No Face-Touch: Exploiting Wearable Devices and Machine Learning for Gesture Detection

S. Marullo[1,2], T. Lisini Baldi[1], G. Paolocci[1,2], N. D'Aurizio[1,2], and D. Prattichizzo[1,2]

***Abstract*— Avoiding face-touches has been one of the most common medical recommendations since the beginning of the COVID-19 pandemic. This work aims at providing people with help in contrasting this widespread, yet noxious habit. The solution we present exploits wearable devices to detect hand motions ending up into a face-touch and promptly notify the user exploiting haptic feedback. To this aim, we propose a recurrent neural network taking as input temporal sequences of accelerometer data acquired by a smartwatch worn by the user. The trained RNN (NFT_RNN) achieves good generalization capabilities to data coming from different users, besides a lower false detections rate with respect to a rule-based detection algorithm. The suggested solution is ready-to-use and large-scale deployable, being portable on smartwatches, fitness bands and DIY devices.**

Fig. 1: No Face-Touch notifies the user whenever a face-touch movement is detected.

I. Introduction

Despite the advancements in technology and medicine, pandemics are still an open issue, causing millions of deaths even in the 21^{st} century. A paradigmatic example is the current pandemic of SARS-CoV-2, leading to over 34.8 million COVID-19 infections and more than a million deaths by the 5^{th} of October 2020 [1]. Worldwide governments and public institutions are fighting against the outbreak, with a great effort in developing vaccines, effective treatments, and policies for limiting the diffusion. Among the policies pursued in response to COVID-19, individual protective behaviour has a great impact on the reduction of the average number of infections caused by a primary case in a population consisting only of susceptibles. Respecting hygiene measures becomes even more valuable when virus transmission can occur by self-inoculation, i.e. by transferring contaminated material from hands to other body sites [2], [3]. Self-inoculation of common respiratory infections (e.g., influenza, coronavirus) has been described in literature [4], [5], [6], and researchers showed that contaminated hands greatly impact in disseminating respiratory infections [7], especially if associated with face-touches [8]. In crucial contexts, such as health care settings and school lessons, frequent face-touching is a potential mechanism of virus acquisition and transmission, therefore avoiding face-touches is a paramount prevention habit to be acquired. This aspect is further supported by a behavioural observation study conducted in [9], where subjects were videotaped to monitor the occurrences of face-touches. Of all face-touches, 44% of them involved contact with a mucous membrane. Of such touches, 36% involved the mouth, 31% involved the nose, 27% involved the eyes, and 6% were a combination of these regions. Furthermore, in [10], authors report that face-touch events occur most of the times with the non-dominant hand.

The act of face-touching is so rooted in some people, that usually happens with little awareness, and an external agent notifying the event is often needed. This is due to the fact that this gesture is a cultural and unconscious way to express an emotional state, such as fear, stress, surprise and focus of attention [11]. Hence, lowering the frequency of face touch occurrences is difficult. Beside that, there exist behavioural disorders strictly related to this movements. Trichotillomania [12], onychophagia [13], and dermatophagia [14] are just few representative examples.

Sensory stimuli proved to be suitable for providing alerts to users [15]. In this light, No Face-Touch proposes a system combining haptic feedback and Machine Learning to give a contribution to limit further transmission of SARS-CoV-2 and, more in general, to help people become more aware of their face-touching. In [16], the authors showed that the haptic feedback is effective in reducing the number of contact occurrences and the time spent by the hand in contact with the face. Here, we present an advancement, whose rationale was to develop an easy-to-use, portable solution with high accuracy. In this respect, differently from previous approaches, we will exploit only acceleration data coming from a device worn on the user's wrist, with the aim of achieving high face-contact detections rate and a low percentage of false

This work was supported by the IEEE Robotics and Automation Society Special Interest Group on Humanitarian Technology (RAS-SIGHT) Humanitarian Project "No Face-Touch: an Open Project to Limit the COVID-19 Outbreak".

[1]are with the Department of Information Engineering and Mathematics, University of Siena, Siena, Italy
Contact: smarullo@diism.unisi.it
[2]are with the Department of Advanced Robotics (ADVR), Istituto Italiano di Tecnologia, Genova, Italy.

detections. We will refer either to DIY or to commercially available devices, such as smartwatches and smart bracelets. We want to reach the widest possible population, considering also who cannot afford the cost of smartwatches. The No Face-Touch concept is visually summarized in Fig. 1.

II. Methodology

No Face-Touch is a project aimed at providing individuals with a solution for an effective reduction of face-touch occurrences. Our hypothesis is that gestures ending up in a face touch are described by common patterns among humans. However, touching the face is a gesture related also to mental states, encapsulating person-specific behavioural and emotional traits. This is why we adopted a systematic approach aimed at developing a solution with good generalization capabilities, potentially tunable according to some specific user behavioural needs. The code is publicly available at `https://github.com/sirslab/NoFaceTouch`. In the following of this section, we detail the pipeline for data collection and processing, including the neural network training procedure.

A. Data Collection

Data used for neural network training consisted of synchronized video recordings and acceleration measurements (acquired by means of a smartwatch). Participants were asked to wear a smartwatch on their non-dominant arm and perform diverse gestures while recorded by a RGB camera (a webcam or a smartphone camera). Video recordings were exploited only to appropriately label sensor data about face-touches. Hence, video recordings undergo a pre-processing synchronization with data coming from the sensor.

One of the main issues in Machine Learning is the need of acquiring a large amount of data to let the neural network learn with good generalization capabilities. In our case, labels have to be manually selected by inspecting the video-frames. In research on Machine Learning, much effort is put on having insights on the neural network explainability [17], i.e., in understanding *what* the neural network has learnt for retrieving the output. Lacking this form of comprehension, collecting a sufficiently diverse dataset is needed. To this aim, either the dataset is very large or it includes examples representative of most of the possible situations. Since labelling is highly time-consuming, we followed the second approach, hence some preliminary investigations have been conducted to identify gestures and patterns effectively meaningful for neural network training. More specifically, we found and included in the dataset several adversarial cases, i.e. gestures, contacts and motions that could be confused with face-touches. Furthermore, it can be easily envisaged that, from a statistical point of view, the probability that a movement involving the hand is due to an interaction with the face is much lower than the probability that the hand is interacting with other objects, or even that the hand is not interacting with anything. This results in an unbalanced dataset, and thus in the tendency of classifiers to predict the most represented class (i.e., the not-contact class). This trend can be mitigated by the use of class weights in the loss function. However, in presence of data scarcity, this strategy could end up in (mis)classifying uncommon motion patterns (i.e., inputs in regions of the pattern space with a low density of negative samples) as if they were face touches. As a consequence, a sufficiently diverse dataset is absolutely required to obtain a trustworthy classifier. The preliminary investigation led to the following outline of gestures:

Face-touches: bring the hand towards the face with slow and/or fast motions; touch diverse areas of the face (i.e., right and left side, forehead, chin); touch the face while sitting and while walking; touch the face with small and large angular variations at the elbow; prolonged touch, i.e., after the contact has occurred, the hand stops moving or continues moving on the face (e.g. during a rubbing).

Uncommon or misleading movements: random movements suddenly stopped without getting in contact with anything; movements pointing towards the ground and stopping after a non-negligible arm extension has been reached; fast spinning the wrist; go upstairs occasionally leaning the hand on a knee; bottom-up movements not ending up on the face; touch on a shoulder; walk with the arm parallel to the ground.

Common movements in Activities of Daily Living: rotational movements aimed at intentionally looking at the watch; hold the phone in the hand; type on a keyboard; browse the pages of a book; grasp and move objects; gesticulate; bring something to a position higher than the shoulder.

Participants to the data collection step were asked to perform spontaneous gestures, including as many as possible of the aforementioned actions.

Accelerometer data were acquired at 50 Hz by exploiting dedicated functions from Android API 29 (further details on the implementation can be found in the publicly available code). Differently from other approaches, we used only the accelerometer sensor. We did not use gyroscope and magnetic sensors because they drain the smartwatch battery 10-30 times quicker than the accelerometer [18], and are usually not available in low-cost devices.

All video recordings started before the beginning of the accelerometer data acquisition and were stopped after the end of the sensor acquisition. To facilitate the synchronization of video and acceleration data, as soon as the sensor started to acquire measurements, the background color of the smartwatch app became white and this color lasted for all the duration of the trial.

B. Data Processing

The first step in data processing was labelling the recorded data. Each video was visually inspected exploiting a simple webapp (HTML webpage with JavaScript) to examine video frames. Each frame could be labelled as *i)* neutral, *ii)* start contact, or *iii)* stop contact. The rationale behind this labelling choice relies on the need of identifying data sequences that start from a neutral position of the hand

Parameter	Value / Interval
LSTM cells	[1, 10)
hidden neurons	[2, 50)
learning rate	{0.001, 0.01, 0.1, 1.0, 0.0001, 1e-5, 1e-6, 1e-7}
max epochs	200
sequence length	[30, 100)
additional_time	[0.1, 0.8) · (sequence length) / (sampling rate)

TABLE I: Hyperparameters table.

(i.e., a position where the hand is not in contact with the face), contain the face-touch and stop at the end of the contact. The label *"start contact"* is useful to discriminate between data preceding the contact and data related to the actual interaction of the hand with the face. In so doing, no *a-priori* assumptions are made on the amount of data required to characterize the gesture, and this hyperparameter can be tuned in a later step. On the basis of the list of tuples (neutral, start contact, stop contact), a segmentation of the file containing all the acceleration data is performed, generating as many files as the cardinality of the list of tuples. In other words, the *i*-th file contains acceleration data belonging to the tuple (neutral$_i$, start contact$_i$, stop contact$_i$).

To retrieve data for training and validation sets, files related to each recording were splitted as follows: 80% for training, 20% for validation. As explained in the following, a separate dataset was acquired for testing purposes.

C. Neural Network Training Procedure

The most natural way to approach the training of an artificial neural network for the problem at hand was identified in training a Recurrent Neural Network (RNN) based on Long Short-Term Memory (LSTM) cells, which are capable of processing temporal sequences. This choice was also suggested by a preliminary visualization of the acceleration data (see Fig. 2), showing that instantaneous readings are not easily separable. On the contrary, we will show that by taking into account a sequence of readings, the problem becomes feasible. Best practice in Machine Learning prescribes to avoid arbitrary restrictions in the learning problem formulation, because it would result in a restriction of the configuration space where the learning process can take place (i.e., where the loss function is minimized). On the other hand, some architectural choices are necessary, because learning and model selection steps should not take a disproportionate amount of time. In this light, some LSTM cells were connected to a hidden fully-connected layer which, in turn, was connected to an output neuron. The weighted binary crossentropy was set as loss function (as typical of classification tasks with unbalanced dataset), ReLU (Rectified Linear Unit) and sigmoid were selected as activation functions of the dense and output layers, respectively. To evaluate the neural network performance, we chose the Matthews Correlation Coefficient

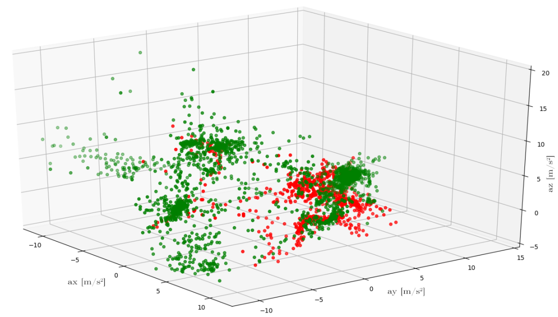

Fig. 2: Plot of some instantaneous acceleration readings. Red dots represent acceleration measured during face touch. Green color refers to other acceleration values.

(MCC)[1]. We did not set specific values for the number of LSTM cells, number of dense layer neurons, learning rate, number of training epochs. Instead, we identified intervals for these hyperparameters and performed a random search [19] within these intervals. Similarly, rather than selecting a fixed value, suitable intervals were defined also for the length of the temporal sequences fed to the network and for the *additional_time*, i.e. the time during which the hand stays in touch with the face after the contact has been established. The rationale behind this choice is that we want to let the model selection procedure define the relevant time interval describing the contact dynamics (which is related to force exchange, hence accelerations during time). In Table I, a summary of the intervals for the hyperparameters is provided.

The number of LSTM and dense layer neurons was limited to a small number so that the resulting model is suitable to the computational capabilities of smartwatches and DIY devices. Moreover, preliminary investigations showed that a higher number of LSTM cells usually made the optimization problem more prone to overfitting, resulting in a reduction of the generalization capability.

Our model selection procedure was based on the evaluation of the MCC value on the validation set. At the end of each epoch of each run, the MCC was computed on the validation set and, if an improvement occurred, the model was stored. Early stopping was implemented to reduce overfitting and save time, with a patience of 30 epochs and a hard limit of 200 epochs (total).

Being a classification task (i.e., detect whether or not the hand is in contact with the face), neural network training was performed by feeding the network with the pairs *(readings_sequence, target)*. Hence, the tuples detailed in Sec. II-B (neutral, start contact, stop contact) were translated to standard binary labelling at the sequence level: a sequence had target output 0 if the hand did not contact the face and 1 if during that sequence a face contact occurred. As explained few lines above, different runs can have different duration of

[1] In classification tasks, MCC is a meaningful measure of performance, especially for unbalanced problems. It accounts for true positives, true negatives, false positives and false negatives and assumes continuous values spanning from -1 to +1. The value -1 corresponds to an inverse classifier, 0 to a classifier performing random choices and +1 to an ideal classifier.

the temporal sequence. Hence, the input/target pairs effectively involved in the training were dynamically built once the parameter set was generated (and the aforementioned duration selected). From a practical point of view, a sequence with target output 1 lies in the interval [*t_s, start_contact_time* + *additional_time*], where t_s is the time instant such that the cardinality of the set of data belonging to the interval [*t_s, start_contact_time* + *additional_time*] is equal to sequence length. All the other sequences have target output 0. Neural network training exploited the Keras library (v2.4) [20].

D. Neural Network Evaluation Procedure

The procedure described in Sec. II-C was aimed at selecting a well-performing model. The selected model was then characterized on data never involved either in the training or in the validation process. Data labelling was performed similarly to what described in Sec. II-B, with the slight difference that label 1 was attached to all the sequences belonging to the interval [*start_contact_time, stop_contact_time*]. This choice was due to the fact that the data labelling in this step was used to evaluate the neural network effectivess in detecting face-contacts: from a user perspective, it is crucial to receive at least one alert during a face-contact, regardless of the exact moment in which it occurs during the contact. Moreover, this labelling choice was needed to have a fair comparison with different approaches to the problem.

To evaluate the model, we collected the response of the network, instant-by-instant, when fed with the readings coming from the accelerometer stream. In particular, we processed the accelerometer log by feeding in, at each time instant, the last seq_length readings. Given the sigmoid activation function, the network outputs continuous values spanning from 0 to 1. We considered as *contact* an output greater than the threshold 0.9. Moreover, we defined two metrics: True Alarm Rate (TAR) and False Alarm Rate (FAR). TAR is defined as the ratio between detected contacts and actual contacts. More specifically, a detected contact occurs when the network output is greater than the threshold for at least one time instant belonging to a time interval whose supervision is 1. FAR, instead, accounts for false positives: it is defined as the percentage of the number of suprathreshold network outputs over the total amount of samples in a time interval whose supervision is 0. False alarms occurring within $1.5\,\mathrm{s}$ after *contact_stop* events were discarded in the computation, due to the not-negligible length of the sequences fed into the network.

To visualize the neural network output synchronized with video and acceleration data, we developed a webapp which plays the streams and produces real-time plots. An example of data related to a face-contact is shown in Fig. 3 and Fig. 4.

III. Experiments

In this section, we report experimental results aimed at evaluating the No Face-Touch approach based on a recurrent neural network trained by following the methodology described in Sec. II. Moreover, we provide a comparison between the RNN approach and the approach proposed in [16]. Both methods take as input only data coming from the smartwatch accelerometer. To have a fair comparison between the methods, they were evaluated on the same dataset. All the involved subjects were right-handed and wore the smartwatch on their left wrist, in accordance with the fact that face touch events occur most of the times with the non-dominant hand [10]. This is particularly true during sedentary activities, like working at the PC, reading books, etc.

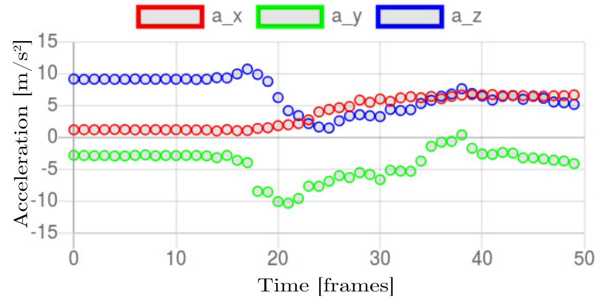

Fig. 3: Example of a temporal sequences of acceleration data related to a face-contact.

A. No Face-Touch Recurrent Neural Network Training and Evaluation

Accelerometer and video data for neural network training and model selection had an overall duration of about 40 minutes, and were acquired from a single subject. While a larger amount of supervised data should increase the performance of the proposed model, we noticed that such dataset may actually suffice to the aim. During recordings, the subject was asked to freely perform gestures accounting also for those reported in Sec. II-A. Data were processed as detailed in Sec. II-B and Sec. II-C.

The model selection process led to a recurrent neural network whose MCC was 0.92. Its architecture is described by 5 LSTM cells connected to 9 dense neurons, which in turn are connected to the output neuron. The learning rate (Adam optimizer employed) was 0.1. The length of the temporal sequence processed by the network is $1.58\,\mathrm{s}$, $0.89\,\mathrm{s}$ of which consists in what we called *additional_time*. In other words, when the network detects a contact, $0.69\,\mathrm{s}$ are related to data preceding the contact occurrence, while the remaining $0.89\,\mathrm{s}$ are related to the contact dynamics.

To evaluate neural network performance on data never involved in training and validation processes, held-out data were considered. Moreover, to have insights on the neural network generalization capabilities, other participants were recruited for testing purposes. Data acquisition involved 12 subjects (none of them involved in previous steps), 7 females and 5 males, with age ranging from 18 to 59 years old. Twenty recordings were acquired, with an overall duration of about 120 minutes. Data were recorded in conformity with the European General Data Protection Regulation 2016/679, stored on local repositories with anonymized identities (i.e., Subject 1, Subject 2, etc.), and used only for the post processing evaluation procedure. No sensitive data were recorded.

Method	FAR	TAR	TTA [s]
NFT_RNN	0.031 ± 0.023	1.000 ± 0.000	0.613 ± 0.265
NFT	0.093 ± 0.080	0.907 ± 0.045	0.534 ± 0.220
NFT rising 1σ	0.065 ± 0.032	0.813 ± 0.135	0.707 ± 0.210
NFT rising 2σ	0.042 ± 0.021	0.720 ± 0.151	0.781 ± 0.227
NFT rising 3σ	0.036 ± 0.025	0.649 ± 0.149	0.810 ± 0.260

TABLE II: Averaged experimental results on True Alarm Rate (TAR) and False Alarm Rate (FAR) computed on the test set according to different methods. NFT_RNN denotes the approach relying on the recurrent neural network, NFT the method in [16] without the rising check, the last three rows are related to the rising check accounting for possible noise. TTA stands for Time To Alert, i.e. the average time required for the method to detect an occurred face contact.

During experiments, participants were asked to freely perform gestures related to activities of daily life, accounting also for the gestures reported in Sec. II-A, e.g., gesticulating, reading a book, moving objects, etc. Video and accelerometer data were processed as detailed in Sec. II-B, and sequence labelling was performed as in Sec. II-D. Results on FAR (False Alarm Rate) and TAR (True Alarm Rate) are reported in Table II under the acronym of NFT_RNN.

B. Comparison

As a metric of performance, we compared our approach with the one proposed in [16], where the authors presented a rule-based method exploiting roll and pitch angles of the measured acceleration joined with a detection of hand rising motions. More specifically, a contact is deemed to occur when the hand is rising, and roll and pitch angles are in the so-called *unsafe range*, i.e. $\theta \in [-90, 70]$ and $\phi \in [-100, -30]$, respectively. The reference system related to acceleration measurements corresponds to the reference system shown in the Android API [21]. Hence, the hand is considered as rising when the first time derivative of the pitch angle is negative. Besides that, to discriminate between actual rising motion and false rising due to noise, we consider also to accumulate a rising cue only if the absolute value of the first pitch angle time derivative was greater than $\tilde{\sigma}$, where $\tilde{\sigma} \in \{1\sigma, 2\sigma, 3\sigma\}$, and σ denotes the standard deviation of the pitch first time derivative computed during a calibration phase (a zero-mean Gaussian distribution for the noise was assumed). Other details on the implementation of the comparison algorithm can be found in [16]. To achieve a fair comparison on the performance of the methods, the same data used to evaluate the No Face-Touch RNN were fed into the algorithm presented in [16]. Averaged experimental results on the comparison between the methods are reported in Table II, under the acronyms *NFT_RNN, NFT, NFT rising* $\tilde{\sigma}$.

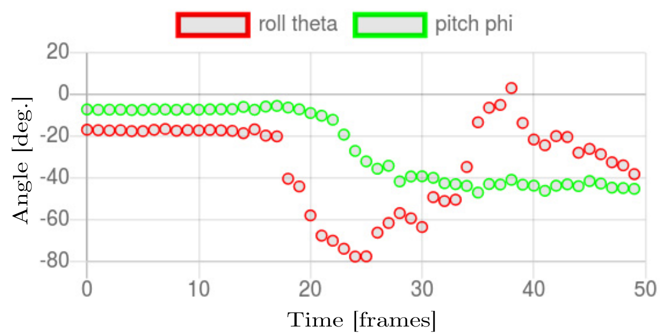

Fig. 4: Example of a temporal sequences of roll and pitch angles related to a face contact, whose acceleration data are shown in Fig. 3.

IV. Discussion

In this section, we discuss the experimental results provided in Sec. III, recalling that the aim of this work is to provide a satisfactory exploitation of data coming from the accelerometer sensor to detect occurrences of face touching. We hold to be true that acceleration data are satisfactorily exploited if they allow to detect face contacts with high accuracy without generating many false alarms, otherwise the usage of No Face-Touch would be compromised, due to a bad impact on the user's comfort during daily life activities.

A. On the Recurrent Neural Network

The recurrent neural network we trained takes as input the acceleration measured by the smartwatch, i.e., $\vec{a} = \vec{a_m} + \vec{g}$, where $\vec{g}$ is the gravity acceleration, and $\vec{a_m}$ is the acceleration resulting from the arm movement. Our hypothesis is that the neural network is capable of implicitly extracting the watch orientation and time derivatives from accelerometer readings. This claim is supported by the fact that short sequences (less than $1.0\,\mathrm{s}$) resulted in RNNs with very poor performance. Conversely, with longer sequences of acceleration values, motions ending up with a face contact can be learnt whether the action is followed by a rest or by a slow motion (e.g., a rubbing or head scratching).

By feeding the network with examples of contacts with the face and contacts with other objects, the network is capable of discriminating between different kind of contacts thanks to different watch orientation.

As a general consideration, we noticed that a high value of MCC can be achieved also with a greater number of LSTM cells. However, it is worth recalling that the aim of the No Face-Touch project is to run the app on smartwatches or low cost devices. Hence, battery consumption should be reduced as much as possible and LSTM modules are complex modules: one more reason to follow the "*Lex parsimonae*" principle. Another consideration concerns the length of the temporal sequence processed by the network. We noticed that longer temporal sequences usually led to a higher rate of false positives. It may be due to the fact that longer sequences does not allow the network to be focused on the dynamics of the contact, introducing in the same sequence pre-contact, contact-related, and post-contact data.

A further consideration regards false positives: It must be recalled that no information concerning the relative position of the arm with respect to the human body is encapsulated in the data. The only information provided consists in roll and pitch angles (explicitly computed in [16] or implicitly used by the RNN). Even more importantly, roll and pitch angles are related to the wrist orientation, but no information at all is provided on the hand pose while performing gestures. Hence, the measurements corresponding to a face-touch with a given hand orientation can correspond to a different kind of interaction with a different hand orientation. This unavoidable noisy data labelling results in situations that can fool the neural network. Some improvements could be obtained by exploiting also the gyroscope, but it is not always available on smartwatches, and is responsible of heavier power consumption. Furthermore, as shown in Table II, the RNN shows a good generalization capability on different users, although trained on data from a single participant.

B. On the Comparison

Results reported in Table II show that acceleration data can be successfully exploited as input to the RNN, achieving a false detection rate of 3.1% and the highest score for true detections. Moreover, taking into account the methods achieving low false detection rates, the average time that NFT_RNN required to detect the contact (Time To Alert) is the lowest among the results. As it can be easily envisaged, developing technologies for gesture detection with low false alarm rate is a requirement for effective and comfortable use of technologies. Results for the NFT approach do not meet this condition (9% false detections rate). The idea of reducing the false contact occurrences by filtering the rising signal (1σ, 2σ, 3σ confidence interval exploitation) results in a decrement of the false detections as well as in a decrement of the true detections, joined with an increment of the Time To Alert (TTA) value. Hence, we consider this workaround a not sufficiently satisfying way to approach the problem.

A further consideration can be done about the TTA value. As a matter of fact, to reduce the probability of being infected by SARS-CoV-2 because of unaware face-touch occurrences, a fast detection of occurred contacts is necessary, so that also brief face-touches can be notified. Hence, a low TTA is highly desirable. With the same motivation, attaining a very high true detections rate (TAR) is extremely relevant. As shown in Table II, the approach based on the neural network achieves the best result.

C. Generalization to Different Devices

The core of the No Face-Touch project relies in our strong belief that events like the SARS-CoV-2 pandemic can be faced only by acting simultaneously on two complementary aspects: providing people with tools preserving their heath and, at the same time, promoting (explicitly and implicitly) proactivity of individuals in taking care of their health, with particular focus on their feeling of agency. We also believe that situations like the one we are living can be faced only in a collaborative framework, where different communities of researchers make contributions related to their field. In this light, No Face-Touch wants to be a small, but effective, contribution to personal and collective health, arising from the haptic field. The No Face-Touch project involves devices of daily usage, for an easy-to-spread human-device task accomplishment, whose aim is to preserve user's health. This is why we decided to make publicly available the code for data processing, neural network training and app upgrading. Other considerations can be done on the type of device running the proposed approach. Although smartwatches are gradually spreading, they are still expensive devices, with small market share in developing countries. Low cost fitness devices (such as Fitbit, Xiaomi Mi Band, Samsung Galaxy Fit, etc.) work with companion smartphones: in this case, the computations for the No Face-Touch RNN could be performed on the smartphone side. Besides that, cheaper (and more engaging) solutions can be found, and *ad hoc* devices can be realized in the DIY spirit. No Face-Touch RNN, indeed, can potentially run on a microcontroller connected to a triaxial accelerometer. The hardware could be enclosed in a little case designed via CAD and 3D printed, examples of which are available at [22]. The opportunity of using DIY devices makes this solution suitable for children and large-scale deployable.

V. Conclusions

In this work, we propose a ready-to-use solution to reduce the occurrence of face-touches, contributing to limit the spreading of SARS-CoV-2. In this light, the No Face-Touch project is aimed at providing users with multiple solutions, so that anyone can choose the one more suitable to its preferences. Any device embedding an accelerometer meets the algorithm requirements, since accelerations allow a simple physical description of the gesture, without additional features. According to this approach, the neural network we trained and tested proved to be the most effective way to exploit such data, achieving a very high true detections rate, a low false detection rate (mean 3.1%) and a short time to detect the contact (mean $0.6\,\mathrm{s}$). Furthermore, the trained model proved to have sufficient generalization capabilities to correctly process previously unseen data generated also from different users. With a limited effort, No Face-Touch can be exploited in a wide group of smartwatches, smartbands, or alternatively, a simple DIY bracelet can be built with off-the-shelf electronic components. Since No Face-Touch is an open project, all the software mentioned in this manuscript was developed by the authors and is freely available in a public repository, released under GNU GPL software license[2]. Alongside of the CAD models, we share the trained model, besides the code for neural network training and app developing, believing in the power of cooperation and GNU philosophy.

[2] https://github.com/sirslab/NoFaceTouch

References

[1] WHO - World Health Organization, "Coronavirus disease (COVID-19), Situation Report – 102." Accessed: 2020-10-10.

[2] M. Nicas and D. Best, "A Study Quantifying the Hand-to-Face Contact Rate and Its Potential Application to Predicting Respiratory Tract Infection," *Journal of occupational and environmental hygiene*, vol. 5, no. 6, pp. 347–352, 2008.

[3] G. Di Giuseppe, R. Abbate, L. Albano, P. Marinelli, and I. F. Angelillo, "A survey of knowledge, attitudes and practices towards avian influenza in an adult population of Italy," *BMC infectious diseases*, vol. 8, no. 1, p. 36, 2008.

[4] B. Winther, K. McCue, K. Ashe, J. R. Rubino, and J. O. Hendley, "Environmental contamination with rhinovirus and transfer to fingers of healthy individuals by daily life activity," *Journal of medical virology*, vol. 79, no. 10, pp. 1606–1610, 2007.

[5] J. M. Gwaltney Jr and J. O. Hendley, "Transmission of experimental rhinovirus infection by contaminated surfaces," *American Journal of Epidemiology*, vol. 116, no. 5, pp. 828–833, 1982.

[6] J. Gu, Y. Zhong, Y. Hao, D. Zhou, H. Tsui, C. Hao, Q. Gao, W. Ling, and J. T. F. Lau, "Preventive Behaviors and Mental Distress in Response to H1N1 Among University Students in Guangzhou, China," *Asia Pacific Journal of Public Health*, vol. 27, no. 2, pp. NP1867–NP1879, 2015.

[7] "As Many As 30 Percent Of Travelers Don't Wash Hands After Using Public Restrooms At Airports ." Accessed: 2020-04-04.

[8] A. Macias, A. De la Torre, S. Moreno-Espinosa, P. Leal, M. Bourlon, and G. Ruiz-Palacios, "Controlling the novel A (H1N1) influenza virus: don't touch your face!," *Journal of Hospital Infection*, vol. 73, no. 3, pp. 280–281, 2009.

[9] Y. L. A. Kwok, J. Gralton, and M.-L. McLaws, "Face touching: A frequent habit that has implications for hand hygiene," *American journal of infection control*, vol. 43, no. 2, pp. 112–114, 2015.

[10] N. Zhang, W. Jia, P. Wang, M.-F. King, P.-T. Chan, and Y. Li, "Most self-touches are with the nondominant hand," *Scientific Reports*, vol. 10, 2020.

[11] F. Noroozi, K. Kaminska, C. Corneanu, T. Sapinski, S. Escalera, and G. Anbarjafari, "Survey on emotional body gesture recognition," *IEEE transactions on affective computing*, 2018.

[12] D. W. Woods, C. T. Wetterneck, and C. A. Flessner, "A controlled evaluation of acceptance and commitment therapy plus habit reversal for trichotillomania," *Behaviour research and therapy*, vol. 44, no. 5, pp. 639–656, 2006.

[13] O. M. Tanaka, R. W. F. Vitral, G. Y. Tanaka, A. P. Guerrero, and E. S. Camargo, "Nailbiting, or onychophagia: a special habit," *American Journal of Orthodontics and Dentofacial Orthopedics*, vol. 134, no. 2, pp. 305–308, 2008.

[14] K. Al Hawsawi, K. Al Aboud, and V. Ramesh, "Dermatophagia simulating callosities," *Dermatology and Psychosomatics/Dermatologie und Psychosomatik*, vol. 4, no. 1, pp. 42–43, 2003.

[15] J. T. Rapp, R. G. Miltenberger, and E. S. Long, "Augmenting simplified habit reversal with an awareness enhancement device: Preliminary findings," *Journal of Applied Behavior Analysis*, vol. 31, no. 4, pp. 665–668, 1998.

[16] N. D'Aurizio, T. Lisini Baldi, G. Paolocci, and D. Prattichizzo, "Preventing undesired face-touches with wearable devices and haptic feedback," *IEEE Access*, vol. 8, pp. 139033–139043, 2020.

[17] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," in *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 80–89, 2018.

[18] A. Sucerquia, J. D. López, and J. F. Vargas-Bonilla, "Real-life/real-time elderly fall detection with a triaxial accelerometer," *Sensors (Basel, Switzerland)*, vol. 18, April 2018.

[19] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 281–305, 2012.

[20] F. Chollet *et al.*, "Keras: The python deep learning library," *ascl*, pp. ascl–1806, 2018.

[21] Android API, "Sensor Coordinate System." Accessed: 2020-10-09.

[22] "NoFaceTouch - GitHub Public Repository." Accessed: 2020-10-10.