss [24] and N-Pair Loss [25]. Magnet Loss [26], on the other hand, penalizes the degree of overlap between clusters.

3.1. Dataset Structure and Acquisition

To train and evaluate the proposed algorithm's quality, we gathered multiple signature samples, using an early version of the biometric pen, from 2264 participants. Each person provided, on average, five signatures during a single session. In total, 10,622 signatures were obtained. Biometric system safety needs to accept the natural variability occurring in one's handwriting style while also rejecting uncharacteristic deviations that might indicate an attack attempt.

We required that each person provide successive signatures in a single style characteristic for that participant's signature-making style. Consequently, we found multiple recurring types of mistakes across the dataset. Examples of such errors are shown in detail in Figure 5.

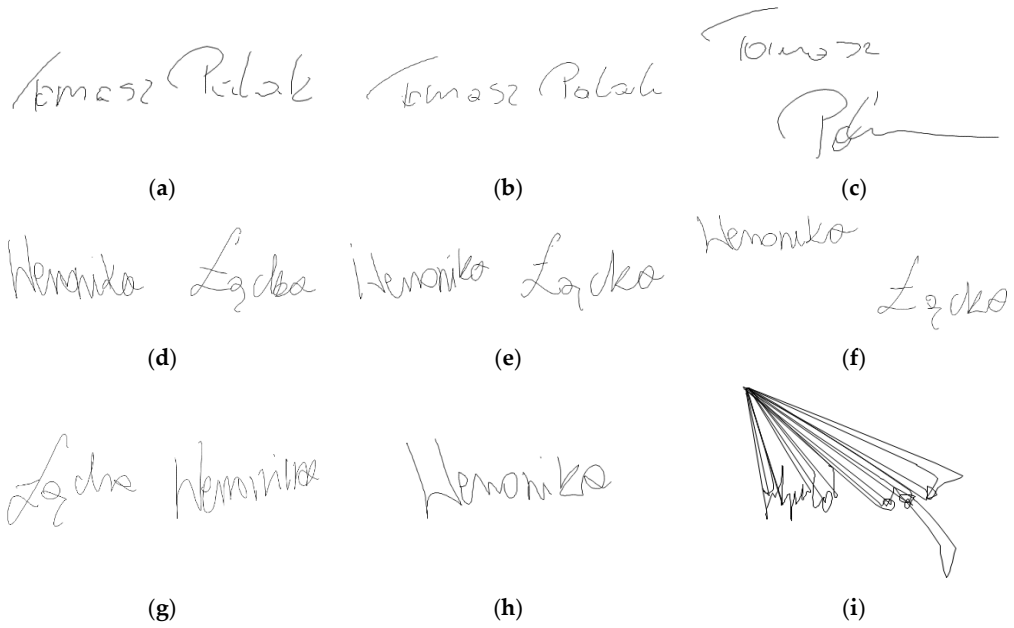

Figure 5. Selected mistake types found in the dataset. Person 1 provided two correct signatures (**a**) and (**b**), but in (**c**) wrote their surname below the first name instead of to the right. Similarly, Person 2 correctly provided signatures (**d**) and (**e**) but wrote their surname in a different place in (**f**) and swapped first name's and surname's positions in (**g**). Lastly, they only provided the first name, without a surname, in (**h**). Example (**i**) from Person 3, on the other hand, shows anomalies introduced by erroneous touch events registered during signature acquisition.

The presence of such mistakes throughout the dataset necessitated the need to perform a manual cleanup to guarantee that each person's signatures are consistent and can be treated as multiple occurrences of one's signature style. Despite most of such cases being successfully eliminated, a small margin of them persisted in the dataset. In spite of that, the neural network training managed to converge. We mention this issue as we think it is essential to keep in mind that dataset acquisition is just as crucial as a thoughtful design of the network architecture or training algorithm.

For evaluation purposes, we gathered an additional smaller set of 244 signatures collected from 10 people. Each time we gathered signatures, we asked a pair of participants to sign on average 10 times. While the first participant was signing, the other one studied the first one's way of signing and practiced forging their signature. After the first participant had finished, the second one tried to forge the first participant's signatures on average 10 times. After that, they switched roles. In this way, we generated, on average, 40 signatures per pair of participants—10 reference signatures and 10 forgery

attempts of each person. Figure 6 shows an example of skilled forgery cases acquired in the additional dataset.

(**a**) (**b**)

Figure 6. Image (**a**) depicts the first participant's signature, while (**b**) shows the other one's forgery attempt.

For both datasets, the signatures were acquired on a capacitive touch screen with a biometric pen. Each participant was sitting with the screen lying flat in front of them.

3.2. Selection of Neural Network Architecture

Even though fully convolutional neural networks do not restrict input data to a fixed size, to generate embeddings of the same size, we adjust collected data so that the number of samples used is constant at all times. To do this, we resample each data series to 512 samples via linear interpolation. After this step, a single signature contains 512 samples, each having 9 values, which adds up to 4608 values per signature.

After resampling and before the signature can be used as an input for the neural network, it undergoes further preprocessing, consisting of coordinate system conversion and the normalization of samples gathered from the biometric pen and screen.

- Screen coordinate samples conversion—we calculate the mean position from the screen coordinates so that the center of the coordinate system lies in the middle of the signature. Afterward, each (x, y) sample is converted to polar coordinates (r, θ). The normalization step divides radius r by the maximum value for the signature so that it remains in the range $\langle 0, 1 \rangle$. We also normalize azimuth angle θ from range $\langle -\pi, \pi \rangle$ to $\langle -1, 1 \rangle$.
- Biometric pen samples conversion—We convert accelerometer and gyroscope vectors from cartesian $[x, y, z]$ vectors to spherical $[r, \theta, \varphi]$ coordinates. We divide the radius by the maximal possible acceleration/angular velocity, which was defined to be ± 2 g and $\pm 2000^\circ/\text{s}$, respectively, in order to keep its value in the range of $\langle 0, 1 \rangle$. The azimuth angle θ is normalized from the range of $\langle -\pi, \pi \rangle$ to $\langle -1, 1 \rangle$, while the polar angle φ is from the range of $\langle 0, \pi \rangle$ to $\langle 0, 1 \rangle$. Pen pressure is also constrained to range $\langle 0, 1 \rangle$.

After this procedure, each data series contains a fixed amount of normalized samples in the shape of 512 features per 9 channels. The signature in this form can then be fed into the neural network to produce a 256-dimensional embedding. We tried many other architectures with different amounts of layers and with different hyperparameters. The one that yielded the best results can be seen in Figure 7.

As training the network with the triplet loss algorithm takes a long time to finish, we modified the network's structure mostly by trial and error instead of using automatic methods, such as random search or grid search.

In place of regular convolutional layers, we use depthwise separable convolutional layers, which allow for substantial speedup without sacrificing quality, as explained in [27]. Except for the one directly before the L2 normalization layer, every such layer utilizes the PReLU (Parametric Rectified Linear Unit) [28] activation function. This way, the activation function's slope can be optimized and set for each neuron independently, unlike in Leaky ReLU (Rectified Linear Unit), for instance.

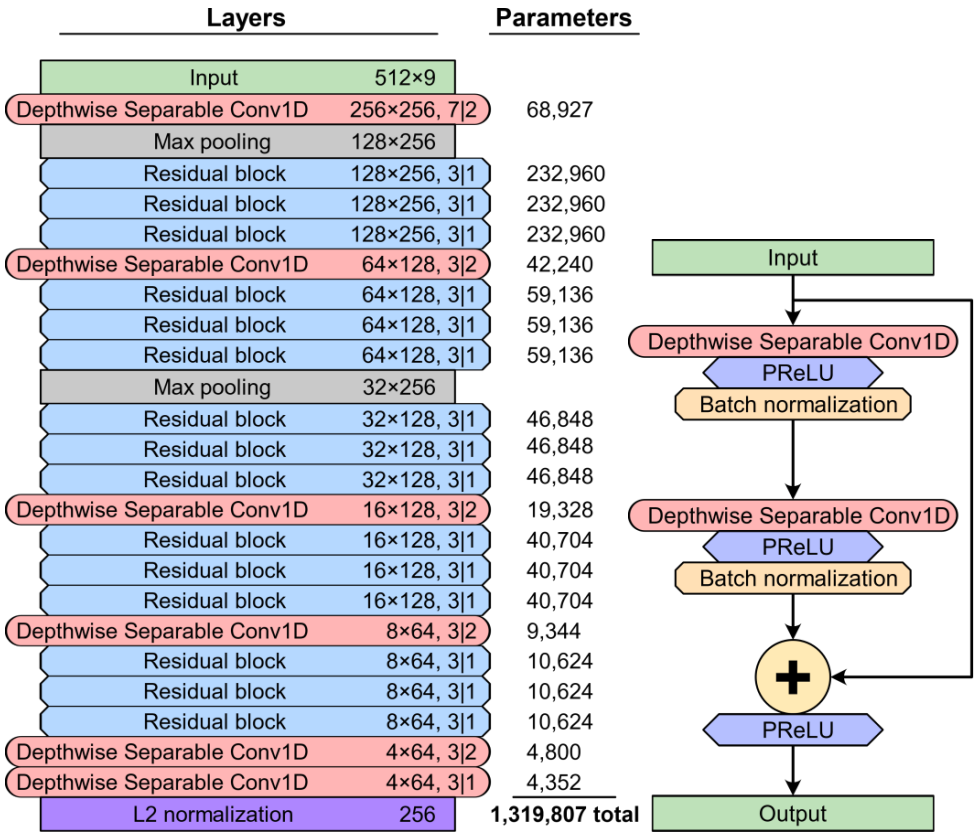

Figure 7. Diagram (**a**) depicts consecutive layers of the proposed neural network architecture. Each layer should be read according to the following schema: [layer_name output_shape filter_size | stride]. The last Depthwise Conv1D layer has an output shape of 4 features, 64 channels, filter size of 3, and stride 1. Diagram (**b**) shows the schema of the Residual block used as a layer type in (**a**).

The last layer, called L2 normalization, flattens the output of its preceding layer to a 256-dimensional vector and projects it onto the surface of a unit hypersphere. It is done by calculating the length of this vector, using the Euclidean metric, and dividing all its components.

3.3. Network Training Algorithm

To train the network, we used the triplet loss algorithm. It is a well-established method that finds its uses in many areas, including image retrieval, object recognition, and biometric verification [22,29,30]. This approach especially lends itself to the problem of biometric verification, as there exist a considerable amount of classes (each person could be considered a separate class), and their total number is not known at the training time. When using the network in a production environment, new people will be registered, therefore introducing new classes. The crucial thing to consider is that the algorithm is expected to work without the need to retrain it after registering new people.

We divide the dataset into 2 subsets—a training set consisting of 2064 clients and a validation set having 200 clients. The training set is randomly sampled with a uniform distribution to generate a batch containing 256 triplets during the training phase. We employ semi-hard negative mining [30] to ensure that every triplet contributes to the gradient update and maintains training stability. Due to this, we also noticed that the training procedure slows down dramatically in the later stages, as it becomes increasingly challenging to find semi-hard triplets.

We used a fixed margin of 0.25 to define the triplet loss formula [30] and Adam optimizer [31] with a learning rate set to 0.01. Neural network accuracy and loss values during training can be seen in Figures 8 and 9, respectively. We define 1 epoch as 5 network's weights updates, as it is infeasible to iterate over the set of all possible triplets.

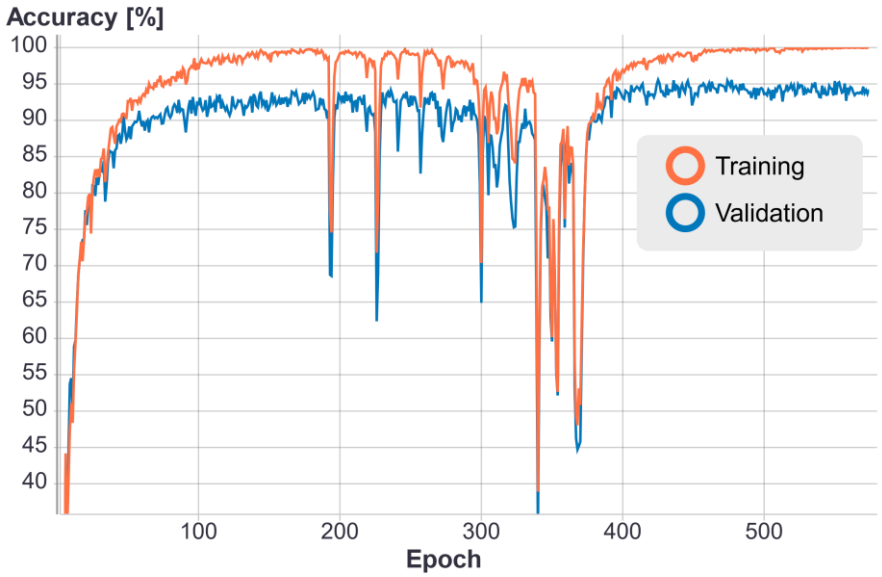

Figure 8. Neural network accuracy during training. Accuracy is defined as the percentage of triplets satisfying the triplet loss margin condition in a randomly sampled batch of 2048 triplets.

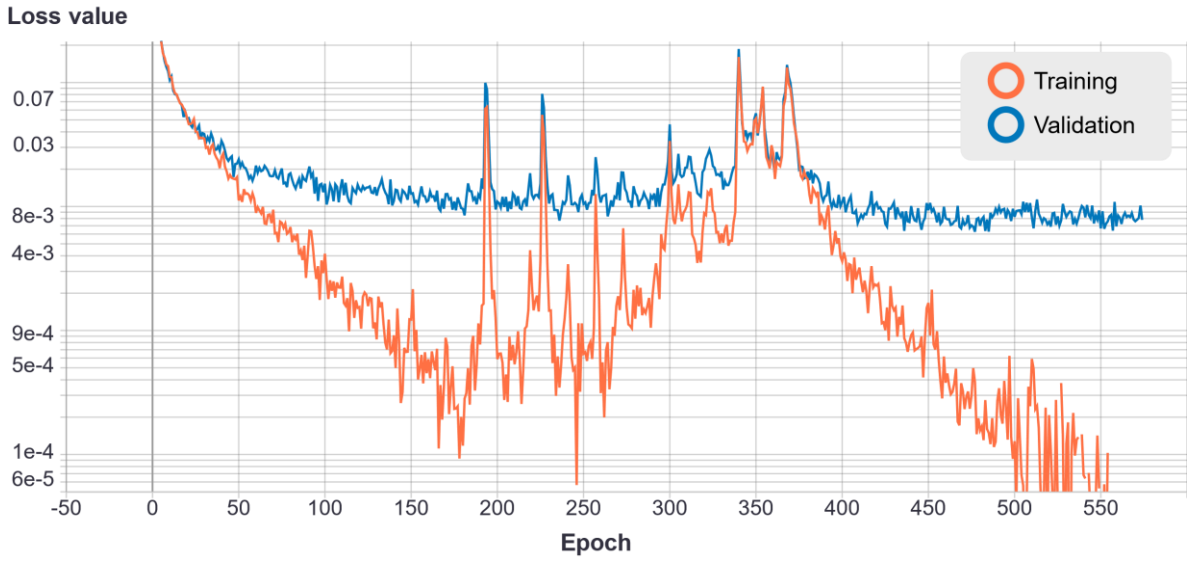

Figure 9. Neural network loss value during training. Estimated loss is calculated for a randomly sampled batch of 2048 triplets.

The total time to train the network was 26 h and 17 min using a PC with an Nvidia Geforce RTX 1080 Ti GPU.

4. Results

This section describes a trained neural network's evaluation results using a validation set to measure its quality, considering signatures that were not taking part in the training. This set contains 200 clients and 1043 signatures in total. We carried out an experiment in which we compare pairs of signatures for every client. If one picks two signatures

belonging to the same person, it is expected that the Euclidean distance between them should be below a certain small threshold. In addition to that, pairs of signatures belonging to two different clients were also compared, yielding a distance above that threshold.

In a single test, we make 2306 comparisons of signature pairs of a single person and 539,721 comparisons of signature pairs where signatures belong to different clients. We repeat this test for 200 thresholds uniformly distributed over the range $\langle 0, 2 \rangle$, as distances between two random embeddings are constrained to this range. Our network scored 5.94% EER (equal error rate) for threshold 1.055. Figure 10 depicts the ROC (receiver operating characteristic) curve where each point corresponds to a single test for a certain threshold.

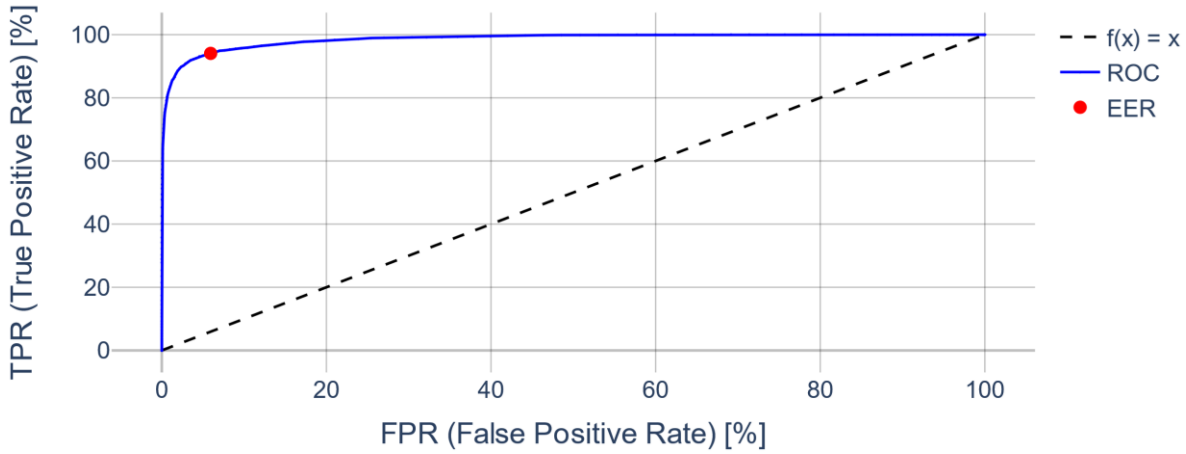

Figure 10. Receiver operating characteristic (ROC) curve for neural network evaluated on a validation set consisting of 200 clients with 1043 signatures in total.

To further ensure that our evaluation method is not dependent on validation set composition (i.e., which samples are used for evaluation) and would generalize well on new data, we carried a test akin to cross-validation. As it is very time-consuming to train a neural network model from scratch, we do it only once. Still, we perform tests described earlier in this section multiple times for randomly sampled subsets of the validation set.

We randomly divide the validation set into 5 subsets of 40 clients. For each subset, we calculate the EER. This procedure is then repeated 20 times to obtain 100 EER values. This way, we obtained a mean EER value of 5.77% and a confidence interval of $\left[\; 5.519\%, \quad 6.024\% \; \right]$ with a standard 95% confidence level. The standard deviation for the EER was 1.265 with minimum and maximum values of 2.69% and 9.508%, respectively.

To present the results graphically, we used the t-SNE (t-Distributed Stochastic Neighbor Embedding) algorithm [32], which can be seen in Figure 11. We used a subset of the validation set containing 100 clients to generate the visualization. Each signature manifests as a color dot. Signatures of a single person are connected with a line. It can be observed that the same person's signatures lie close together and clump into tight groups, separating from other groups, as is expected. Whenever signatures of the different color group together is a sign of incorrect neural network prediction or a consequence of the presence of mistakes, as was described earlier in Section 3.1. Dataset structure and acquisition.

Some signatures' representations are distant in the graph because they are mislabeled samples in the dataset. Such samples would be incorrectly shown with the wrong person's color and connected by a line to the wrong sample group. It is one of the reasons why we mention errors in the dataset in Section 3.1. Due to the big size of the dataset, it is difficult to perform a complete cleanup of mistakes present. In the case of mislabeled signatures, one would have to manually compare pairs of signatures, which is infeasible due to the number of comparisons that would have to be performed. On the other hand, heuristic methods by definition do not guarantee that a cleanup performed with such techniques is correct. Some feature vectors could also be outliers and thus be distant in the graph.

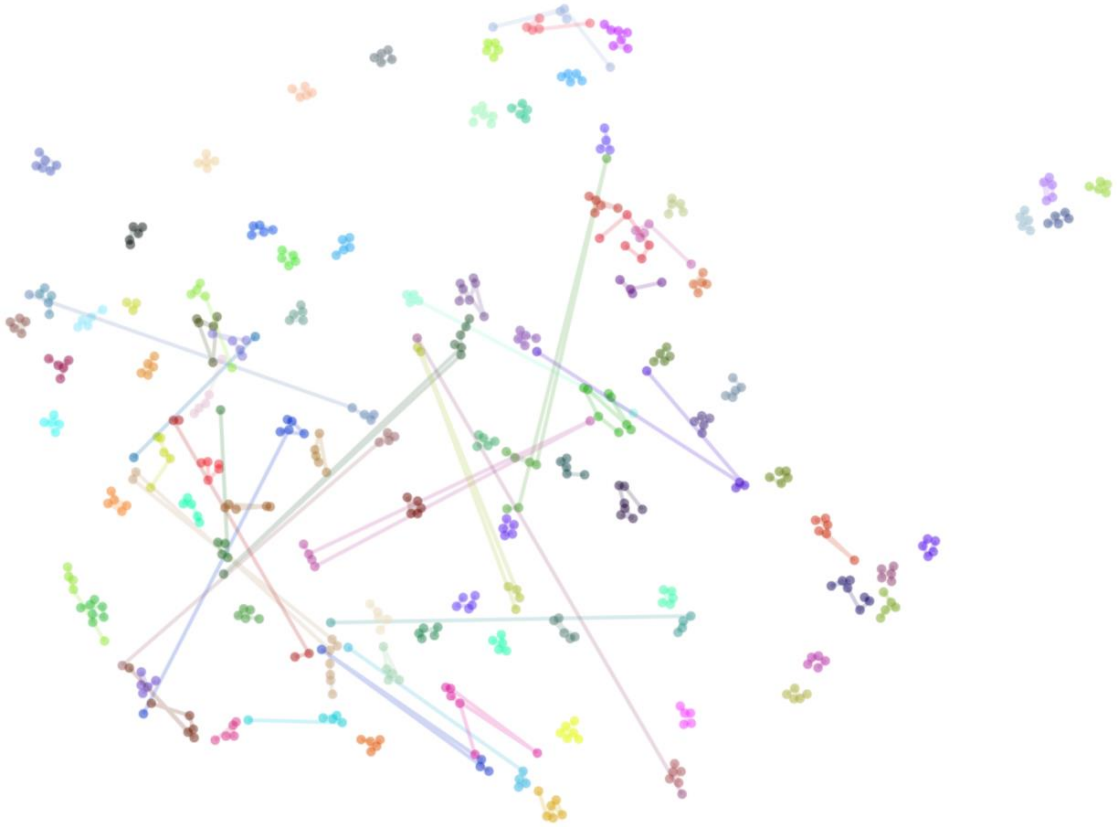

Figure 11. t-Distributed Stochastic Neighbor Embedding (t-SNE) visualization of a subset of the validation set containing 100 clients. Each signature is represented as a color dot. Signatures belonging to a single person are connected with a line.

To emphasize the separation of distances obtained for pairs of similar signatures (of the same person) and dissimilar ones (from different clients), we calculated histograms for a single test using the whole validation set, which can be seen in Figure 12. A "Genuine" histogram was created using 2306 comparisons and an "Impostor" with 539,721.

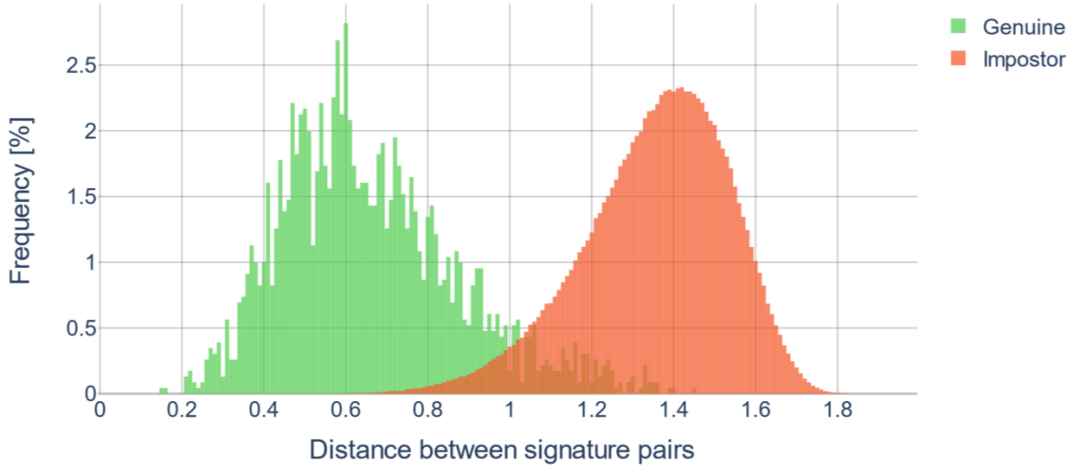

Figure 12. "Genuine" histogram shows the distribution of distances for pairs of signatures belonging to a single person, while "Impostor" depicts the distance distribution for pairs of signatures of different clients.

The "Genuine" histogram has a mean distance of 0.6653 with a confidence interval $\left[\ 0.6564,\quad 0.6742\ \right]$ and a standard deviation of 0.2173. "Impostor" has a mean distance of 1.3601 with a confidence interval $\left[\ 1.3596,\quad 1.3605\ \right]$ and a standard deviation of 0.178.

Lastly, we show the separation using a boxplot in Figure 13. The solid line inside each box is the mean value, and the dotted line shows the median. Box bottom and top boundaries are defined by the 1st and 3rd quartile, and the whiskers are 1.5*IQR (interquartile range) apart from the boundaries.

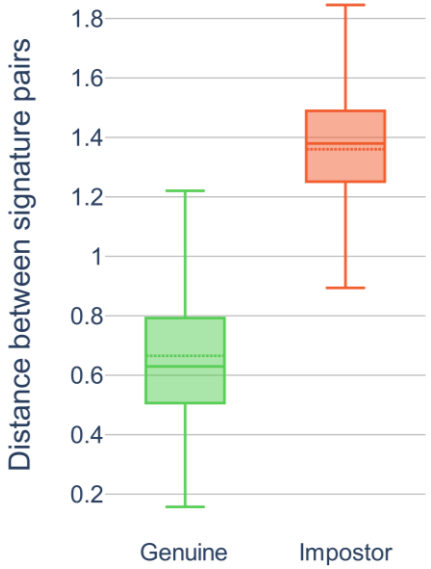

Figure 13. Boxplot of distances between signature pairs for "Genuine" and "Impostor" case.

We also evaluated our method on the smaller set we described at the end of Section 3.1. to check the neural network's ability to differentiate between genuine signatures of a particular person and forgery attempts of an attacker trying to mimic that person's shape and dynamic features. We calculated distances between each participant's pairs of signatures. Additionally, distances between genuine and skilled forgery signatures have been calculated. Afterward, all calculated distances were compared with the threshold. Our neural network achieved 11.114% EER in this test. The ROC curve can be seen in Figure 14.

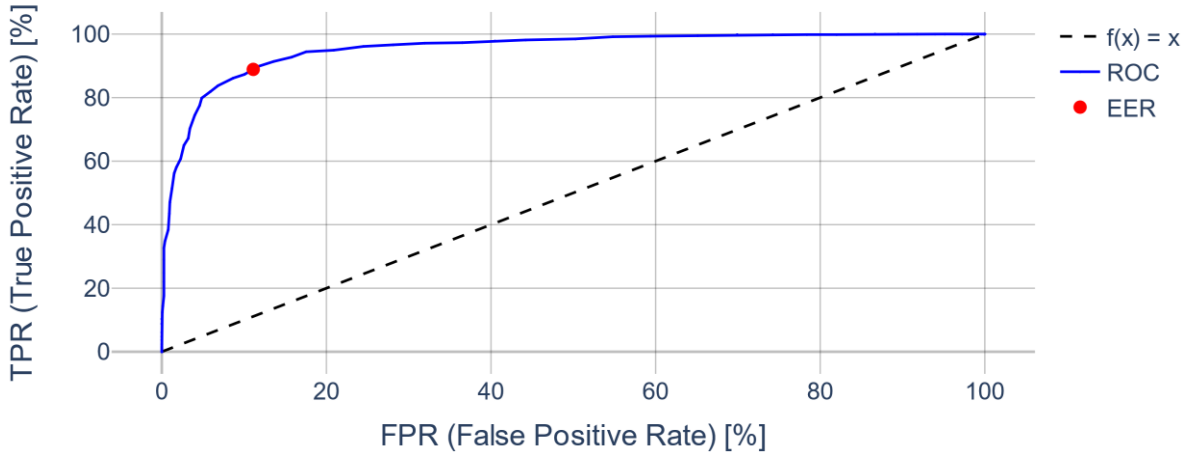

Figure 14. ROC curve for neural network evaluated on 244 signatures of 10 people containing skilled forgery attempts.

The distribution of distances for pairs of genuine signatures and pairs consisting of one genuine and one forged signature can be seen in Figure 15.

A boxplot showing the separation between "Genuine" and "Impostor" scenarios is additionally shown in Figure 16.

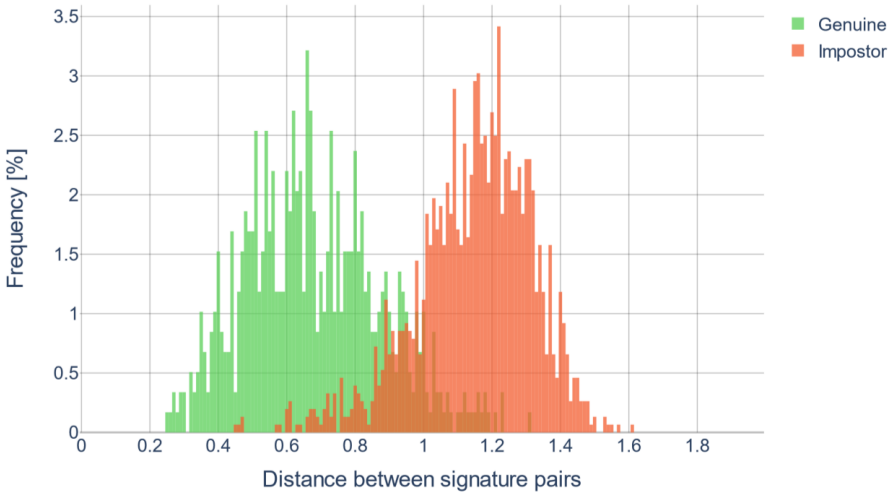

Figure 15. "Genuine" histogram shows the distribution of distances for pairs of signatures belonging to a single person. Simultaneously, "Impostor" depicts the distance distribution for pairs consisting of one genuine and one forged signature.

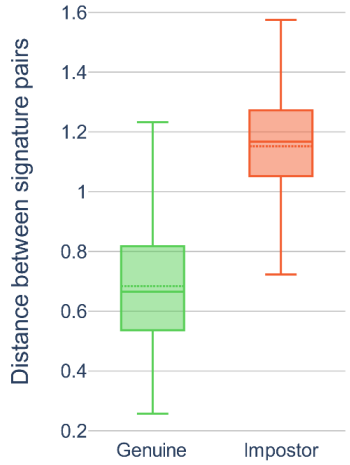

Figure 16. Boxplot of distances between signature pairs for "Genuine" and "Impostor" case evaluated on a small set containing skilled forgery attempts.

5. Discussion

We demonstrated that it is feasible to construct an algorithm to perform user authentication employing a neural network to extract meaningful features from handwritten signatures. We used a device designed specifically for biometric verification; however, this method is general enough to be used in other areas, where different hardware is employed to acquire samples. In the following subsections, we compare our results with other authors' findings. Lastly, we indicate areas we would like to focus on in future work.

5.1. Performance Concerning the State of the Art

In this subsection, we explore other approaches to online signature verification compared to the method presented in this article.

Tolosana et al. [20] employ a Siamese architecture based on bidirectional LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) neural networks. The authors used a BiosecurID database [33] of comparable size to ours, consisting of 11,200 signatures. The recurrent network extracts meaningful features from both tested and registered signatures, and after that, a separate multilayer perceptron network calculates the similarity between features of both signatures. In our method, a single neural network is used to extract features, which can then be directly compared using the L2 norm. This distance comparison

is a counterpart to the multilayer perceptron used. The authors achieve a comparable 5.28% EER using the BLSTM (Bidirectional Long Short-Term Memory) network trained on random forgery cases while comparing one test signature to one registered signature. However, our experiments report EER values as low as 2.69%. To perform a fair comparison, both methods ought to be implemented on the same dataset to determine which method is more applicable.

Ahrabian and BabaAli [21] propose a method where an LSTM autoencoder transforms signatures into a fixed-length latent space representation and uses a Siamese network to determine how similar the tested signatures are. The authors achieve 8.65% EER on the SigWiComp2013 Japanese dataset.

Schroff et al. [22] use the triplet loss algorithm to directly teach a neural network how to encode face images to a fixed-length 128-dimensional representation. The authors use an approach similar to the one presented in this article, even though it concerns a different modality.

A comparison of the approach presented in this article with the state-of-the-art methods can be found in Table 2.

Table 2. Evaluation results obtained on various datasets by state-of-the-art methods in comparison to results for our method.

Dataset	Evaluated on	Method	EER
BiosecurID	Random forgery	BLSTM trained only on random forgery [20]	5.28%
BiosecurID	Skilled forgery	BLSTM trained only on random forgery [20]	15.31%
SigWiComp2013	Random and skilled forgery	Autoencoder + Siamese [21]	8.65%
Our	Random forgery	Deep convolutional network trained with triplet loss	Best: 2.69% Mean: 5.77% Worst: 9.508%
Our	Skilled forgery	Deep convolutional network trained with triplet loss	11.114%

5.2. Future Work

An interesting problem is comparing a "wet" signature (written on paper) with a signature placed on an electronic screen's surface. In order to perform research on this subject, an extensive set of data should be collected, including wet signatures generated with a regular pen on paper and so-called in vitro signatures created with an electronic stylus. Hence, the task is to determine the relationship between these two types of signatures to determine whether the signatures made with styluses have enough standard parameters for recognition and the nature of those parameters. Such signatures can be compared because each literate person has preconceived graphology parameters of their signature. These parameters may not be valid for electronic signatures. However, we presume that there exists a relation between wet and in vitro signatures. Based on the collected samples of different signatures from many individuals, we will investigate the correlation and relationships between signatures.

We want to investigate our method's performance on skilled forgery attempts using a neural network trained using random forgery and skilled forgery training examples. Due to our dataset not having skilled forgery cases, we could not perform this kind of analysis so far.

So far, we only considered how our method behaves when only a single reference signature is assigned to each user. We believe one could achieve better results if more signatures are used and would like to emphasize how to do this in the future.

Another area we would like to focus on is signature aging. As pointed out by Galbally et al. [13], aging in the signature trait is a specifically user-dependent effect. Moreover, it does not seem to depend on the signature's complexity, as it affects both simple and complex ones. The main conclusion is that, in general, a user affected by signature aging will perform poorly regardless of the method used. The simplest solution could be to give a user a way to register again, perhaps after successful authentication with an additional biometric modality.

We would like to explore how other deep metric learning methods could be used further to improve the quality of such a biometric verification system. One of the significant flaws of the triplet loss algorithm used in our method is that it becomes progressively slower as the training procedure continues. Lastly, we want to investigate how one could integrate this approach to dynamic signature verification as a part of a comprehensive multimodal biometric system.

6. Conclusions

We presented an efficient user verification approach based on a dynamic signature using a specialized device designed for this task. Our method utilizes the triplet loss algorithm to train a neural network model that can be used to extract meaningful features from signatures, in the form of fixed-length embeddings that group well for signatures of a single person and separate from other signature groups pertaining to other clients. The deep metric learning method has been successfully applied to a biometric verification scheme based on face images [22]. We showed that this approach could also be used in the modality of dynamic signatures, using our dataset collected with a device specifically designed for biometric verification. We achieved a mean of 5.77% EER in our evaluation method for random forgery attempts and 11.114% EER for skilled forgery using a neural network trained only on random forgery cases.

Author Contributions: Conceptualization, M.K., A.C. and A.S.; methodology, M.K. and A.C.; software, M.K.; validation, M.K.; formal analysis, A.C.; investigation, M.K., A.S. and A.C.; data curation, M.K. and G.B.; writing—original, M.K. and A.C.; draft preparation, M.K., A.C. and A.S.; writing—review and editing, M.K. and A.C.; visualization, M.K., A.S. and G.B.; supervision A.C. and G.B.; project administration, A.C.; funding acquisition, A.C. and G.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded from the budget of project No. POIR.01.01.01-0092/19 entitled: "BIOPUAP—a biometric cloud authentication system", currently financed by the Polish National Centre for Research and Development (NCBR) from the European Regional Development Fund.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

BLE	Bluetooth Low Energy
BLSTM	Bidirectional Long Short-Term Memory
EER	Equal Error Rate
FPR	False-Positive Rate
GRU	Gated Recurrent Unit
IQR	Interquartile Range
LSTM	Long Short-Term Memory
MEMS	Microelectromechanical systems
NFC	Near-Field Communication
PReLU	Parametric Rectified Linear Unit
ReLU	Rectified Linear Unit
ROC	Receiver Operating Characteristic
TPR	True-Positive Rate
t-SNE	t-Distributed Stochastic Neighbor Embedding
USB	Universal Serial Bus

References

1. Szczuko, P.; Czyżewski, A.; Hoffmann, P.; Bratoszewski, P.; Lech, M. Validating data acquired with experimental multimodal biometric system installed in bank branches. *J. Intell. Inf. Syst.* **2019**, *52*, 1–31. [CrossRef]
2. Czyzewski, A.; Hoffmann, P.; Szczuko, P.; Kurowski, A.; Lech, M.; Szczodrak, M. Analysis of results of large-scale multimodal biometric identity verification experiment. *IET Biom.* **2019**, *8*, 92–100. [CrossRef]
3. Lech, M.; Czyzewski, A. A handwritten signature verification method employing a tablet. In Proceedings of the Signal Processing—Algorithms, Architectures, Arrangements, and Applications Conference Proceedings, SPA, Poznan, Poland, 21–23 September 2016; IEEE Computer Society: Washington, DC, USA, 2016; pp. 45–50.
4. Lech, M.; Czyzewski, A. Handwritten Signature Verification System Employing Wireless Biometric Pen. In *BT—Intelligent Methods and Big Data in Industrial Applications*; Springer: Cham, Switzerland, 2019; pp. 307–319.
5. Huber, R.; Headrick, A. *Handwriting Identification*; CRC Press: Boca Raton, FL, USA, 1999.
6. Impedovo, D.; Pirlo, G. Automatic signature verification: The state of the art. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2008**, *38*, 609–635. [CrossRef]
7. Harralson, H.H. *Developments in Handwriting and Signature Identification in the Digital Age*; Taylor and Francis: Milton Park, UK, 2014; ISBN 9781315721736.
8. Stewart, L.F. The Process of Forensic Handwriting Examinations. *Foresic Res. Criminol. Int. J.* **2017**, *4*, 139–141. [CrossRef]
9. Bird, C.; Found, B.; Rogers, D. Forensic document examiners' examiners' skill in distinguishing between natural and disguised handwriting behaviors. *J. Forensic Sci.* **2010**, *55*, 1291–1295. [CrossRef] [PubMed]
10. Guarnera, L.; Farinella, G.M.; Furnari, A.; Salici, A.; Ciampini, C.; Matranga, V.; Battiato, S. *GRAPHJ: A Forensics Tool for Handwriting Analysis BT—Image Analysis and Processing, ICIAP 2017*; Battiato, S., Gallo, G., Schettini, R., Stanco, F., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 591–601.
11. Diaz, M.; Ferrer, M.A.; Impedovo, D.; Malik, M.I.; Pirlo, G.; Plamondon, R. A perspective analysis of handwritten signature technology. *ACM Comput. Surv.* **2019**, *51*, 1–39. [CrossRef]
12. O'Reilly, C.; Plamondon, R. Development of a Sigma–Lognormal representation for on-line signatures. *Pattern Recognit.* **2009**, *42*, 3324–3337. [CrossRef]
13. Galbally, J.; Martinez-Diaz, M.; Fierrez, J. Aging in Biometrics: An Experimental Analysis on On-Lline Signature. *PLoS ONE* **2013**, *8*, e69897. [CrossRef] [PubMed]
14. Tolosana, R.; Vera-Rodriguez, R.; Fierrez, J.; Ortega-Garcia, J. Reducing the template ageing effect in on-line signature biometrics. *IET Biometrics* **2019**, *8*, 422–430. [CrossRef]
15. Pirlo, G.; Diaz, M.; Ferrer, M.A.; Impedovo, D.; Occhionero, F.; Zurlo, U. Early diagnosis of neurodegenerative diseases by handwritten signature analysis. In Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Cham, Switzerland, 2015; Volume 9281, pp. 290–297.
16. Bou, A.; Fischer, A.; Plamondon, R. Omega-Lognormal Analysis of Oscillatory Movements as a Function of Brain Stroke Risk Factors. In Proceedings of the 17th Biennial Conference of the International Graphonomics Society, Pointe-à-Pitre, Guadeloupe, 21–24 June 2015.
17. Bidet-Ildei, C.; Pollak, P.; Kandel, S.; Fraix, V.; Orliaguet, J.P. Handwriting in patients with Parkinson disease: Effect of l-dopa and stimulation of the sub-thalamic nucleus on motor anticipation. *Hum. Mov. Sci.* **2011**, *30*, 783–791. [CrossRef] [PubMed]
18. Wang, Z.; Abazid, M.; Houmani, N.; Garcia-Salicetti, S.; Rigaud, A.S. Online signature analysis for characterizing early stage Alzheimer's disease: A feasibility study. *Entropy* **2019**, *21*, 956. [CrossRef]
19. Sae-Bae, N.; Memon, N. Online signature verification on mobile devices. *IEEE Trans. Inf. Forensics Secur.* **2014**, *9*, 933–947. [CrossRef]
20. Tolosana, R.; Vera-Rodriguez, R.; Fierrez, J.; Ortega-Garcia, J. Exploring Recurrent Neural Networks for On-Lline Handwritten Signature Biometrics. *IEEE Access* **2018**, *6*, 5128–5138. [CrossRef]
21. Ahrabian, K.; BabaAli, B. Usage of autoencoders and Siamese networks for online handwritten signature verification. *Neural Comput. Appl.* **2019**, *31*, 9321–9334. [CrossRef]
22. Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, PR, USA, 7–12 June 1997; pp. 815–823.
23. Hadsell, R.; Chopra, S.; LeCun, Y. Dimensionality reduction by learning an invariant mapping. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 17–22 June 2006; IEEE Computer Society: Washington, DC, USA; Volume 2.
24. Ni, J.; Liu, J.; Zhang, C.; Ye, D.; Ma, Z. Fine-grained patient similarity measuring using deep metric learning. In *Proceedings of the International Conference on Information and Knowledge Management, Proceedings*; Association for Computing Machinery: New York, NY, USA, 2017; Volume Part F1318.
25. Sohn, K. Improved deep metric learning with multi-class N-pair loss objective. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016.
26. Rippel, O.; Paluri, M.; Dollar, P.; Bourdev, L. Metric learning with adaptive density discrimination. In Proceedings of the 4th International Conference on Learning Representations, ICLR 2016—Conference Track Proceedings, San Juan, Puerto Rico, 2–4 May 2016.

27. Guo, J.; Li, Y.; Lin, W.; Chen, Y.; Li, J. Network decoupling: From regular to depthwise separable convolutions. In Proceedings of the British Machine Vision Conference, BMVC 2018, Newcastle, UK, 2–6 September 2018.
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; IEEE Computer Society: Washington, DC, USA, 2015; pp. 1026–1034.
29. Lagunes-Fortiz, M.; Damen, D.I.; Mayol-Cuevas, W. Learning discriminative embeddings for object recognition on-the-fly. In Proceedings of the IEEE International Conference on Robotics and Automation, Montreal, QC, Canada, 20–24 May 2019; IEEE: New York, NY, USA, 2019; Volume 2019, pp. 2932–2938.
30. Kaya, M.; Bilge, H.Ş. Deep metric learning: A survey. *Symmetry* **2019**, *11*, 1066. [CrossRef]
31. Kingma, D.P.; Ba, J.L. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015 Conference Track Proceedings, San Diego, CA, USA, 7–9 May 2015.
32. Van Der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
33. Fierrez, J.; Galbally, J.; Ortega-Garcia, J.; Freire, M.R.; Alonso-Fernandez, F.; Ramos, D.; Toledano, D.T.; Gonzalez-Rodriguez, J.; Siguenza, J.A.; Garrido-Salas, J.; et al. BiosecurID: A multimodal biometric database. *Pattern Anal. Appl.* **2010**, *13*, 235–246. [CrossRef]