Automated evaluation of COVID-19 risk factors coupled with real-time, indoor, personal localization data for potential disease identification, prevention and smart quarantining

J. Barabas*, R. Zalman† and M. Kochlan†
*Department of Electromagnetic and Biomedical Engineering, Faculty of Electrical Engineering and Information Technology, University of Zilina, Slovakia
†Department of Technical Cybernetics, Faculty of Management Science and Informatics, University of Zilina, Slovakia
Email: jan.barabas@feit.uniza.sk

***Abstract*—Since the beginning of the current COVID-19 pandemic, more than five million people have been infected and the numbers are still on the rise. Early symptom detection and proper hygienic standards are thus of utmost importance, especially in venues where people are in random or opportunistic contact with each other. To this end, automated systems with medical-grade body temperature measurement, hygienic compliance evaluation and individualized, person-to-person tracking, are essential, not only for disease spread intervention and prevention, but also to assure economic stability. Herein, we present a system that encapsulates all of the mentioned functionality via readily-available components (both hardware and software) and is further enhanced with preliminary RTLS data acquisition, enabling post-symptom detected, person-to-person interaction identification to asses potential infection vectors and mitigate further propagation thereof by means of smart quarantine.**

***Keywords*—COVID-19; SARS-CoV-2 virus; Raspberry Pi; medical-grade contactless temperature measurement; face mask; RTLS; smart quarantine; infection vectors; neural network**

I. Introduction

On the brink of new year 2020, the world witnessed a new and deeply concerning situation that only a few have anticipated to have such serious impact on a world-wide scale. What started out as a local outbreak in Wuhan, China, quickly became an uncontrolled, world-wide pandemic, mainly due to slow precautionary measures taken by local governments and large underestimation of the emerging threat in the form of the COVID-19 disease. Since then, however, preventive measures have been implemented, in order to at least slow down the already devastating impacts – both social and economic, across the whole world [1].

To this end, devices either for treatment (e.g. ventilators) or for automated, proactive detection and monitoring of possible virus carriers – have become essential tools of trade in the fight against the pandemic and further spread prevention. While reverse transcription polymerase chain reaction or antibody testing is still relatively time consuming and costly, a simple evaluation of the body temperature at various checkpoints using medical-grade infrared sensors and control of proper hygienic standards, coupled with personalized, indoor location data, could be a quick and effective way of identifying and controlling the spread of the disease within closed environments, such as workplaces or factories.

II. Hardware Components

A work in progress prototype of the proposed automated temperature and hygienic compliance testing device is shown in Fig. 1. The device is mounted on a fixed stand before the main entrance to the workplace or at various checkpoints, e.g. entrances between halls.

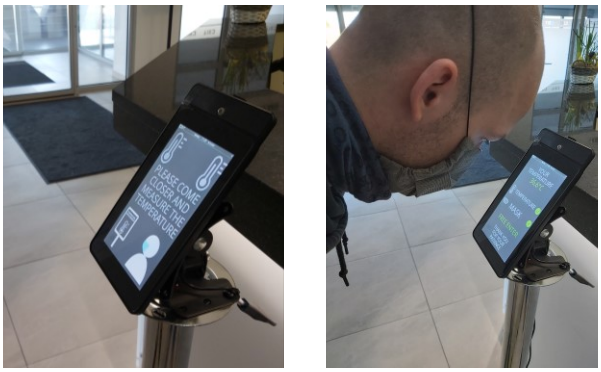

Fig. 1. Installed device prototype at workplace entrance with GUI showing textual information in stand-by mode and measurement results when test person is present

The device is comprised of the following main parts, with emphasis on minimizing manufacturing costs:

- Raspberry Pi 4 Model B w/4GB RAM [2] – core of the whole system, enables real-time processing of all image and sensor data;

This work was supported by the Slovak Research and Development Agency under the contract No. APVV-14-0519 and APVV-18-0167. This publication was also realized with support of Operational Program Research and Innovation in frame of the project: ICT products for intelligent systems communication, code ITMS2014+ 313011T413, co-financed by the European Regional Development Fund.

- Raspberry Pi Camera Module v2 – acquisition of image data (1640 x 1232 pixels) for detection of the face mask and on-screen face navigation;
- MLX90614 infrared thermometer in TO-39 package design, with medical-grade accuracy (±0.1 °C) in human body temperature range [3];
- VL53L0X Time-of-Flight Distance Sensor for presence detection [4];
- 7" capacitive Raspberry Pi LCD module with 800x480px resolution;

The device prototype is connected either via LAN or WiFi connection to a local area network in order to write necessary information to the SQL database server, which stores acquired measurement data and enables correlation with positional data from the independently-operated real-time locating system (RTLS) database. The same LAN subnet is also used to operate the opening of locks once safe conditions are met by sending specially formulated HTTP post requests to specific door lock nodes with self-integrated Arduino relay devices.

The independent RTLS system, used primarily for other monitoring purposes (e.g. productivity assessment or optimization of work processes), is based on commercial devices available from Sewio and utilize ultra-wideband (UWB) technology which enables very precise localization – up to several centimeters in optimal conditions.

Information is processed in real-time using RTLS studio and positional data is stored in a MySQL database. Anchors are placed throughout the building and personal tags, each assigned to specific person, are tracked and recorded in real time – and can also be visualized and further processed. More information about the commercial RTLS system from Sewio can be found in [5].

III. Software Components and Algorithms

The following open-source programming languages and platforms (except for the RTLS system) are employed in the design:

- Python – running natively on the Raspberry Pi, processing of acquired sensor data (camera, IR thermometer and TOF sensor) using the following libraries/packages:
    - OpenCV (esp. Deep Neural Network module) [6],
    - Keras framework & Tensorflow (for deep learning),
    - imutils,
    - dlib;
- Kivy - open source Python library for rapid development of applications with innovative user interfaces [7].

Fig. 2 shows the general workflow of the evaluation.

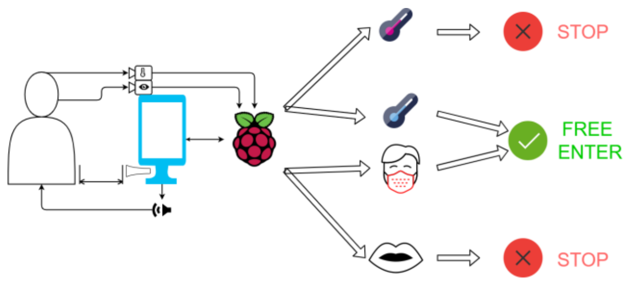

Fig. 2. Generalized workflow of the constructed prototype device

In order for the evaluated person to be granted access, two criteria must be satisfied simultaneously:

- the temperature must be below <37.0 °C,
- the person must wear a hygienical/protective mask.

The following paragraphs describe the individual workflows in more detail.

A. *Presence detection workflow*

Presence is detected within a 4 meter perimeter using the VL53LoX TOF sensor. Data from the sensor is processed using Python only. Fig. 3 shows the standard presence detection workflow in strict mode – other modes can be enabled for relaxed conditions, e.g. to exclude mask detection or disable logging of certain events within the remote database. When no person is present, the device is in standby mode to preserve power – showing only general information (Fig.1 - left).

A progress bar is also used to navigate the user to the measurement terminal – when the distance between the terminal and the face is > 0.5 meters, the progress bar reports 0% - gradually, as the face is located 20 cm or less from the device, the progress bar reports 100% and subsequent processing (mask detection, temperature measurement) is initiated.

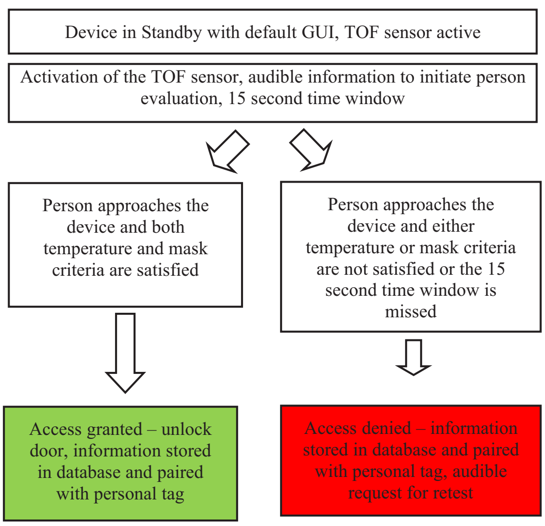

Fig. 3. Presence detection workflow in standard operating mode

B. Mask detection algorithm

The mask detection algorithm is a hybrid solution utilizing both neural networks (NN) and feature vector description based on histogram of oriented gradients (HOG) approach [8]. While images are acquired with a spatial resolution of 1640 x 1232 pixels, to improve performance, they are downscaled to 384 x 288 pixels for further processing.

As a base for face recognition, already trained models freely available with a total of approximately 300 000 faces have been used, based on datasets from e.g. [9,10] and also own datasets. Specific recognition for masks and respirators has been added with further training performed on 220 image sets of persons wearing a mask, respirator, or similar device. This additional set has been manually created for this purpose from freely available picture libraries online. This set is a work in progress and is being continuously updated based also on images acquired from the device prototype for improved recognition of masks, respirators and similar devices.

The resulting working copy of network model used for this case has been stored in Caffe framework format and processed in OpenCV Deep Neural Network module.

However, since the success rate when using the neural network was relatively low, we decided to change the strategy of detection and use the neural network solely for initial face detection – once the face is detected in the field of view of the device, we switch to HOG-based approach for further face feature processing and trait detection, as detailed in Fig. 4.

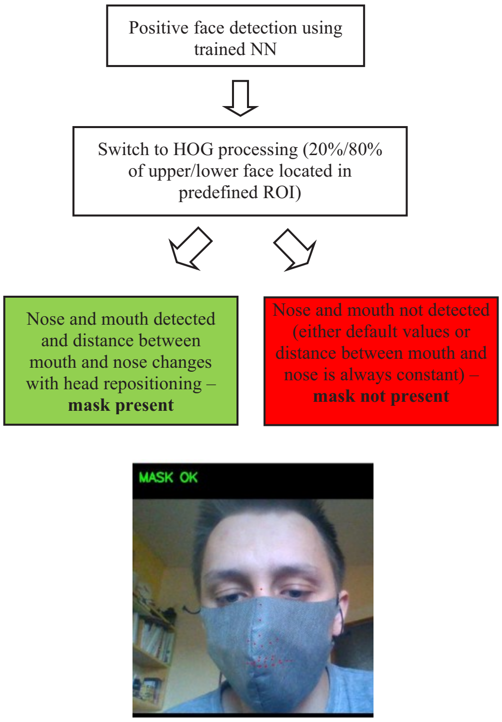

Fig. 4. Simplified workflow of the mask detection algorithm and processed image with detected mask (bottom)

The performance of the proposed solution has been evaluated on over 300 trials of 22 volunteers with different faces and different masks and protective devices. The masks had mainly white and black textile material without pattern but also marginal number of masks with pattern have been evaluated. Table 1 shows the overall success ratio of mask identification based on the present model. Average identification time for given mask group is also listed.

TABLE I. OVERALL SUCCESS RATIO AND IDENTIFICATION TIME

Mask Type	*Total identification trials*	*Successful identification trials*	*Overall identification success rate*	*Average identification time per image*
White	153	149	*97.4%*	*0.22s*
Black	129	125	*96.9%*	*0.23s*
Pattern	35	23	*65.7%*	*0.26s*

Given the computing constraints of the employed Raspberry Pi platform, the detection algorithm is able to process approximately 4 frames per second of data with 75% utilization of the main CPU when all sensors are active.

The performance of Raspberry Pi 4 module is thus sufficient for real-time mask detection. Although, in order to obtain long term stability in detection accuracy, larger training set of faces with masks and respirators and other similar protective devices should be considered. Despite promising results, ambient light conditions must be close to ideal (laboratory) conditions. Otherwise, mask detection success ratio decreases significantly. Ambient light conditions and impact thereof on mask identification success level are subject to future development.

Even though the selected Raspberry Pi 4 platform has shown sufficient performance, there are many platforms that compete with this popular platform, such as NVIDIA Jetson Nano or Rock Pi N10. The following table (table 2) briefly compares the mentioned development platforms.

TABLE II. COMPARISON OF SELECTED DEVELOPMENT PLATFORMS SUITABLE FOR IMAGE RECOGNITION AND ARTIFICIAL INTELLIGENCE

Attribute	*Rock Pi N10 [11]*	*Jetson Nano [12]*	*Raspberry Pi 4 [2]*
CPU	Dual Cortex-A72 1.8GHz & Quad Cortex-A53 1.4GHz	Quad Cortex-A57 1.43 GHz	Quad Cortex-A72 1.5 GHz
GPU	Mali T860MP4	NVIDIA Maxwell	Broadcom VideoCore VI
RAM	up to 8GB LPDDR3	up to 4GB LPDDR4	up to 4GB LPDDR4
LAN	Gigabit Ethernet	Gigabit Ethernet	Gigabit Ethernet
WiFi	-	-	802.11b/g/n/ac
Bluetooth	-	-	5.0
Display	HDMI 2.0	HDMI 2.0 + eDP 1.4	2x uHDMI
Video Encoding	H264 (1080p30)	H264 (1080p30)	H264/H265 (4Kp30)
Video Decoding	H264 (1080p60) H265 (4Kp60)	H264 (1080p60) H265 (4Kp60)	H264/H265 (4Kp60)
Price	**130.00 €**	**100.00 €**	**50.00 €**

It should be pointed out that Rock Pi N10 utilizes a neural network processing unit. This feature makes Rock Pi N10 superior in comparison with other mentioned platforms. Nonetheless, the employed Raspberry Pi platform performance, wide range of connectivity options and especially low cost justify choice thereof as a development platform for this project.

C. *Temperature measurement*

Proper temperature evaluation is critical in our prototype. Measurements are based on data obtained from the MLX90614 infrared thermometer, which has a sensitivity of ±0.1 °C when measuring in human body temperature range. Temperature is measured only when the head is located 20 cm or less from the device also within the defined ROI – in order to improve the accuracy of the measurement and obtain constant results. The initial algorithm evaluated the temperature based on the maximum value obtained from 10 consecutive measurements – the final version of the algorithm discards maximum and minimum values and computes the mean value from the resulting values. The mean value was manually compared with a certified thermometer reading and the average temperature delta was found to be within ±0.2°C based on more than 1000 individual measurements during a two month testing period.

It is also important to note that temperature measurements are also influenced by ambient temperatures to which the person was previously subject (e.g. arrival to work with sub-zero outside temperatures), however, when placed in multiple locations in the workplace (e.g. between different rooms or halls, when entering the elevator, etc.), it is possible to dynamically record and evaluate temperatures and intervene momentarily according to current state of health of the person.

D. *Preliminary RTLS integration*

A significant benefit of the proposed device prototype is the combination of measurement results with accurate indoor tracking of movement of individual persons using RTLS from Sewio [5]. This enables the localization of possible suspect person in real-time (e.g. when abnormal temperature is found during day and not at time of entry) and also allows us to review previous interactions with people.

This functionality is not directly integrated within the device prototype but is accessible via simple web interface developed in HTML/PHP. The current implementation allows the end user to textually evaluate the following data related to the measurements:

- Display saved records (tag ID/user, measured temperature, mask compliance, location ID and coordinates, time and date of measurement) from arbitrary time frame;
- Display interaction between specific tags (persons) in a predefined radius (cm precision) in arbitrary time frame;

Each of the above operations performs SQL queries between the device database and the RTLS database – based on user-defined parameters (e.g. tag number, temperature) and returns textual results, which can be exported to CSV format for further processing. We are currently working on adding graphical elements to the UI along with graphical representation of the tag spatial coordinates based on the building layout.

IV. Conclusion

Given the current dire epidemiological situation and enormous socio-economic impacts, devices that can either help identify or prevent further spread of COVID-19 disease are much sought for. The proposed device prototype described in this paper can help in two ways – via automatic evaluation of bodily temperature at various checkpoints and by enforcing proper hygiene standards related to face masks. Additional integration with RTLS systems, which is currently under development, can help gain valuable insight into personal interaction and, in case of confirmed health risk, help pinpoint persons which were in close contact with the suspected individual (smart quarantining).

References

[1] "World Health Organization Coronavirus disease (COVID-2019) situation reports website", available online [10.4.2020] at https://www.who.int/emergencies/diseases/novel-coronavirus-2019/situation-reports

[2] "Raspberry Pi 4 Model B website", available online [20.3.2020] at https://www.raspberrypi.org/products/raspberry-pi-4-model-b/

[3] "Datasheet for Melexis MLX90614", available online [20.3.2020] at https://www.melexis.com/-/media/files/documents/datasheets/mlx90614-datasheet-melexis.pdf

[4] "Datasheet for - VL53L0X", available online [20.3.2020] at https://www.st.com/resource/en/datasheet/vl53l0x.pdf

[5] "SEWIO UWB Real-Time Location System (RTLS)", available online [5.4.2020] at https://www.sewio.net/real-time-location-system-rtls-on-uwb/

[6] "OpenCV packages for Python", available online [10.4.2020] at https://pypi.org/project/opencv-python/

[7] "Kivy – Cross-platform Python Framework for NUI development", available online [10.4.2020] at https://kivy.org

[8] C. Sagonas, E. Antonakos, G, Tzimiropoulos, S. Zafeiriou, M. Pantic. 300 faces In-the-wild challenge: Database and results. Image and Vision Computing (IMAVIS), Special Issue on Facial Landmark Localisation "In-The-Wild". 2016.

[9] Yang, Shuo and Luo, Ping and Loy, Chen Change and Tang, Xiaoou, IEEE Conference on Computer Vision and Pattern Recognition (CVPR). WIDER FACE: A Face Detection Benchmark, 2016.

[10] Ge, Shiming and Li, Jia and Ye, Qiting and Luo, Zhao, Detecting Masked Faces in the Wild With LLE-CNNs, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[11] "ROCK Pi N10 website", available online [28.5.2020] at https://wiki.radxa.com/RockpiN10

[12] "Jetson Nano Developer Kit website", available online [28.5.2020] at https://developer.nvidia.com/embedded/jetson-nano-developer-kit