Smart scan of medical device displays: user validation

Pedro Lobo*[a, b], Alberto Simões[a, b], Pedro Morais[a, b], João L. Vilaça[a, b]
*pjlobo@ipca.pt
[a]2AI, School of Technology, IPCA, Barcelos, Portugal; [b]LASI-Associate Laboratory of Intelligent Systems, Guimarães, Portugal

ABSTRACT

Continuous health monitoring is an increasingly prominent concept aimed at promoting early diagnosis of health issues and tracking physiological changes over time. The democratization of certain medical devices, such as blood pressure monitors, oximeters, and glucometers, in the home environment, as well as the adoption of wearable systems such as smart bands and smartwatches, has been crucial in this regard, significantly contributing to the continuous acquisition and monitoring of physiological data. Unfortunately, many of these devices lack connectivity with mHealth applications, and those that do are generally limited to manufacturers' applications. As a result, creating an mHealth system that consolidates all medical data from patients is not possible, making management and diagnosis more difficult. This fragmentation is burdensome for patients, reducing their adherence and the frequency with which they record vital metrics. To address this issue, we previously developed the Smart Scan of Medical Device Displays to standardize the recording of medical data acquired by patients within a single mHealth application. In this work, we adapted this Smart Scan system for a mobile environment by developing a Kotlin library, integrating it into a demo app, and testing it with a group of 20 volunteers. The results obtained were promising, indicating a detection and classification rate of 96.3%, as well as a preference for data entry with our system of 18 out of 20 volunteers.

Keywords: mHealth, AI, continuous health monitoring, blood pressure, imaging informatics.

1. INTRODUCTION

Globally, in 2019, Europe and North America had the highest percentage of the aged population, with 18% of the population being 65 years or older. In 1990, 54 million people were aged 80 years or older globally, and by 2019, this number had nearly tripled to 143 million [1], [2]. This trend, particularly in Europe, is expected to continue growing at an even faster rate. As widely known, aging is directly linked to the deterioration of individuals' physical and cognitive functioning and their propensity to rely on conventional healthcare methods [3], [4]. With this issue in mind, it is imperative to act and allocate more efforts to the field of remote health care to avoid overwhelming traditional healthcare systems as we currently know them [5]. By further implementing remote health care, we can not only provide closer and personalized monitoring of individuals but also use it continuously throughout people's lives as a preventive tool.

The availability of various medical devices in the home environment, such as blood pressure monitors, oximeters, and glucometers, combined with the adoption of wearable systems, has enabled the routine analysis of physiological and biometric parameters in everyday life. The quantity and frequency of these measurements offer a clearer view of physiological changes over time and support diagnostics through machine learning model training, allowing for subject-to-subject transfer frameworks [6]. However, this is achievable only if all medical data for an individual can be centralized within a single mHealth application. Currently, this is not possible because of the fragmentation of devices across different manufacturers' ecosystems or their lack of connectivity. Additionally, it is essential to promote the performance and recording of these analyses within a unified mHealth application, a concept that our research team has already explored in relation to other medical data, such as blood test analyses [7].

To address this issue, we previously proposed an intelligent scanning system designed to simplify the collection of data displayed on various medical device screens by recognizing values and optionally integrating them with centralized databases through open protocols [8]. Like our work, other studies have developed comparable systems, with a particular focus on devices with 7-segment displays [9] or supporting only a limited number of device types [10]. Moreover, a common issue is that none of these systems have been successfully implemented on smartphones.

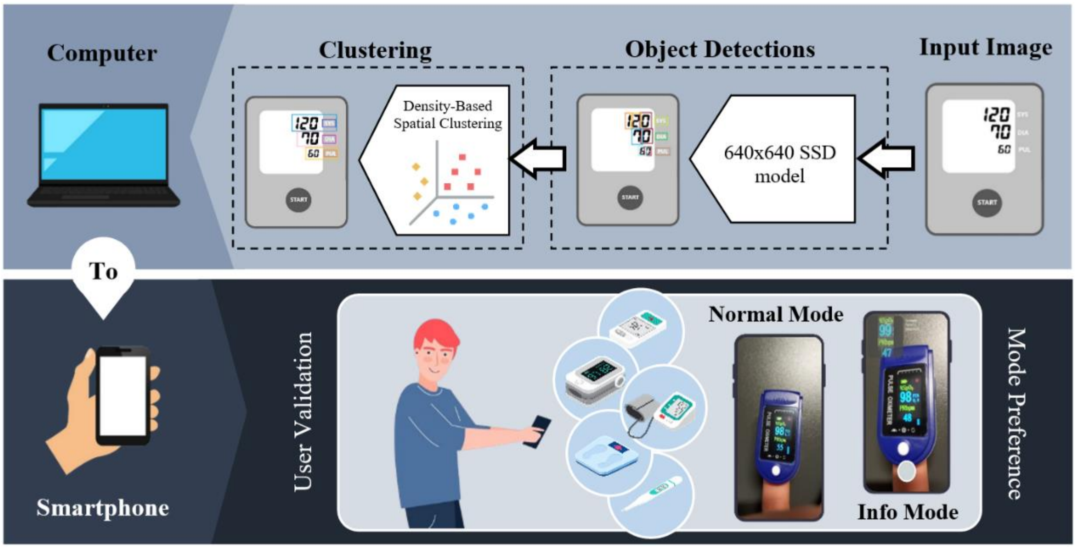

Figure 1. Overview of the proposed work.

In this work, we build upon the scan system developed in our previous study [6], adapting it for a mobile environment and subsequently validating it with a group of volunteers. The conversion process involved not only porting the Smart Scan system to a mobile environment but also developing it as a library, integrating it into an mHealth application, and implementing two scanning modes, the normal and info mode. These modes were tested by users to collect their feedback.

Henceforth, this paper is structured as follows: Section II presents the framework through the introduction of the methods. Section III details the implementation aspects. Section IV describes the experiments conducted. Section V presents the results obtained, while Section VI provides an analysis of these results. Finally, Section VII concludes the paper with final remarks.

2. METHODS

As a brief overview of the previously developed work [8], the scanning system involved training object detection models to identify the digits displayed on medical device screens. Following this, a condition tree based on clustering process, specifically Density-Based Spatial Clustering, was applied to group the digits via analysis, as many devices simultaneously present analyses of different physiological parameters. Building on this project, in this paper, we have converted the system to a mobile environment, specifically for Android (Kotlin), by developing a library that was integrated into a demo mHealth application (see Figure 1).

While some methods, such as the clustering algorithm, are directly transferred, others, such as the object detection model, required adjustments to facilitate the scanning process in practical terms. Specifically, clustering was implemented via the Kotlin Statistics Library[1], which includes the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm. Conversely, the object detection model, 640x640 SSD, needed adaptations before being converted to mobile (section 2.1).

2.1 Object detector

In practical terms, the efficiency of a scanning process is not only related to the quality of the method itself but also to framing the screens of the medical device being digitalized with the image captured by the mobile device's camera. To improve this alignment, we modified the original model to provide feedback to the user to assist in the scanning process. With this assistance, the scanning process is intended to be faster and more accurate through more adequate framing of the information to be extracted. Specifically, framing concerns the distance between the camera and the screen of a medical device. To achieve this, the SSD Mobilenet v2 640x640 [11], [12] was retrained under the same conditions, but

---

[1] https://haifengl.github.io/api/kotlin/smile-kotlin/smile.clustering/dbscan.html, (2024)

this time for detecting and identifying sixteen labels: medical device screens, digits (10), and units (5). On the basis of the bounding box area of the screens detected in pixels and the image resolution acquired by the mobile device's camera, the scan distance is estimated to provide feedback if a threshold distance is exceeded. If the screen is not close enough to allow feasible extraction of the screen information, feedback is given to the user to readjust the medical/mobile device.

2.1.1 Single image processing

The single image process receives, as input, the image (Bitmap) to be processed, the rotation (Int) at which it was acquired (to be corrected), and the device or devices being analyzed. Device selection is managed through an 'int' input variable ranging from 0-6. If the value '1' is provided, the device type detection is automatic; however, we also offer the option to process devices individually. The function returns two outputs: an image (Bitmap) and a list (List<Float>). The image output includes the region of the device's screen with detected objects and their bounding boxes.

2.1.2 Live camera processing

Live camera processing consists of real-time processing of images from the camera of the mobile device. Live camera processing can be executed in two different ways: normal mode or Info mode. While normal mode functions like a QR code scanner, automatically ending the live camera session once it detects and verifies the analysis by ensuring that the grouping of the digits aligns with the expected range for each type of analysis, the Info mode displays the area of interest containing the analyses and aggregated digits in the upper left corner of the smartphone screen. It only terminates the scanning process when the user manually signals through a button, typically after confirming that the analyses shown in the upper left corner are correct.

3. IMPLEMENTATION DETAILS

3.1 Dataset and training

The training and testing datasets used in this project were the same as those used to train and evaluate the scanning system presented in previous work [8]. However, a label corresponding to the bounding boxes of the devices' screens was added to the dataset manually. The model was trained for 50000 steps and resulted in a loss of 0.2772. For training, a workstation with a NVidia GTX 1070 8GB, an Intel Core i7-8700, and 32 GB of RAM was used, running with the NVIDIA libraries CUDA 11.0 and CUDNN 8.1.0. The TensorFlow Model Garden API (v2)[2] was used to train and validate the model.

3.2 Conversion

As previously mentioned, the conversion was focused primarily on the transposition of two methods. These methods are the object detector, which detects and distinguishes digits, acronyms/units, and displays, and the DBSCAN cluster, which integrates a condition tree to aggregate digits via analyses. For the retrained object detector model, after applying the aforementioned modifications, conversion/compression was necessary to make it lighter, faster, and especially compatible with mobile devices. After converting these two methods to Kotlin, we proceeded to join them to replicate the previously developed system. This integration resulted in a function that takes a single image as input and outputs the extracted analysis. To enable operation with a live camera, we used the CameraX library[3] for camera management and executed the scanning system via a callback to extract the analyses present in the camera images.

4. EXPERIMENTS

4.1 Demo mHealth application

A mHealth application was developed that incorporates the library that integrates the Smart Scan of Medical Device Displays. This application not only enabled the two scanning modes (Normal and Info) but also included a manual data entry feature for validation purposes.

4.2 Environment

To carry out the experiments, we set up a circuit composed of four medical devices: a blood pressure meter, an oximeter, a thermometer, and a scale. The glucometer was not selected given that it is invasive, and it would be likely that most

[2] https://github.com/tensorflow/models, (2024)
[3] https://developer.android.com/media/camera/camerax?hl=pt-br, (2024)

volunteers would refuse to collaborate if that device were considered. Each volunteer used each of the devices and recorded the measured values in three different possible ways: 1) inserting the analysis manual; 2) using the smart scanning system – normal mode, and 3) using the smart scanning system - Info mode. The selected devices were common devices available on the market.

4.3 Validation Test

The validation of the Smart Scan of Medical Device Displays was carried out using the demo mHealth application in which it was integrated. The evaluation process involved asking a group of volunteers to complete a circuit comprising four medical devices, performing the necessary analyses on each. After completing the analysis, the volunteers were first asked to manually enter the results into the mHealth application. In the second phase, they were required to input the analyses via live camera processing in normal mode. During this phase, we recorded whether the volunteers were able to successfully enter the data through this method and the time taken for the entry, which included the start-up time for the Smart Scan and the time spent framing the camera with the information to be read. Additionally, the volunteers were subsequently asked about their preference between manual entry and automatic entry in normal mode. In the third phase, the volunteers used live camera processing in Info mode and were then questioned about their preference between normal mode and Info mode.

4.4 Volunteers

The group of volunteers was composed of 20 members, aged between 22 and 33 years. All volunteers are common users of smartphones to minimize constraints derived from the difficulty of using mobile devices. The developed application was never tested before by any of the volunteers. The test process was explained so that all the elements could know exactly what steps to perform.

5. RESULTS

The modifications applied during model retraining did not result in any loss of performance, despite the addition of an extra label for detection, which consisted of precision, recall, F1-score, and accuracy values of 0.9589, 0.7269, 0.8269, and 0.7082, respectively, for an IoU of 50%. With respect to the validation of the scan system, the scanning times are summarized in a box plot. Out of a total of 80 scans, with 20 scans per device, there were three instances where the pulse was not recorded by the blood pressure meters, although the remaining analyses were successfully scanned. Conversely, for the oximeter, three pulse measurements were incorrectly recorded because the AI method confused two digits (1for 4, and 2 for 3). In the user preference experiment regarding automatic Normal Mode data entry, 17 out of 20 volunteers preferred automatic scanning. When comparing preferences between the normal mode and Info mode of the automatic scanning, the volunteers were evenly split: 9 preferred the normal mode, 9 prefer the info mode, and 2 had no preference (see Figure 2).

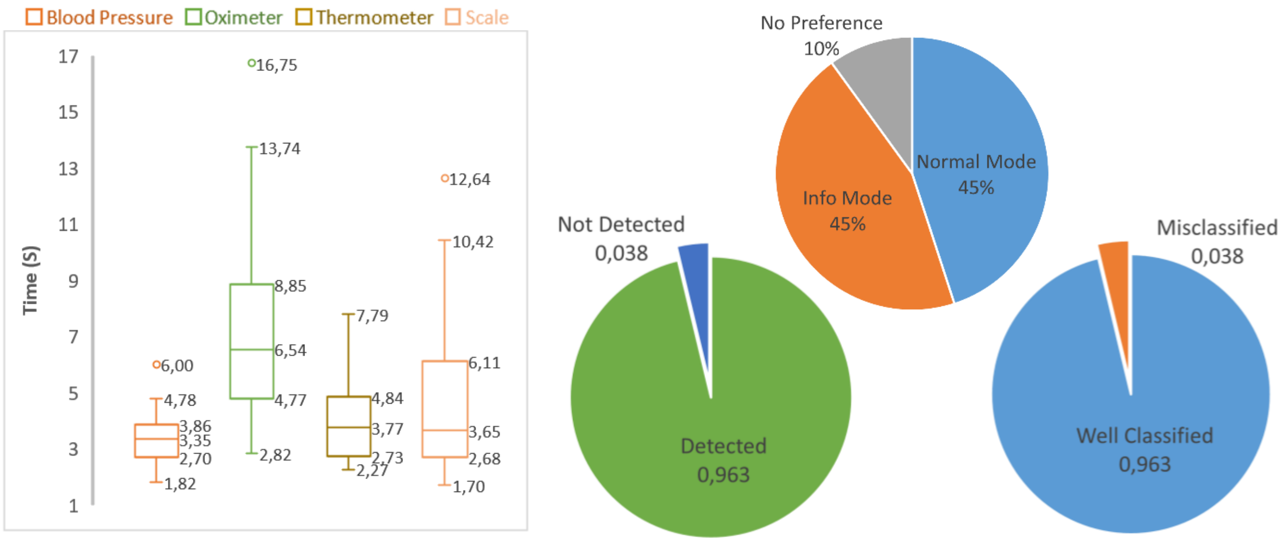

Figure 2. System validation results: box plotter of scan time, user preference (normal/info mode), detection and classification rate.

6. DISCUSSION

Starting by analyzing the scanning times (see Figure 2), it is possible to verify by that the time of digitization of the blood pressure meter is shorter when compared with the remaining devices. This can be explained by the fact that the screen size of the blood pressure meter is significantly larger than the others, which helps to frame the camera with the screen faster. Concerning the accuracy of the detections, it was verified that the analyses were successful 92.5% of the time since 74 of the 80 analyses (20 volunteers * 4 devices) were correctly performed. Concerning the remaining 6 unsuccessful detections, two different situations were detected. When no pulse was detected in a blood pressure monitor, it was due to a threshold that is applied to the detected digits based on their bounding box areas. Thus, the pulse bounding box is detected for these cases but with a small area. Adjusting the threshold to the minimum acceptable area can avoid this type of problem. For the case of the wrong classification of digits in the oximeters, this can be explained the fact that the digits are not seven segments as expected from most of the digits from our dataset. To overcome this problem more examples of this type of digits should be added to the training dataset.

In the experiment in which the volunteers were subjected to the insertion of the analyzes manually or with the smart scan without assistance, the 3 elements who preferred the manual method were motivated by different factors. One did not find it practical to frame the information displayed on the screen of the medical devices while carrying out the measurement. Although the vision system had no problem with reflections caused by medical device screens, one of the three volunteers preferred manual insertion because he was afraid that the vision system would fail. The remaining volunteer that preferred manual entry justified the preference due to the timing of a scan that took some time.

Between Smart Scan mode with or without assistance there is no dominant preference for one of the modes. It would be advisable to make both modes available in mHealth applications that will integrate the smart scanning system, so that the user himself can define the mode he wants to use.

It is necessary to mention that some of the volunteers suggested the integration of digital zoom in the smart scan modes so that they did not have to crouch to better digitize the information contained in the scale. This feature would also be relevant for future users with mobility issues.

7. CONCLUSION

In this work, a Smart Scan of Medical Device Displays was adapted for a mobile environment and provided as an Android library. It was also developed a prototype application that imports the library developed to validate the proposed system and to make studies regarding usability preferences. It is now expected that, after rectifying the detected gaps, this solution will be implemented in mHealth applications.

Although the vision system developed in this work fits most medical devices, with the implementation and use on a larger scale, devices with different analysis display layouts may arise for which our system was not developed. As future work, it is necessary to keep a constant updating of the system for the different medical devices that may appear on the market.

ACKNOWLEDGEMENTS

This work was funded by the Innovation Pact HfPT – Health from Portugal, co-funded from the "Mobilizing Agendas for Business Innovation" of the "Next Generation EU" program of Component 5 of the Recovery and Resilience Plan (RRP), concerning "Capitalization and Business Innovation", under the Regulation of the Incentive System "Agendas for Business Innovation". The work was also funded by Foundation for Science and Technology (FCT) - projects UIDB/05549/2020 (DOI: 10.54499/UIDB/05549/2020), UIDP/05549/2020 (DOI: 10.54499/UIDP/05549/2020), LASI/LA/P/0104/2020, and CEECINST/00039/2021 for their support of this research.

REFERENCES

[1] I. Ahmad *et al.*, «Emerging Technologies for Next Generation Remote Health Care and Assisted Living», *IEEE Access*, vol. 10, pp. 56094–56132, 2022, doi: 10.1109/ACCESS.2022.3177278.

[2] Y. W. Chong, W. Ismail, K. Ko, e C. Y. Lee, «Energy Harvesting for Wearable Devices: A Review», *IEEE Sens. J.*, vol. 19, n.º 20, pp. 9047–9062, out. 2019, doi: 10.1109/JSEN.2019.2925638.

[3] A. E. Barrett e C. Gumber, «Feeling Old, Body and Soul: The Effect of Aging Body Reminders on Age Identity», *J. Gerontol. Ser. B*, vol. 75, n.º 3, pp. 625–629, fev. 2020, doi: 10.1093/GERONB/GBY085.

[4] L. Ferrucci *et al.*, «Measuring biological aging in humans: A quest», *Aging Cell*, vol. 19, n.º 2, fev. 2020, doi: 10.1111/ACEL.13080.

[5] P. Lobo, P. Morais, P. Murray, e J. L. Vilaça, «Trends and Innovations in Wearable Technology for Motor Rehabilitation, Prediction, and Monitoring: A Comprehensive Review», *Sensors*, vol. 24, n.º 24, Art. n.º 24, jan. 2024, doi: 10.3390/s24247973.

[6] S. Kanoga, T. Hoshino, e H. Asoh, «Semi-supervised style transfer mapping-based framework for sEMG-based pattern recognition with 1- or 2-DoF forearm motions», *Biomed. Signal Process. Control*, vol. 68, jul. 2021, doi: 10.1016/J.BSPC.2021.102817.

[7] P. Lobo, J. Vilaça, H. Torres, B. Oliveira, e A. Simões, «Smart Scan of Blood Test Documents to be Integrated in a mHealth Application», em *2022 10th E-Health and Bioengineering Conference, EHB 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/EHB55594.2022.9991279.

[8] P. Lobo, J. L. Vilaça, H. Torres, B. Oliveira, e A. Simões, «Smart scan of medical device displays to integrate with a mHealth application», *Heliyon*, vol. 9, n.º 6, jun. 2023, doi: 10.1016/j.heliyon.2023.e16297.

[9] E. Finnegan, M. Villarroel, C. Velardo, e L. Tarassenko, «Automated method for detecting and reading seven-segment digits from images of blood glucose metres and blood pressure monitors», *J. Med. Eng. Technol.*, vol. 43, n.º 6, pp. 341–355, ago. 2019, doi: 10.1080/03091902.2019.1673844.

[10] D. Tsiktsiris, K. Kechagias, M. Dasygenis, e P. Angelidis, *Accelerated Seven Segment Optical Character Recognition Algorithm*. IEEE, 2019.

[11] W. Liu *et al.*, «SSD: Single Shot MultiBox Detector», vol. 9905, 2016, pp. 21–37. doi: 10.1007/978-3-319-46448-0_2.

[12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, e L.-C. Chen, «MobileNetV2: Inverted Residuals and Linear Bottlenecks», 21 de março de 2019, *arXiv*: arXiv:1801.04381. doi: 10.48550/arXiv.1801.04381.