UNIVERSIDAD CARLOS III DE MADRID

EXTENDED SUMMARY

SIGNATURE FORGERS:

TOOL FOR ACQUISITION OF BIOMETRIC SIGNATURE SAMPLES

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

Author: Laura Benito Martín

Tutor: Judith Liu Jiménez

Leganés, 17 de Junio de 2015

1. Introduction

Nowadays, the amount of information that we share and the number of transactions that we execute are really large. The necessity of a secure way to be authenticated in the digital world has been increasing substantially over the years. People have the right to feel safe in their lives. Therefore any study that may improve their security is highly worthwhile. But to improve security it is essential to fight attacks, and in order to fight attacks, the experts shall be able to reproduce those attacks as to test if the new improvements resist them.

Signature Forgers is a C# application for Windows that trains forgers to forge an original signature. In the application, a forger will pass through different training levels, in which he/she will be given different amounts of information about the genuine signature. By doing this, it will be possible to study the performance of a forger depending on the amount of knowledge he/she has about the signature being forged. All the information and parameters saved during the training can be used to do research and, therefore, make progress in the field of detecting forgeries. However, such research is out of the scope of this project.

2. State of the art

2.1. Biometrics

Biometrics is a field studying the physical or behavioural features that are unique to each human being and hence used for authentication.

A biometric system has two stages: enrolment or registration, in which user information is saved in the system, and identification, in which the system scans the biometric feature of a user and compares it with the previously saved data. After the comparison, if there is a highly probable match between the user scanned and the user registered, the user will be verified as the one who registered in the first place.

There are plenty of modalities of biometrics depending on what it is being measured. Physical features, also known as "static", are those characteristics that physiologically differentiate one person from another, for example: face, fingerprints, iris, ear or DNA. Behavioural features, also known as "dynamic", are those that are not part of our body but depend on the way we act, for example: signature, gait or voice, although the latter also depends on physical characteristics.

In order to useful for biometrics, a feature has to fulfil several requirements:

- **Universality**: It must be a characteristic that all future prospective users of the system have
- **Permanence**: It must remain sufficiently time invariant.
- **Uniqueness**: It must be unique to each individual. Two people's features have to be distinguishable each from another.
- **Performance**: It must be possible to quickly acquire an accurate sample of the feature.
- **Collectability**: It must be easily and quantitatively measured.
- **Circumvention**: The possibility of cheating the system with fraudulent methods must be low enough.
- **Acceptability**: The measurement of this feature must be relatively accepted by population.

The social acceptance of biometric sensors is increasing gradually as people are becoming willing to use devices that help them access their data easily, but without losing their privacy. The convenience of accessing information from a smartphone by just putting a finger on a button is greater than, for example, by writing a secret password that, in addition, might be stolen or forgotten. This ease of use results in benefits not only from a user perspective, to be happy with the product purchased, but also from a business perspective, as companies will be able to guarantee the security of the product they offer.

Security has always been an important matter. However, when speaking about biometrics, the importance of a good security increases considerably. The biometric information of a person is something that will be valid for the person's whole life, which means that if someone steals anyone else's features, the former person wholly loses the possibility of using it further in the future. And that is one of the biggest disadvantages of biometric systems. Nonetheless, there are a lot of advantages, such as the convenience and ease with which users are authenticated when biometrics is used, and the highly difficult ways to falsify this kind of systems.

Biometrics is a field that has a lot to offer to society. It makes the protection of private information easier, while offering guarantee and ease. Consequently, all the research aimed at improving the velocity, reliability or performance of these technologies will result in a more secure society.

2.2. Signature Recognition

Signature recognition is a behavioural type of biometric. There are two different ways to understand these systems: static and dynamic. Static signature, also called offline, refers to the final image of a signature. Dynamic signature, also called online, refers to how the signature is made, the whole process of signing: the different coordinates in time, the velocity, the pressure of the pen, etc.

Dynamic signature provides more security than static ones, however, it is also more difficult to implement, as it needs the use of special sensors (a digital tablet and a stylus), in order to register all the different parameters. Despite the necessity of these sensors, this is a pretty accepted modality, probably because people are used to sign all its important documents using pen and paper in order to identify themselves.

Once acquired, the samples need to be processed with an algorithm that can state whether the signature saved has been made by the person originally registered in the database. There are plenty of algorithms for this purpose, but one of the most relevant is Dynamic Time Warping (DTW).

This algorithm measures the similarity between two temporal sequences that may vary in velocity or time. It was originally developed for voice recognition, so the best example of this works is related to this field: Two people can say out loud the same sentence, however, they might have different intonation, one might talk faster than the other, or may change the emphasis put on the words. Consequently, the sounds will be, distributed differently throughout time, even though it is the same sentence. The aim of this algorithm is to remove those temporal differences between common points, so that both sentences can be aligned and, therefore, be compared in a more optimal way.

3. Design and implementation

The application developed is a tool that allows forgers to learn and practise how to sign like another person, becoming thereby able to mimic the behaviour of the signature, not only the final image. The purpose of this application is to study how accurate signatures are depending on the amount of information provided to the user. It uses DTW algorithm for the comparison of original and forged signatures. It needs a specific device to acquire the signatures: a tablet. The tablet chosen is Wacom STU-500.

It had several designing requirements:

- Training parameters are configurable.
- It registers users and saves their personal information.
- It acquires signatures from the users
- Training has 10 levels
- Forgers have to do the training level by level
- It saves all the information related to the forgery process.

There are two different types of users: genuine users, who are the ones providing the database with real signatures, and forger users, who are the ones trying to falsify the genuine signatures from the database. Besides, there is also a third type of user: an administrator, who will be in charge of the configuration of the parameters.

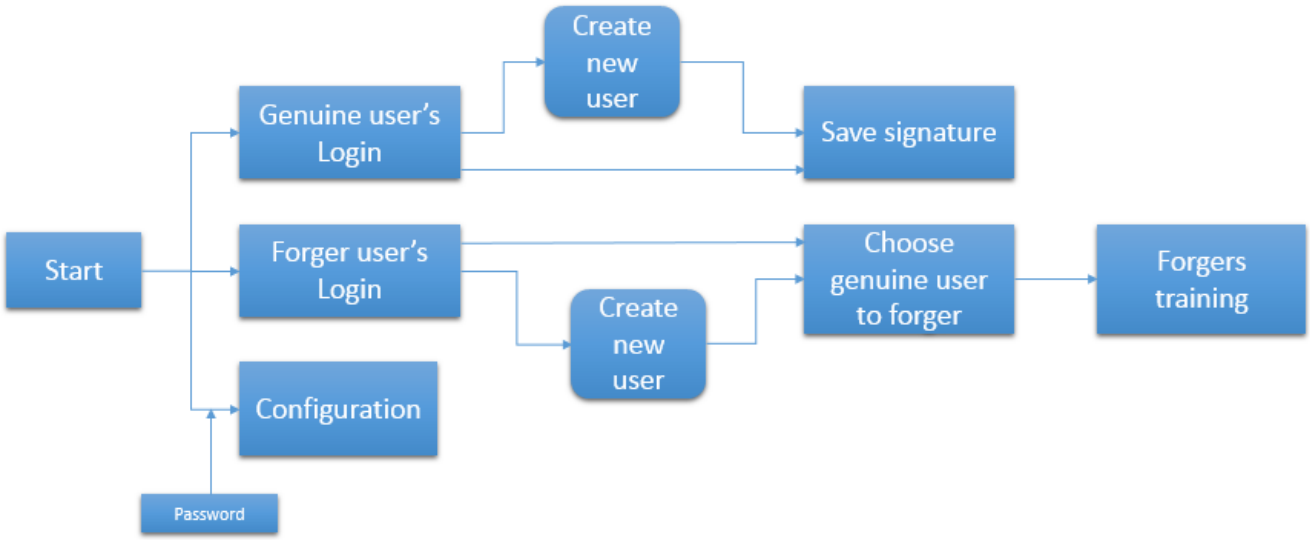

Figure 1 – Functionality diagram

The application can be divided in four different parts: configuration, user registration, genuine signature acquirement and forger training.

3.1. User registration

Both user types follow the same registration process in which they are asked to fulfil a form of their personal information (name, surname, age, National ID number, phone number, email and whether they are right- or left-handed). The application checks that all this information meets

the format needed and, in the case of National ID number, it also checks if the ID letter is the one resulting from the numbers registered.

Before the information is saved, the application prints a document that informs the users about their rights regarding personal data. Users must sign this document in order to finish the registration process. If the users decide not to sign, no personal data will be saved in the application database. Once the registration process has finished, the application assign an ID number to the users.

3.2. Genuine signature acquirement

Once registered, the genuine users have to save their signature in the database as many times as the application requires. By using the digital tablet (Wacom STU-500 model), genuine users will be allowed to sign as many times as they need and choose whether to save the signature or to delete it and try again.

Every time a user signs in the tablet, several data such as date, time, user, sample number or the number of times the sample has been deleted, is saved in a log that can be studied afterwards.

Once users have finished saving all the signatures requested, they can either save more samples or finish the process. If they decide to exit, they will nonetheless be able to enter again the application in the future by using the ID assigned or the National ID number, and increase the amount of signatures of the database.

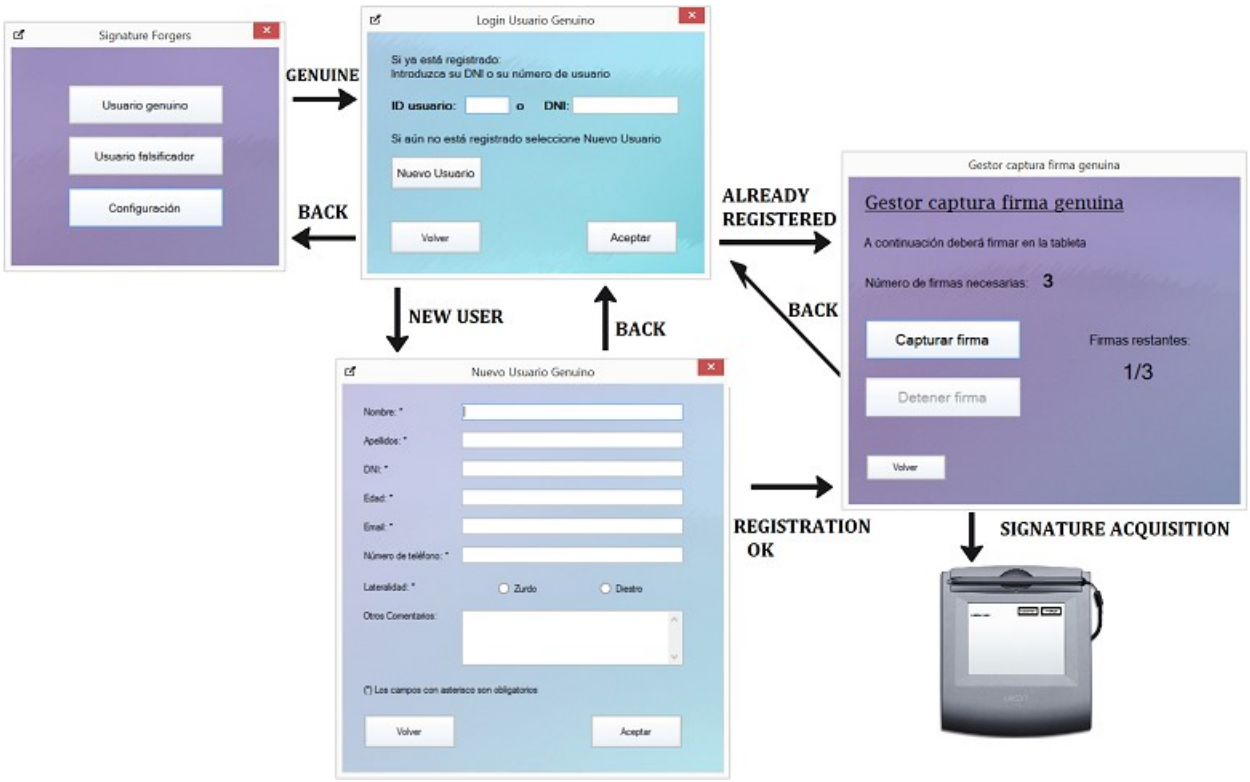

Figure 2 – Window diagram of genuine users functionality

3.3. Forger training

If registered as forger user, users will be able to see a list of all genuine users susceptible to being forged. When one of the genuine users is chosen, the training begins. Forger users have to complete 10 levels of training along which the application will provide them with more information about the signatures as the training levels increases. They have to save a specified number of correct samples in each level. DTW algorithm will determine which samples are correct and which are not depending on whether they overpass a settled threshold or.

Whenever users sign in the tablet, all the information of the signature is saved in a training log that can be studied after the training to see how the user has evolved. In this log there is information about: date, hour, forger user ID, genuine user that is being forged ID, level, number of sample, number of times the user deleted a sample, the absolute result of the algorithm and the final status of the signature (0 if it passed the threshold, 1 if it did not, and 2 if it is a deleted signature).

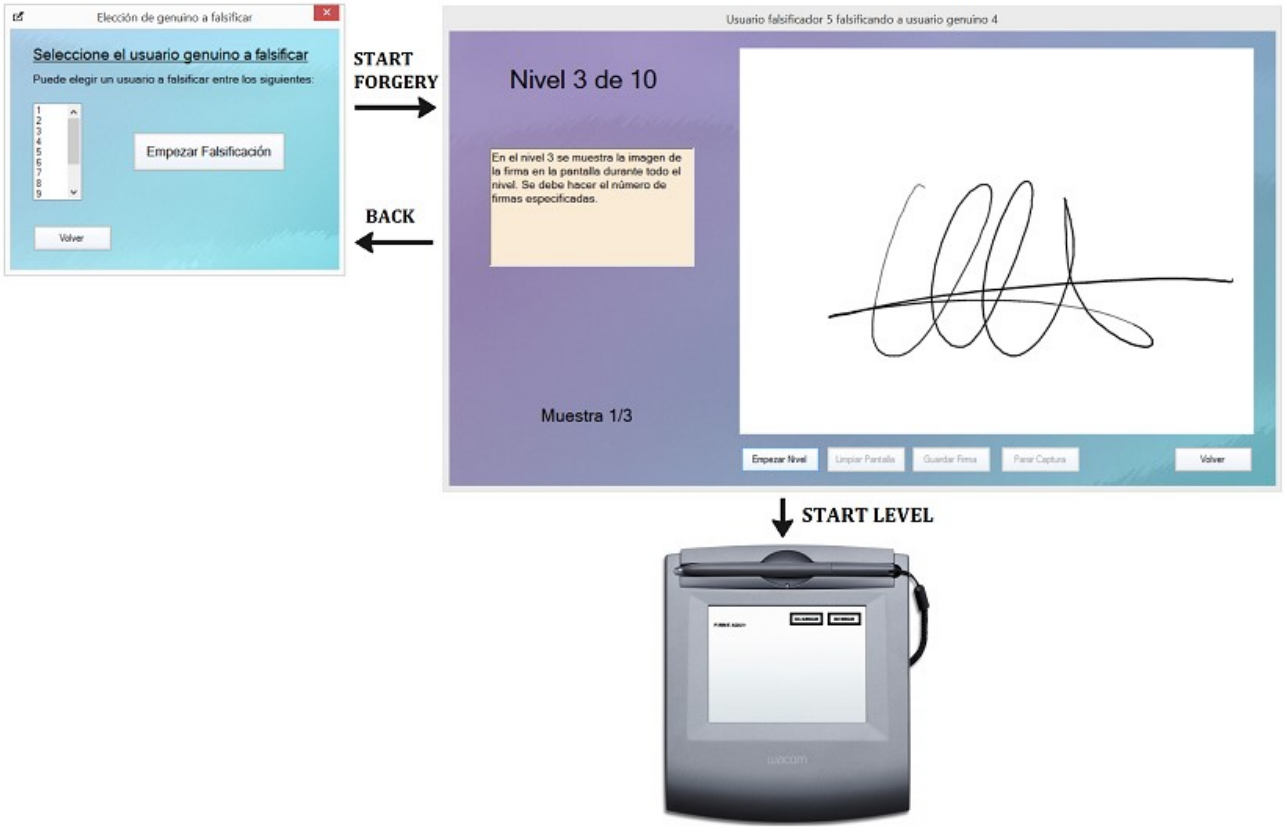

Figure 3 – Window diagram of forger training

3.4. Configuration

This part of the application can only be accessed by authorized people, as it requires the introduction of a password. Here, all available parameters can be configured with the required values. Those parameters are:

- Number of samples that genuine users must save

- Number of samples that forger users must save per level
- Number of signatures required for the algorithm pattern
- Waiting times between levels 8-9, and 9-10
- Algorithm Threshold in the different levels

3.5. Levels

During the training, forger users have to pass 10 levels in order to finish the forgery. Algorithm threshold is only fixed from levels 4 to 7.

Training starts at level 1, in which users have zero information about the signature, so that they just sign whatever they consider as many the times as needed and then pass to the next level.

In level 2, the image of the signature is shown in the computer for 5 seconds and then disappears. Users try to replicate it the times required and then pass to the next level.

In level 3, the image of the signature is shown in the computer permanently, so users can see it as long as they need and then sign the times requested to pass to the next level.

In level 4, the image of the signature is shown in the tablet permanently, so users can sign on the real signature. In this level users already need to pass a certain threshold of similarity with the original signature. Users can go to the next level if they pass that threshold with all the signatures requested.

In level 5, a video of how the signature was performed is shown once in the computer. Users need to pass a threshold in this level too.

In level6, there is a complete video player so that the user can pause, replay or slow down the video of the signature as many times as needed. Also in this level users have to pass a threshold.

Level 7 combines both levels 4 and 6, which means that users have a video player of the signature in the computer and the image of the signature in the tablet. This is the level in which users have the greatest amount of information about the signature. It is also the last one to have a threshold.

The last 3 levels do not show any information of the signature, as users are supposed to have already learnt how to reproduce it. These levels do not have any fixed threshold either because users will not know anymore whether the signature made is good or not, it is just registered in the logs.

Right after finishing level 7, users can start level 8, in which they will have to save the amount of samples required. After finishing level 8, users have to wait an established amount of time before they start level 9, by default 60 minutes.

After waiting the time specified, users can start level 9. They will save the number of samples requested. When level 9 is finished, users have to wait an established amount of time, longer than before, to start level 10, by default 1 day.

Finally, after waiting, users will sign in level 10 as in the latest levels. When users finish this level, training for the current genuine user will be over, and therefore it will no longer appear in the list of genuine users available for that forger user.

4. Tests

During the development, the application has been tested every time a new functionality was added to check whether it worked properly or not. Nevertheless, once the application was completely finished, a general test was made in order to check that the behaviour of the program was appropriate. For that purpose, a fictitious database was created, with genuine users signing randomly, and forger users falsifying those signatures.

Registration of all those users worked properly without errors, genuine signatures captured were saved correctly and the corresponding files were correct as well.

The falsification process was also correct, users completed the training level by level, saving the appropriate number of samples and with the application showing the necessary messages to the user. Moreover, the information about the signature provided to the user was appropriate to each level. Furthermore, the training log saved a record every time the user signed, as specified in the application requirements.

Although an evaluation of the forgers based on training logs is out of the scope of this project, a small graphic of a user's training is presented in order to show an approximate evolution of the forger's knowledge of a signature during the training. The result of using DTW algorithm to compare the original signature with the forged one has ranges from 0, identical, to 5, completely different. In this training, the number of samples the user had to create per level amount to 3, and the algorithm threshold that decided whether a signature is good enough or not was 1.50.

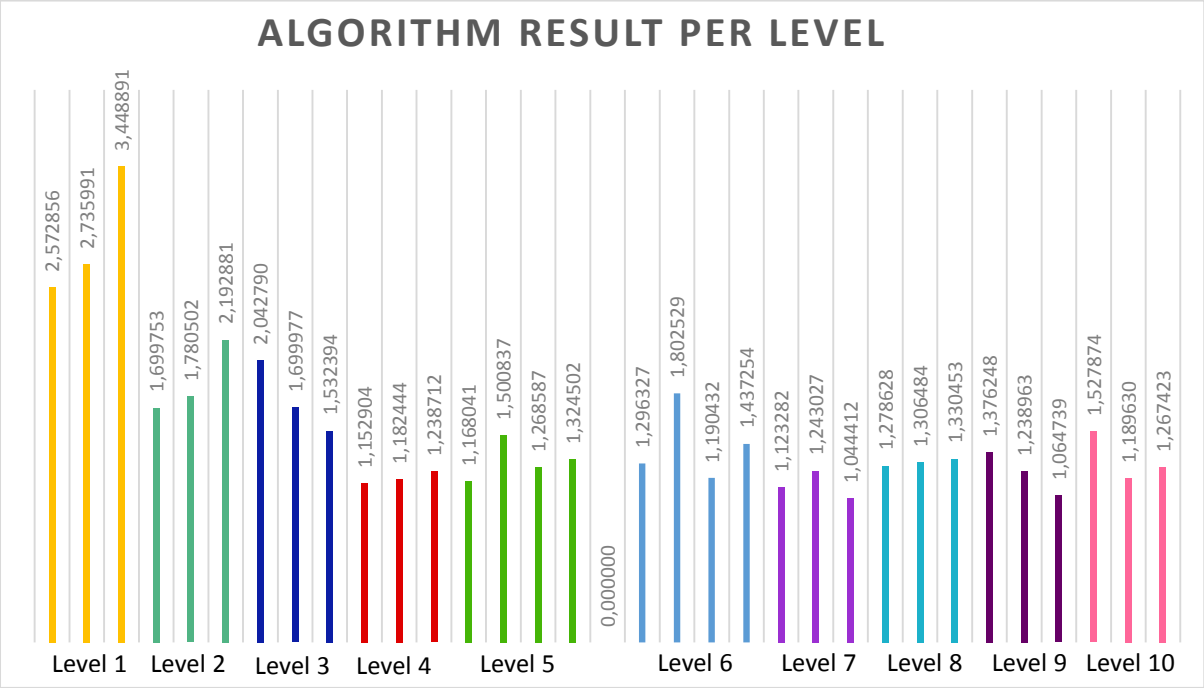

Figure 4 – Algorithm result of all signatures of a particular training

In level 1, the user did not know anything about the signature he was trying to falsify. It is therefore the level with the highest result in the comparison. In level 2, the user could see the signature for barely 5 seconds, so the first samples saved provided a better result than the third one (as time passed the memories of the signature faded). In level 3, as the user could see the signature as many times as he needed, the results were hence better than in the previous level. In level 4, one of the best results was reached, as the signature was shown on the table while the user, who could actually draw upon it, was trying to imitate it.

In levels 5 and 6 a video of the signature was shown to the user, however, in this particular training, there is not a huge improvement of the result, although it is better than in level 3. Level 7 is the one that gives the user the greatest amount of information about the signature, and therefore it is the level with the best result, as expected. The first sample of level 6 was erased by the user, so the algorithm does not compare it with the original sample, and it returns a symbolic 0 instead.

In the following levels, no information of the signature is provided, so that the user has to demonstrate what he has learned. The results after completion of level 7 and after having waited for one hour after finishing level 8 are quite similar. However, in level 10 there is a slightly increment of the result as the user waited for a day between levels 9 and 10.

Overall, there is a gain of knowledge about the signature as the user passes levels. Nevertheless, the user did not obtain a result smaller than 1 in any sample, so if the threshold would have been smaller than 1.50 the user would have had some difficulties during the training, and he would have had to practise the signature more.

7. Conclusions and future work

This project's main goals were to create a tool that trains forgers to falsify a signature by using a tablet, and record all training data for future studies.

This application can capture genuine signatures in order to create a database to be forged afterwards by a forger. A database for forgers can be created as well, since forgers have to forge original signatures as part of their training. The training consists in 10 different levels in which the program will vary the amount of information about the original signature to be shown to forgers. Moreover, during the whole training process, there would be information from every time the user signs in the tablet being recorded in a training log. In addition, all different training parameters of interest can be configured by the administrator of the program in order to create the environment needed for the research.

After the development of the application and with all the tests run, it can be asserted that the initial requirements have been fulfilled and that the application works properly: it captures signatures in the right format, it saves all files with the names requested, it registers all the information needed in training logs and it guides the different users through all the process of falsifying a signature.

As it can be seen in the tests, even though a formal evaluation of the system was not made, the ability of a trained user to imitate the original signature improves after going through the levels, which was indeed the expected behaviour

Even though the application accomplished the initial requirements, there are several improvements that could be made in order to enhance its performance:

- It would be really interesting to combine two types of algorithm: on the one hand, the dynamic algorithm (DTW) could process the behaviour of the signature and all its parameters, and on the other hand, a static algorithm could be added as well in order check the final image of both genuine and forged signatures, resulting in an increase of the reliability of the identity verification process.
- Another interesting improvement would be to adapt the application to different models of tablets, not only for Wacom STU-500.

Besides the improvements in the application, another future line of work would be the study itself of all the way forgeries are made and how to improve biometric systems using the knowledge this application can provide.

Thanks to the training logs saved, it is possible to study how the forger has been learning and improving its forgery. Therefore this application is a tool very useful to conduct a formal signature evaluation for research. This kind of evaluations helps to improve the security of algorithms, by determining how difficult it is to deceive the system and hence, its real security.

It can also measure the real risk of a signature to be forged. As the application's last levels simulate how well a forger will perform not only when he/she finishes the training, but also some time after the completion of training as well (some hours or even a whole day later). This could help to model what happens to the forger's knowledge after he/she has slept and tries to repeat the signature again the following day.

Overall, evaluations could allow researchers to quantify the level of expertise an attacker truly has.