Eye-based keystroke prediction for natural texts – a feasibility analysis

José Reverte Cazorla[1], José María de Fuentes[1], and Lorena González-Manzano[1]

[1]Computer Security Lab (COSEC), Universidad Carlos III de Madrid, Madrid, Spain.

***Abstract*—The use of videoconferencing is on the rise after COVID-19. Thus, it is common to observe the interlocutor while typing in the keyboard. A side-channel attack may be launched to infer the text written from the face image. In this paper, we analyse the feasibility of such an attack, being the first proposal which work with a complete keyset (50 keys) and natural texts. We use different scenarios, lighting conditions and natural texts to increase realism. Our study involves 30 participants, who typed 49,365 keystrokes. We characterize the effect of lighting, gender, age and use of glasses. Our results show that on average 13.71 % of keystrokes are revealed without error, and up to 31.8%, 52.5% and 61.2% are guessed with a maximum error of 1, 2 and 3 keys, respectively.**

***Keywords*-Keystroke; Eye tracking; Prediction**

I. Introduction

Nowadays, people around the world interact with electronic devices in their daily life to perform essential activities, for example, to make bank transactions or to browse the web [1], [2]. This may lead to the emergence of risks that threaten user's privacy, such as whether devices protect users' data in the right way or if generated data is useful for identifying the users' behaviours (e.g. hand or eye movement) or biometric traces (e.g. fingerprints). Indeed, the use of users' biometric and behavioral traces is common in many tasks, such as for authentication, human-computer interaction, user identification or inference attacks. One popular approach of exploiting the human behaviour in these fields is keystroke prediction attacks using side-channels [3], that is, data is indirectly exfiltrated without direct contact between victim and attacker.

To address this issue, many approaches reconstruct users' keystrokes based on side channels such as WiFi based [4]–[7], sensor based [8]–[11], video based [12]–[19] or eye tracking based [20], [21]. These last ones are inspired by the fact that human eyes naturally focus on and follow the keys they type. By recording the user's eyes and processing the images making use of eye-tracking techniques, it is extracted where the user is looking at to reconstruct, e.g., authentication patterns and passwords.

However, eye-movement approaches only focus on touch-screen devices with smaller range of possible keys like 10-digit screen keyboards, 9-points authentication patterns or the alphabet keys but only with a certain set of words. The use of these techniques to extract user keystrokes in a computer keyboard without input limitation is not yet investigated. With the recent increase in videoconferences motivated by the COVID-19 [22]–[24], these techniques could be used to attack an interlocutor when typing on his/ her keyboard during the call. Thus, in this work a mechanism to infer what a user is typing on a physical keyboard for natural texts under different light conditions is proposed. In particular, the contributions are as follows:

- A technique to extract features from keyboard typing leveraging a video source is developed.
- An experimental feasibility analysis is carried out by 30 users in different scenarios and light conditions.
- A proof of concept implementation is released along with the experimental dataset.

The organization of the rest of this work is as follows: In Section II the background is introduced; in Section III the proposed mechanism to address the model definition is defined; in Section IV an evaluation of the proposed mechanism is carried out and its impact factors are analysed. Later, in Section V related works similar to this research are studied; and, finally, in Section VI conclusions and some future research lines are outlined.

II. Background

This section provides the main notions related to the proposal in terms of face recognition techniques and AI classifiers.

A. Face recognition techniques

Face recognition models are used in order to extract face features. Face recognition has become one of the most active applications of pattern recognition and image analysis. Recently, significant research efforts have been focused on video-based face modeling, recognition and system integration [25]. For example, algorithms like K-Nearest Neighbours (KNN), Support Vector Machine (SVM) or J48 are used for this task [26]. Similarly, SVM is also pointed out for this purpose in [27].

B. Artificial Intelligence-based classifiers

Artificial Intelligence classification algorithms are used for predicting classes from data based on supervised learning techniques. Many classification algorithms can be found in this area but in this work five different well known classification algorithms [28], [29] are introduced for being the ones applied: Logistic regression (LR), K-Nearest Neighbours (KNN), Support Vector Machine (SVM), J48 and Logistic Model Tree (LMT). LR is an statistical method used for finding the best fitting model that represents the best dependent variable (the class). KNN is a lazy learning algorithm which stores instances of a dataset in a n-dimensional space and when a new unknown instance is given, it returns its class prediction by searching the most common class in the closest k instances. SVM is based on differentiating the classes by using kernels, which are n-dimensional functions for separating data features of different classes. J48 decision tree is an implementation of the C4.5 algorithm [30] made by WEKA [31] which basically chooses an attribute of the data that most effectively splits the samples for every node of the tree. Lastly, LMT is a decision tree which uses logistic regression algorithms at its leaves. It works by transforming the typical decision tree output as a dependent variable for training the logistic regression models with it.

III. Approach

In this Section, the proposed approach is introduced (Section III-A). The goals at stake and the experimental process are described in Section III-B and III-C, respectively.

A. Description

The goal is to assess the feasibility of guessing keystrokes by analysing a video streaming of the user face. To achieve this goal, several tasks are carried out, depicted in Figure 1.

Extracted features are analysed in streaming mode so that no image is saved on disk for later processing. Moreover, not all video frames represent keystrokes. In this approach, a frame is collected when a key is pressed.

The first step is to extract information of the user's face. For this task, DLIB's frontal face recognizer [32] is used (Figure 1, step 1). This face recognizer uses Histograms of Oriented Gradients (HoG) together with a Linear SVM classifier making it lightweight and computationally efficient [33], which is ideal for real time image processing. Based on the frontal face output, a shape recognition machine learning model [34] is applied to detect the facial landmarks (step 2). For this purpose, we leverage the approach proposed by Amato et al. [35]. In particular, three features are at stake – right and left eyes pupil centre, an invariable face point to represent the position of the face in the video and the head tilt angle. All of them are extracted or computed using OpenCV library [36].

Afterwards (step 3), different filters are applied for better pupil recognition to exaggerate the rounded shape of the pupil, along with the application of a threshold to emphasize the color difference between the pupils and the rest of the image. It must be noted that the threshold can be customized with values from 0 to 255, as it depends on the light conditions. In particular, the two zones delimited by the eyes landmarks are expanded with a dilate using a square kernel and one iteration for defining a pupil search zone. Moreover, the entire image is set in white but the search zones are colored in gray scale. The custom threshold in the search zones is applied to get the pupil contours more delimited. Afterwards, two iterations of erode are carried out for rounding the pupils extracted in previous step. Dilatation is then applied 4 times so the pupil zone is expanded without the noise removed in previous step. Lastly, blur is applied using the median filter so the pupils appear smoother and the pupil's contours are found using the Suzuki algorithm [37] and their centre is calculated with the moment of the shape.

The two pupil-related features (i.e. the pupils' centres) are computed based on a rectangle surrounding the eyes. In particular, the highest, lowest, leftmost and rightmost coordinates of the 6 landmark points, representing the eyes, define the sides of the rectangles (see Figure 2). These values are useful to ascertain the pupil position with respect to the whole eye. Thus, when looking at certain point and moving the head in the two-dimensional plane parallel to the camera, pupils change position as seen on Figure 2-a and 2-b. In the case of head movement in the row angle also the rectangle deforms (Figure 2-c). Due to these factors, the point representing the position of the face in the video and the head tilt angle are also features applied for the classification task (step 4).

Concerning the head tilt value, we compute the angle formed by a 180-degree line and a line calculated from the points 1 and 17 from the landmarks array previously obtained. Our experiments show that this is a suitable value while preserving the overall accuracy and speed of the model.

B. Goals

- **Multi-environment.** The system should work under different ambient lights and in different scenarios.
- **Accuracy.** The system should reveal the pressed keys of the user with precision.
- **Multi-user.** The mechanism must be suitable for users with different profiles (e.g., age, gender, etc.).

C. Experimental process

The experiment is carried out as follows:

1) The subject is asked to sit down and to adapt the chair and the laptop to feel comfortable when typing as they do naturally.
2) When the subject is ready, the data extractor is launched. A random text appears for the user to write, at the time the webcam records his/her face together with a threshold adapter.
3) The subject is asked to position themselves concentrating their gaze on the keyboard as if they were going to start typing. At this moment the threshold is configured according to the existing light condition. As depicted in figure 3, if excessive threshold is applied (b), pupil blends in with the rest of the eye causing a very poor accuracy when extracting the pupil's center. In the other side, low thresholds (c) causes no pupil detection or partially detection.
4) The user copies the presented text and, when completed, presses the key 'ESC' for finishing the test. Then, file with the keystrokes and face features is saved on disk.

IV. Assessment

This section presents the assessment results. In particular, Section IV-A introduces the experimental settings, whereas

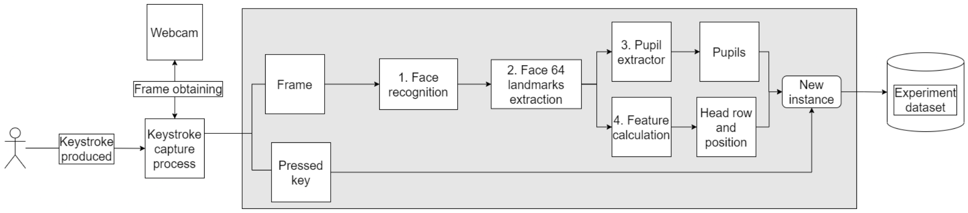
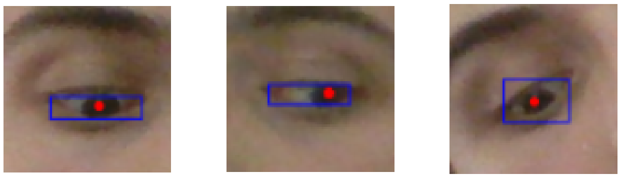


Fig. 1: Feature extraction process



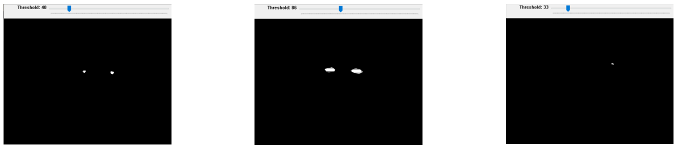
(a) Head at x=459 y=245 (keyboard's left side)

(b) Head at x=97 y=194 (keyboard's right side)

(c) Tilted head 40.5 degrees to the right

NOTE: The frame shape is x=640, y=480

Fig. 2: Pupil extraction pointing to 'q' key with different head positions



(a) Optimal

(b) Too much

(c) Very low

Fig. 3: Adjusting the threshold for an optimal pupil extraction

Section IV-B describes the parameters and assessment metrics. Overall results are discussed in Section IV-C and the analysis per user and environment factor is carried out in Section IV-D. A comparison against the experimental results of previos work is shown on Section IV-E. Lastly, the limitations of our study are addressed in Section IV-F.

A. Experimental settings

The experiment consists of making users write a text between 222 and 634 characters long while their face and eyes are being monitored by the laptop camera. 30 participants took part by typing at least one text. A total of 116 tests were made, collecting 58,384 keystrokes. These were preprocessed to remove irrelevant symbols and to normalize characters. For example, 'é' was converted into 'e' as it is not produced by a single keystroke. Thus, the final dataset contained 49,635 keystrokes. The keyset contains 50 keys – 27 characters of the Spanish alphabet, 10 digits, 11 symbols and the "backspace" and "Enter" keys (see Appendix).

The dataset aims to gather a plethora of different user settings. Thus, 57 of the tests were carried out by males and 59 by females, 103 from ages between 17 and 28 and 13 from people aged 29-56. Moreover, 22 of the 116 tests were done by people with glasses, 94 without them. Concerning lighting 53 tests were carried out in a well-lit environment, 31 in dim light and 32 in a very good artificially illuminated space. Lastly, in 59 of the tests participants were told to force their gaze as much as they can key by key, the remaining ones (56) were told to type the way they normally do.

To comply with data protection regulations, every participant signed a consent for the processing of their data through an online form, declaring their freely and voluntary participation in the experiment and authorizing the treatment of their personal data. Note that, because the designed mechanism operates processing images in streaming, no image of participants' faces were saved on disk.

The laptops for the experiment were an ASUS UX410U and an ASUS UX430U which practically have the same shape and keyboard. In particular, the keyboard layout is Spanish QWERTY. Experiments were carried out in 11 different places to increase the realism. Participants were asked to change the ambient light to adjust it to the defined three types of lighting (i.e. blinds, curtains, etc) when possible.

The code for the experiments and the dataset are publicly released in Google Drive[1] to foster further research.

B. Parameters and metrics

The five machine learning classifiers introduced in section II-B, in line with [26], are used in the evaluation, as well as collected data.

Fine-tuning was firstly addressed for each classifier to find the best parameters (Table I, marked in bold). Afterwards, 10-fold cross validation was applied for the assessment. A total of 270 classifiers were trained.

Algorithm	Parameter	Values
LR	Ridge	1E-12,5, **10**
KNN	Neighbours	1, 32, **65**
J48	Confidence factor	**0.01**, 0.05, 0.1
J48	Minimum instances per leave	**6**, 9, 12
LMT	Minimum instances per leave	**1**, 15, 31
LMT	Trimming weight	**0**, 0.5, 1
SVM	Cost	**1**, 2.5, 5

TABLE I: Fine tuning, best parameters in bold

Once the best parameters are known, by aggregating keystrokes depending on different factors (i.e. gender, youth, glasses, forced gaze, lighting conditions, 12 different smaller

[1] https://drive.google.com/drive/folders/1hLkHpsBLakNuh4iHKGwYwyBMuj_Bt46l?usp=sharing

sets of keystrokes where obtained from the original dataset. The 5 classifier were trained with them in a 10 cross validation. A total of 700 classifiers were trained and evaluated. The following metrics are considered in the analysis:

1) Physical Error (PE), which is the distance in the keyboard between the predicted key and the original key. For example, if a 'w' keystroke is classified as 't', then $PE = 3$.
2) Accuracy (Acc) represents the percentage of correctly classified keystrokes.

It must be noted that PE helps on characterizing the uncertainty of a prediction. In the keyboard at stake, each key has an average of 4.72 keys for $PE = 1$, whereas it raises to 12.48 for $PE = 2$ and to 19.96 for $PE = 3$. Therefore, in the following we analyze not only the accuracy in the absence of errors (that is, $PE = 0$), but also for different PEs.

C. Overall accuracy and physical error

Results are depicted in Table II. In general, J48 is the best classification algorithm, with $Acc = 13.71\%$ without error (i.e., $PE = 0$), whereas $Acc = 31.88\%52.50\%and61.25\%$ for $PE = 1, 2and3$, respectively, on average for KNN in the first case and J48 in the last cases. However, KNN is also relevant as it outperforms J48 under dim light.

In order to assess how the type of gaze (i.e. forced or natural) impacts the system, both of them were studied independently using J48. Results are depicted in Table III. Forced gaze leads to an average increase of 3.92 % of Acc and then, results are not significantly different regardless of the compared feature. Since forced gaze is not common in real scenarios, this result supports that the natural use of the keyboard offers similar accuracy results.

The only remarkable difference in terms of natural-forced gazed is for old-aged people, with glasses and artificial light leading to 18.38 %, 18.83 % and 18.05 % of difference in Acc for $PE = 3$. Specially the use of glasses and artificial light is quite challenging because of reflection (recall Section III), thus a forced gaze contributes to have better results.

D. Environment and user suitability. Results per factor

An analysis of the results for each of the factors at stake is carried out in the following (Table II).

1) Gender: Surprisingly, almost all algorithms achieve higher accuracy for men with no PE but the different is really small – around 1-2% of Acc. Using LR and SVM women results are slightly better than men, while the opposite happens in the remaining set of algorithms. However, in any case, gender-based differences are not remarkable. This is consistent among executions considering the low values of standard deviation.

2) Age: Results from young people (i.e., between 17 and 28 years old) produce more accurate predictions in all algorithms except for LR. Interestingly, a difference of 13.75 is relevant for $PE \geq 1$. Indeed, more than half of the predictions are made ($Acc = 52.66$) with $PE \leq 2$ for young people, while it leads to $PE = 3$ for old participants. Moreover, predictions for old users are subject to greater variability as the standard deviation is higher (around 0.5 more).

3) Glasses: The use of glasses has a remarkable impact in the results. In particular, one third of keystrokes are predicted with an error of one key at most (i.e, $PE \leq 1$), whereas the error is doubled when the user wears glasses. There is a difference in accuracy of 8 % when no glasses are present, leading to $Acc = 32.96\%$, 53.33%, 62.53% for $PE = 1$, 2 and 3, respectively and J48 algorithm in all cases.

4) Type of gaze: When participants were asked to force their gaze, all algorithms perform better except for LR. However, the improvement is constant for all PE values, with a maximum increase of 3.92 %. Therefore, forced gaze does not lead to a substantial gain.

5) Light conditions: Both dim and natural light lead to similar results, consistently higher than artificial light. The difference is constant for all PEs. Although differences are not remarkable, the overall results are relevant – more than one third of keystrokes are predicted with a maximum error of one physical key (i.e., $PE \leq 1$). Moreover, artificial light leads to higher variability of results according to the standard deviation.

E. Comparison against previous works

Among all related works that will be introduced in Section V, a comparison with the results of the most similar pair of works [20], [21] is depicted in Table IV.

The comparison is not straightforward as their assessment method is not equivalent to ours. Thus, EyeTell offers a set of top-k candidates, whereas GazeRevealer compares the real input with the predicted value as in our case. For the sake of fairness, we convert the reported top-5 accuracy from EyeTell with a factor of 20 %, which is the probability of getting the right value at the top-1. Note that top-1 can be regarded as equivalent to the prediction made by GazeRevealer and our work. Moreover, it must be noted that our work is the only one considering the impact of different user-related factors.

On the other hand, even if our accuracy figures are smaller, it is relevant to note the predicted inputs. In particular, both EyeTell and GazeRevealer focus on PINs (i.e., digits only). EyeTell also predicts one word out of a list of 27. On the contrary, we predict keystrokes from a 50-keys keyboard. For the sake of fairness, the gain against a random guess is computed. Our results show that our work is in line with GazeRevealer and indeed better accuracy is achieved when forced gaze is at stake. Lastly, as opposed to both works, we characterize the physical error so that we anticipate the amount of uncertainty for each prediction.

F. Impact factors and limitations

Factors that impact in the experiment are discussed in the following.

1) Gaze: Not everyone looked at the keyboard while typing. In fact, 8 of the 30 participants barely looked at the keys when typing for the natural gaze tests. This means that except for the forced gaze tests, there is a lot of noise in the dataset

		All keystrokes	Gender		Age		Glasses		Gaze		Light		
			Men	Women	Old	Young	Yes	No	Forced	Natural	Natural	Darkness	Artificial
KNN Acc (stdev)	PE=0	13.69 (0.36)	14.47 (0.77)	13.31 (0.39)	13.28 (0.92)	13.63 (0.58)	13.03 (0.48)	13.63 (0.37)	15.25 (0.55)	12.61 (0.32)	14.57 (0.66)	15.01 (0.77)	12.91 (0.88)
	PE≤1	31.88 (0.79)	32.42 (0.8)	32.23 (0.58)	20.71 (1.28)	31.89 (0.89)	18.07 (0.36)	31.57 (0.51)	35.74 (0.69)	29.56 (0.34)	32.51 (0.69)	35.59 (0.68)	31.42 (1.52)
	PE≤2	51.25 (0.6)	51.4 (0.77)	51.96 (0.59)	37.33 (1.88)	51.03 (0.45)	34.64 (1.05)	51.03 (0.43)	54.85 (0.76)	48.84 (0.62)	51.23 (0.71)	55.56 (0.62)	51.37 (1.11)
	PE≤3	60.97 (0.66)	61.13 (0.83)	61.74 (0.67)	53.13 (1.55)	60.87 (0.27)	50.53 (1.55)	61.3 (0.57)	65.11 (0.65)	58.35 (0.77)	61.15 (1.27)	65.01 (0.71)	60.69 (1.02)
J48 Acc (stdev)	PE=0	13.71 (0.31)	14.98 (0.4)	13.38 (0.52)	12.99 (1.19)	13.94 (0.37)	13.48 (0.27)	14.51 (0.46)	15.79 (0.73)	13.21 (0.17)	14.88 (0.45)	14.78 (0.47)	13.0 (0.49)
	PE≤1	31.76 (0.45)	33.69 (0.44)	31.91 (0.73)	18.38 (1.97)	32.13 (0.2)	17.27 (0.35)	32.96 (0.49)	34.84 (0.54)	31.65 (0.4)	33.15 (0.65)	34.88 (0.56)	31.63 (0.81)
	PE≤2	52.50 (0.34)	53.02 (0.61)	53.22 (0.91)	35.43 (1.95)	52.66 (0.47)	36.49 (0.45)	53.33 (0.5)	54.19 (0.69)	52.82 (0.77)	53.05 (0.97)	55.0 (0.68)	52.27 (0.81)
	PE≤3	61.25 (0.29)	61.99 (0.58)	61.62 (0.83)	52.75 (1.87)	61.43 (0.5)	54.43 (0.32)	62.53 (0.59)	63.98 (0.49)	60.02 (0.66)	61.81 (0.83)	62.93 (0.62)	60.8 (0.87)
LMT Acc (stdev)	PE=0	13.3 (0.35)	15.09 (0.48)	13.03 (0.99)	12.37 (0.73)	13.71 (0.32)	13.27 (0.62)	14.05 (0.32)	15.66 (0.62)	12.7 (0.19)	14.42 (0.73)	14.12 (0.57)	12.13 (0.89)
	PE≤1	31.0 (0.4)	33.39 (0.7)	31.3 (1.17)	16.96 (1.16)	31.38 (0.51)	17.14 (0.75)	32.11 (0.66)	35.0 (0.77)	29.79 (0.94)	32.17 (1.14)	33.31 (0.63)	29.56 (1.14)
	PE≤2	51.08 (0.7)	52.76 (0.62)	51.67 (0.85)	35.24 (1.72)	51.1 (0.57)	36.41 (0.84)	51.74 (0.82)	53.56 (0.91)	51.06 (0.81)	51.11 (0.97)	52.67 (0.81)	49.5 (1.22)
	PE≤3	60.15 (0.5)	61.6 (0.56)	60.84 (0.7)	53.49 (1.36)	60.61 (0.56)	54.21 (0.65)	61.51 (0.62)	63.55 (1.21)	59.08 (0.69)	60.97 (0.93)	61.52 (0.76)	58.73 (0.84)
LR Acc (stdev)	PE=0	11.05 (0.05)	10.56 (0.15)	11.59 (0.14)	12.7 (0.25)	10.94 (0.08)	12.88 (0.3)	10.59 (0.05)	10.15 (0.21)	12.08 (0.1)	10.52 (0.19)	12.24 (0.25)	11.01 (0.04)
	PE≤1	26.18 (0.09)	25.83 (0.17)	26.77 (0.18)	15.9 (0.61)	25.92 (0.11)	16.44 (0.45)	25.21 (0.16)	25.42 (0.41)	28.13 (0.15)	25.92 (0.27)	27.94 (0.29)	25.97 (0.09)
	PE≤2	48.42 (0.11)	47.73 (0.25)	49.04 (0.2)	34.73 (0.99)	48.0 (0.09)	35.65 (0.55)	47.36 (0.18)	47.66 (0.49)	49.72 (0.25)	48.62 (0.29)	47.49 (0.48)	48.97 (0.2)
	PE≤3	56.66 (0.18)	56.04 (0.36)	57.32 (0.2)	53.66 (1.1)	56.41 (0.1)	54.02 (0.51)	56.18 (0.22)	57.45 (0.71)	57.0 (0.28)	57.78 (0.31)	54.73 (0.55)	57.3 (0.19)
SVM Acc (stdev)	PE=0	13.38 (0.35)	12.62 (0.62)	14.22 (0.69)	11.45 (0.89)	13.56 (0.31)	12.3 (0.49)	13.58 (0.43)	15.45 (0.32)	11.23 (0.21)	12.63 (0.48)	12.78 (0.68)	10.69 (0.7)
	PE≤1	29.33 (0.42)	28.33 (0.65)	30.86 (0.84)	16.46 (1.42)	29.28 (0.53)	16.99 (0.53)	29.19 (0.46)	31.35 (0.73)	27.79 (0.52)	27.78 (0.47)	30.89 (0.78)	27.46 (1.05)
	PE≤2	49.58 (0.52)	48.89 (0.64)	50.68 (0.49)	33.99 (1.51)	49.49 (0.63)	35.02 (0.86)	49.38 (0.36)	51.5 (0.5)	48.24 (0.44)	48.4 (0.55)	51.0 (0.51)	48.93 (1.08)
	PE≤3	59.09 (0.48)	58.38 (0.48)	60.21 (0.42)	52.24 (1.11)	59.14 (0.53)	52.27 (0.78)	59.39 (0.38)	61.53 (0.34)	57.15 (0.27)	58.33 (0.38)	59.43 (0.88)	59.08 (1.04)

TABLE II: Accuracy for each Physical Error (PE). Results per factor and algorithm

		All keystrokes	Gender		Age		Glasses		Light		
			Men	Women	Old	Young	Yes	No	Natural	Darkness	Artificial
J48 Forced Gaze Acc (stdev)	PE=0	15.79 (0.73)	18.05 (1.0)	14.87 (0.92)	12.1 (1.46)	16.71 (0.86)	12.36 (1.06)	17.07 (0.64)	18.56 (1.08)	15.46 (0.79)	14.86 (1.22)
	PE≤1	34.84 (0.54)	36.99 (1.32)	33.86 (1.26)	22.14 (2.12)	35.69 (1.17)	20.45 (0.99)	37.26 (0.57)	37.57 (0.97)	37.26 (1.49)	35.99 (1.82)
	PE≤2	54.19 (0.69)	55.61 (1.16)	43.54 (1.13)	33.1 (2.22)	55.11 (0.87)	39.98 (0.93)	55.39 (0.41)	55.77 (0.71)	58.54 (1.18)	55.72 (1.63)
	PE≤3	63.98 (0.49)	64.71 (1.06)	57.68 (1.33)	42.35 (2.98)	64.89 (1.01)	62.93 (1.03)	65.27 (0.85)	65.26 (1.19)	65.9 (1.15)	65.16 (2.09)
J48 Natural Gaze Acc (stdev)	PE=0	13.21 (0.17)	13.8 (0.46)	13.26 (0.28)	14.67 (0.68)	13.39 (0.29)	14.91 (0.73)	13.25 (0.35)	13.67 (0.32)	15.1 (1.02)	13.13 (0.91)
	PE≤1	31.65 (0.4)	33.11 (0.99)	31.25 (1.06)	21.44 (1.34)	31.87 (0.4)	18.97 (0.61)	31.66 (0.66)	32.56 (0.76)	35.16 (1.76)	20.27 (1.11)
	PE≤2	52.82 (0.77)	54.06 (0.92)	52.25 (0.74)	40.37 (1.67)	52.7 (0.53)	28.38 (0.81)	52.67 (0.65)	52.95 (1.15)	54.62 (1.57)	30.88 (1.42)
	PE≤3	60.02 (0.66)	60.77 (0.79)	60.07 (0.65)	60.69 (1.37)	60.26 (0.68)	44.1 (0.98)	60.44 (0.54)	59.95 (1.12)	63.05 (1.41)	47.11 (1.49)

TABLE III: Gaze comparison using J48 algorithm

	Experimental results				User and environment factors			
	Predicted	Eval. method	Accuracy (%)	Gain against random	Gender	Age	Glasses	Light
EyeTell [20]	Entire 4-digit PIN	top-1	39.0	390	X	X	X	√
	Entire 6-digit PIN	top-1	39.0	39,000	X	X	X	√
	A word	top-5	7.68	207.36	X	X	X	√
GazeRevealer [21]	Single keystroke (4-d PIN)	Correct classified	73.6	7.36	X	X	√	√
	Single keystroke (6-d PIN)	Correct classified	73.9	7.39	X	X	√	√
Our research	Single keystroke (natural / forced gaze)	Correct classified	13.71 / 15.78	6.86 / 7.89	√	√	√	√

TABLE IV: Experimental results comparison

which greatly affects the accuracy of the model. However, it increases the realism of the experiments as not all users need to look at the keyboard.

2) Glasses: Two experiments required the user setup to be adapted by changing the distance with the light source to avoid reflections in glasses.

3) Light direction: During the experiments, light direction affected the model performance. It must be recalled that the threshold is the same for both eyes. Thus, when the optimal value was set, one eye could receive much more light than other, leading to the generation of noise. The best setting was to record users' faces when looking straight to the camera.

4) Distance from the light source: Light intensity was one of the most sensitive factors for data collection. When pointing directly to the user with the light source at a very close distance (e.g., using a desk lamp or being in front of a window), light reflection reduced the size of the pupil and its detection was harder. As explained in Section III, pupil centre is extracted by handling image colors so the only dark shape in the picture are the pupils. If reflection appears it is hard to extract pupils' centre. Therefore, when working under artificial light, the best setting was having the light source indirectly pointing to user's face and far away enough to avoid reflections.

Beyond the user and environment factors, a set of limitations have been identified. First, our predictions are extracted directly from the video with no use of natural language processing. Since we characterize the degree of uncertainty by means of PE, our accuracy results could be enhanced with this technique.

On the other hand, our work is focused on a particular keyboard layout (Spanish QWERTY) and Spanish texts. Thus, our results should be confirmed with varied layouts and languages. Lastly, our approach does not consider filtering the pupil readings when they fall out of the scope of the keyboard. As it is very keyboard-specific, we decided to leave this issue out of the study, but this could have a positive effect in both Acc and PE results.

V. Related work

Related works in terms of keystroke inference attacks based on different side channels are studied in the following.

A. WiFi-based

This type of attack infers the keystrokes by recording changes from the channel state information corresponding to hand or finger movements. There are approaches like WiKey [4] and WindTalker [5] which get the keystrokes on a physical and soft keyboards respectively using external WiFi access points as signal sources. WiPass [6] is presented as a way of getting passwords and authentication patterns using the victim's smartphone personal hotspot as emanation font or, recently, Shen et al. [7] present a similar approach to Ali et al. [4] and Li et al. [5] but using a different classification model. These attacks cannot tolerate changes in the environment other than the victim's hand or finger movement. Furthermore, transmitter and receiver should be spaced 1 meter apart at the most ([5]). Also, the orientation of the victim's device, and the victim's typing gestures were all fixed in the experiments.

B. Sensor-based

Keystrokes are indirectly inferred by using internal and external sensors. In [8], [9] the attack is carried out by the use of accelerometers and in [10] accelerometer data is combined with that of the gyroscope. More recently, AlphaLogger [11] combine accelerometer, gyroscope and magnetometer data for this purpose. In [38], [39] keystrokes of physical keyboards are predicted by analyzing acoustic emanations of the typing process recorded by a microphone. Later, other approaches came up using the microphone and the gyroscope all together [40] and analyzing the time difference between keystrokes [41]. These approaches need to collect sensor data from the victim's device in a stealthy way, via malware infection or in a physically visible way.

C. Video-based

Video-based keystroke inference attacks consist of predicting the keystrokes from a video recording, pointing directly or indirectly to the keyboard with more or less vision obstructions while the victim is typing. Backes et al. [14], [15] predict content of a screen by using reflections on many objects. Raguram et al. [42] infer the keystrokes by the light diffusion around the keyboard in a video recording. This needs the attacker to directly video taped the keyboard while the user is typing. In [17], [18] a direct video of user typing is also used but in the case of [18] the image processing is done in streaming. Recently, Cardaiolli et al. [13] accomplish a way of extracting 4 digit PINs from ATMs PIN pads when users hide their keystroke with one of their hands. Another similar research infers smartphone authentication PINs by analyzing hand and finger movements in videos recorded in the typing process [19]. Back in 2012, Adam J. Aviv [12] showed a way to infer smart devices unlock patterns by analyzing pictures of the smudge marks left on the screen taken by other smart device. Another approach of attack is PIN skimmer [43], which uses the smartphone front camera and the microphone to infer smartphone authentication PINs. In particular, they analyze image distortions in the video produced by the hand movements along with the sound made by the finger when typing in the screen.

D. Eye motion based

It can be considered a subgroup of video based inference attacks but it is independently studied due to the fact that this field of research is the one closed to this proposal. There are two main works, namely, EyeTell [20], and GazeRevealer [21]. In [20], keystrokes are inferred by capturing victims eye movement in the typing process on a smartphone. An external camera is used for this purpose, located at a fixed distance from the victim's eyes and recording angle (i.e. how tilted the camera is towards user eyes). The videos are saved for later processing which consists of extracting the pupils from every frame by using a Haar-like feature-based cascade classifier [44] and then, dividing the whole trace into a sequence of segments. On the other hand, GazeRevealer [21] infers authentication PIN passwords of smartphone users by capturing victim's eye movement using their front camera. It uses the Maximum IsoCenter technique [45] to extract the pupil centre from an image. For extracting the frames where keypress is produced, they used image histograms as the metric to distinguish sensitive images from the video. Also, because the user does not always stay still while typing, they take the angles of head movements and put them along with the eye position as features for later classification.

Our contribution is in line with the pair of mentioned proposals. Table V presents a comparison, where it is noticed that our proposal is the one based on eyes that uses a laptop keyboard, where the processing is online, with a significant amount of keystrokes analysed, bigger key set than in other proposals and comparable number of participants. More specifically, (1) we make keystroke prediction on a physical keyboard of a laptop, not in a soft keyboard; (2) the proposed model does not save video records of the user typing for later processing but the images are analyzed in streaming. Then, features used in the classification have to be extracted in real time; (3) in this research the problem of getting the exact frames, where the keystroke is produced, is set aside by only processing a frame per keypress, as opposed to EyeTell and GazeRevealer; (4) EyeTell and GazeRevealer study smaller range of possible keys and fixed input lengths. In contrast, in our proposal no input length limit is set and, also, a bigger range of possibilities is allowed, that is 50 possible keys; and (5) EyeTell and GazeRevealer do not analyze the impact of gender and age in their models, and the use of glasses neither in EyeTell.

VI. Conclusion. Future work

In this paper, the feasibility of a video-based keystroke inference attack has been assessed. In contrast to previous works, our approach is based on a full keyset from a physical keyboard and users were requested to type texts involving characters, digits and symbols. Furthermore, different user- and environment-related factors have been considered. Our results show that even if a relatively low fraction of keystrokes can be accurately predicted, a substantial amount of predictions are made with a limited error.

As a future work, model accuracy could be improved by adding a final phase of prediction through the use of natural language processing. Also, performance could enhance by reducing noise discarding key presses when the user is not

	Type	**Device**	**Data extraction**	**Number of participants**	**Total keystrokes collected**	**Key set**	**Input length**
Ali et al. [4]	WiFi	Laptop keyboard	Offline	10	4.070	37	No limitation
Li et al. [5]	WiFi	Smartphone keyboard	Offline	10	100	10	10
Zhang et al. [6]	WiFi	Smartphone keyboard	Offline	N/A	25	Pattern	Pattern
Shen et al. [7]	WiFi	Smartphone keyboard	N/A	5	250	10	No limitation
J. Aviv [12]	Video images	Smartphone keyboard	Offline	N/A	4	Pattern	Pattern
Backes et al. [14]	Video images	LCD Screen	Offline	-	-	-	-
Backes et al. Revisited [15]	Video images	LCD Screen	Offline	-	-	-	-
Raguram et al. [42]	Video images	Smartphone keyboard	Online	N/A	39 sentences	26	No limitation
Maggi et al. [18]	Video images	Smartphone keyboard	Online	N/A	2.246	N\A	No limitation
Balzarotti et al. [17]	Video images	PC keyboards	Offline	2	236 words	26	No limitation
Cardaioli et al. [13]	Video images	ATM’s PIN pad	Offline	40	29.000	10	5
Shukia et al. [19]	Video images	Smartphone keyboard	Offline	65	920	10	7 at much
Asonov et al. [38]	Sound sensors	PC keyboards	Offline	N/A	270	2	No limitation
Zhuang et al. [39]	Sound sensors	PC keyboards	Offline	N/A	16.478	30	5 to 10
Narain et al. [40]	Sound sensors	Smartphone and tablet	Offline	2	2.000	10	4 or 6
Zhu et al. [41]	Sound sensors	Smartphones	N/A	2	6.824	29	No limitation
Simon and Anderson [43]	Sound and motion sensors	Smartphone	Offline	4	1.800	10	4 or 8
Cai and Chen [8]	Motion sensors	Smartphone keyboard	Online	N/A	449	16	4 to 25
Owusu et al. [9]	Motion sensors	Smartphone keyboard	Online	4	1.300	60	6
Xu et al. [10]	Motion sensors	Smartphone keyboard	Online	2	120	12	4 to 8
Javed et al. [11]	Motion sensors	Smartphone keyboard	Online	10	N/A	26	No limitation
Chen et al. [20]	eye-based	Smartphone keyboard	Offline	22	1.320	10	4
	eye-based	Smartphone keyboard	Offline	22	4.400	10	6
	eye-based	Smartphone keyboard	Offline	22	29.700	26	7 to 13
Wang et al. [21]	eye-based	Smartphone keyboard	Offline	26	2.600	10	4
	eye-based	Smartphone keyboard	Offline	26	7.800	10	5
Our research	eye-based	Laptop keyboard	Online	30	49.635	50	No limitation

TABLE V: Research comparation table

looking at the keyboard. Finally, more devices and types of keyboards could be involved in the experimental process.

Acknowledgements

This work was supported by the Madrid Goverment (Comunidad de Madrid-Spain) under the multianual agreement with UC3M (“fostering young doctor research”, DEPROFAKE-CM-UC3M) and in the context of the V PRICIT research and technological innovation regional program; by CAM by grant CYNAMON P2018/TCS-4566-CM, cofunded with ERDF; and by Min. of Science and Innovation of Spain by grant ODIO PID2019-111429RB-C21 (AEI/10.13039/501100011033).

References

[1] T. Teo, “Demographic and motivation variables associated with internet usage activities,” *Internet Research*, vol. 11, pp. 125–137, 05 2001.

[2] Y. Yang, Y. Liu, H. Li, and B. Yu, “Understanding perceived risks in mobile payment acceptance,” *Industrial Management & Data Systems*, 2015.

[3] A. K. Sikder, G. Petracca, H. Aksu, T. Jaeger, and A. S. Uluagac, “A survey on sensor-based threats and attacks to smart devices and applications,” *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1125–1159, 2021.

[4] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, “Keystroke recognition using wifi signals,” in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, ser. MobiCom ’15. New York, NY, USA: Association for Computing Machinery, 2015, p. 90–102. [Online]. Available: https://doi.org/10.1145/2789168.2790109

[5] M. Li, Y. Meng, J. Liu, H. Zhu, X. Liang, Y. Liu, and N. Ruan, “When csi meets public wifi: Inferring your mobile phone password via wifi signals,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS ’16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1068–1079. [Online]. Available: https://doi.org/10.1145/2976749.2978397

[6] J. Zhang, X. Zheng, Z. Tang, T. Xing, X. Chen, D. Fang, R. Li, X. Gong, and F. Chen, “Privacy leakage in mobile sensing: Your unlock passwords can be leaked through wireless hotspot functionality,” *Mob. Inf. Syst.*, vol. 2016, pp. 8 793 025:1–8 793 025:14, 2016.

[7] X. Shen, Z. Ni, L. Liu, J. Yang, and K. Ahmed, “Wipass: 1d-cnn-based smartphone keystroke recognition using wifi signals,” *Pervasive and Mobile Computing*, vol. 73, p. 101393, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574119221000523

[8] L. Cai and H. Chen, “Touchlogger: Inferring keystrokes on touch screen from smartphone motion,” in *Proceedings of the 6th USENIX Conference on Hot Topics in Security*, ser. HotSec’11. USA: USENIX Association, 2011, p. 9.

[9] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang, “Accessory: Password inference using accelerometers on smartphones,” in *Proceedings of the Twelfth Workshop on Mobile Computing Systems; Applications*, ser. HotMobile ’12. New York, NY, USA: Association for Computing Machinery, 2012. [Online]. Available: https://doi.org/10.1145/2162081.2162095

[10] Z. Xu, K. Bai, and S. Zhu, “Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors,” in *Proceedings of the Fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WISEC ’12. New York, NY, USA: Association for Computing Machinery, 2012, p. 113–124. [Online]. Available: https://doi.org/10.1145/2185448.2185465

[11] A. R. Javed, M. Beg, T. Baker, and A. Al-Bayatti, “Alphalogger: detecting motion-based side-channel attack using smartphone keystrokes,” *Journal of Ambient Intelligence and Humanized Computing*, 02 2020.

[12] A. J. Aviv and J. M. Smith, “Side channels enabled by smartphone interaction,” 2012.

[13] M. Cardaioli, S. Cecconello, M. Conti, S. Milani, S. Picek, and E. Saraci, “Hand me your pin! inferring atm pins of users typing with a covered hand,” 2021. [Online]. Available: https://arxiv.org/abs/2110.08113

[14] M. Backes, M. Durmuth, and D. Unruh, “Compromising reflections-or-how to read lcd monitors around the corner.”

[15] M. Backes, T. Chen, M. Duermuth, H. P. Lensch, and M. Welk, “Tempest in a teapot: Compromising reflections revisited,” in *2009 30th IEEE Symposium on Security and Privacy*, 2009, pp. 315–327.

[16] Q. Yue, Z. Ling, X. Fu, B. Liu, K. Ren, and W. Zhao, “Blind recognition of touched keys on mobile devices,” in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS ’14. New York, NY, USA: Association for Computing Machinery, 2014, p. 1403–1414. [Online]. Available: https://doi.org/10.1145/2660267.2660288

[17] D. Balzarotti, M. Cova, and G. Vigna, “Clearshot: Eavesdropping on keyboard input from video,” in *2008 IEEE Symposium on Security and Privacy (sp 2008)*, 2008, pp. 170–183.

[18] F. Maggi, A. Volpatto, S. Gasparini, G. Boracchi, and S. Zanero, “A fast eavesdropping attack against touchscreens,” in *2011 7th International Conference on Information Assurance and Security (IAS)*, 2011, pp. 320–325.

[19] D. Shukia, R. Kumar, V. Phoha, and A. Serwadda, “Beware, your hands reveal your secrets !” in *Proceedings of the ACM Conference on Computer and Communications Security*, ser. Proceedings of the ACM Conference on Computer and Communications Security. Association for Computing Machinery, Nov. 2014, pp. 904–917, 21st ACM

Conference on Computer and Communications Security, CCS 2014 ; Conference date: 03-11-2014 Through 07-11-2014.

[20] Y. Chen, T. Li, R. Zhang, Y. Zhang, and T. Hedgpeth, “Eyetell: Video-assisted touchscreen keystroke inference from eye movements,” in *2018 IEEE Symposium on Security and Privacy (SP)*, 2018, pp. 144–160.

[21] Y. Wang, W. Cai, T. Gu, and W. Shao, “Your eyes reveal your secrets: An eye movement based password inference on smartphone,” *IEEE Transactions on Mobile Computing*, vol. 19, no. 11, pp. 2714–2730, 2020.

[22] “Zoom revenue and usage statistics (2022),” Jun 2022. [Online]. Available: https://www.businessofapps.com/data/zoom-statistics/

[23] D. Thorp-Lancaster, “Microsoft teams hits 75 million daily active users, up from 44 million in march,” Apr 2020. [Online]. Available: https://www.windowscentral.com/microsoft-teams-hits-75-million-daily-active-users

[24] J. Peters, “Google’s meet teleconferencing service now adding about 3 million users per day,” Apr 2020. [Online]. Available: https://www.theverge.com/2020/4/28/21240434/google-meet-three-million-users-per-day-pichai-earnings

[25] S. G. Bhele, V. Mankar, *et al.*, “A review paper on face recognition techniques,” *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 1, no. 8, pp. 339–346, 2012.

[26] M. B. Abdulrazaq, M. R. Mahmood, S. R. Zeebaree, M. H. Abdulwahab, R. R. Zebari, and A. B. Sallow, “An analytical appraisal for supervised classifiers’ performance on facial expression recognition based on relief-f feature selection,” in *Journal of Physics: Conference Series*, vol. 1804, no. 1. IOP Publishing, 2021, p. 012055.

[27] M. Sharif, F. Naz, M. Yasmin, M. A. Shahid, and A. Rehman, “Face recognition: A survey.” *Journal of Engineering Science & Technology Review*, vol. 10, no. 2, 2017.

[28] S. Gupta, D. Kumar, and A. Sharma, “Performance analysis of various data mining classification techniques on healthcare data,” *International journal of computer science & Information Technology (IJCSIT)*, vol. 3, no. 4, pp. 155–169, 2011.

[29] A. Baarah, A. Aloqaily, Z. Salah, M. Zamzeer, and M. Sallam, “Machine learning approaches for predicting the severity level of software bug reports in closed source projects,” *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 8, 2019.

[30] J. R. Quinlan, *C4. 5: programs for machine learning*. Elsevier, 2014.

[31] E. Frank, M. Hall, G. Holmes, R. Kirkby, B. Pfahringer, I. H. Witten, and L. Trigg, “Weka-a machine learning workbench for data mining,” in *Data mining and knowledge discovery handbook*. Springer, 2009, pp. 1269–1277.

[32] D. E. King, “Dlib-ml: A machine learning toolkit,” *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[33] M. P. Jacob, “Comparison of popular face detection and recognition techniques,” *International Research Journal of Modernization in Engineering Technology and Science, e-ISSN*, pp. 2582–5208, 2021.

[34] D. E. King, “dlib-models,” https://github.com/davisking/dlib-models, 2021.

[35] G. Amato, F. Falchi, C. Gennaro, and C. Vairo, “A comparison of face verification with facial landmarks and deep features,” in *10th International Conference on Advances in Multimedia (MMEDIA)*, 2018, pp. 1–6.

[36] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

[37] S. Suzuki and K. be, “Topological structural analysis of digitized binary images by border following,” *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32–46, 1985. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0734189X85900167

[38] D. Asonov and R. Agrawal, “Keyboard acoustic emanations,” in *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*, 2004, pp. 3–11.

[39] L. Zhuang, F. Zhou, and J. D. Tygar, “Keyboard acoustic emanations revisited,” *ACM Transactions on Information and System Security (TISSEC)*, vol. 13, no. 1, pp. 1–26, 2009.

[40] S. Narain, A. Sanatinia, and G. Noubir, “Single-stroke language-agnostic keylogging using stereo-microphones and domain specific machine learning,” in *Proceedings of the 2014 ACM Conference on Security and Privacy in Wireless; Mobile Networks*, ser. WiSec ’14. New York, NY, USA: Association for Computing Machinery, 2014, p. 201–212. [Online]. Available: https://doi.org/10.1145/2627393.2627417

[41] T. Zhu, Q. Ma, S. Zhang, and Y. Liu, “Context-free attacks using keyboard acoustic emanations,” in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS ’14. New York, NY, USA: Association for Computing Machinery, 2014, p. 453–464. [Online]. Available: https://doi.org/10.1145/2660267.2660296

[42] R. Raguram, A. M. White, D. Goswami, F. Monrose, and J.-M. Frahm, “Ispy: Automatic reconstruction of typed input from compromising reflections,” in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, ser. CCS ’11. New York, NY, USA: Association for Computing Machinery, 2011, p. 527–536. [Online]. Available: https://doi.org/10.1145/2046707.2046769

[43] L. Simon and R. Anderson, “Pin skimmer: Inferring pins through the camera and microphone,” in *Proceedings of the Third ACM Workshop on Security and Privacy in Smartphones; Mobile Devices*, ser. SPSM ’13. New York, NY, USA: Association for Computing Machinery, 2013, p. 67–78. [Online]. Available: https://doi.org/10.1145/2516760.2516770

[44] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, 2001, pp. I–I.

[45] R. Valenti and T. Gevers, “Accurate eye center location through invariant isocentric patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1785–1798, 2012.

Appendix

A. Excerpt of the experimental texts

Text 1. *El abuelo español de barba blanca me señala una serie de retratos ilustres. Este, me dice, es el gran don Miguel de Cervantes Saavedra, genio y manco, este es Lope de Vega, este Garcilaso, este Quintana. Yo le pregunto por el noble Gracian (...)*

Text 2. *<¡Atrida1 Creo que tendremos que volver atras, yendo otra vez errantes, si escapamos de la muerte, pues si no, la guerra y la peste unidas acabaran con los aqueos. (...)*

Text 3. *1234567890 1234567890 (...)*

Text 4. *qwypfghjkñzxcvbm qwypfghjkñzxcvbm (...)*

Text 5. *’º´‘+ ´ç,.-< º´`´■̧,.-¡ º´...)*