Delft University of Technology

Faculty of Mechanical, Maritime and Materials Engineering
Department of Biomedical Engineering
Specialization: Medical Devices and Bioelectronics

Low-Cost Automated Paper Strip Reader (APSR)

Author:
Tejal Shivaji Shinde

Supervisors:
Prof. Dr. Ninad Mehendale
Prof. Dr. Paddy French

November 26, 2021

Contents

List of figures

List of Tables

List of acronyms

Abbreviations

PAD	Paper-based Analytical Device
ASSURED	Affordable, Sensitive, Specific, User-friendly, Rapid and Robust, Equipment-free, Deliverable
POC	Point-Of-Care
ELISA	Enzyme-linked ImmunoSorbent Assay
RPi	Raspberry Pi
HMI	Human Machine Interface
GUI	Graphical User Interface
OS	Operating System
App	Application
LAMP	Loop-mediated isothermal Amplification
R	Red
G	Green
B	Blue
IoT	Internet of Things
LMIC	Low and Middle-Income Countries
WHO	World Health Organization

Abstract

Paper strip analyzer or a medical strip reader is a low-cost Paper-based Analytical Device (PAD) that is easy-to-use and widely accepted Point-Of-Care (POC) alternative to the relatively complex and expensive ELISA (Enzyme-linked ImmunoSorbent Assay) tests. These devices are used for determining the chemical concentrations (pH, glucose, protein, nitrite, etc.) of the body fluids (blood, urine, sweat, saliva, etc.) to monitor organ functionality and ensure early diagnosis of potential diseases. The test paper strips can be analyzed by comparing the colour changes on the test strip to the existing colour chart either by the operator's visual interpretation or by a smartphone-based automated system which is a fairly new technology from the past decade. Apart from being time consuming, the manual (naked eye) interpretation of results is subject to the robustness of the clinicians' eyesight and ambient lighting conditions that may potentially introduce an error while obtaining results. This presented the need to automate this process of colour detection. Despite the faster computation speed of smartphone-based systems, they may introduce image acquisition related errors owing to the varying camera resolutions in different smartphone models. Smartphones also face the issue of poor battery backup and software (in)compatibility. There exists a research gap in terms of an automated technology that is ASSURED: A = Affordable, S = Sensitive, S = Specific, U = User-friendly, R = Rapid & Robust, E = Equipment-free, D = Deliverable. These are the requirements proposed by the World Health Organization (WHO) for commercial medical sensors. In this study, we try to overcome the shortcomings of smartphone-based technology while adhering to the ASSURED criteria. We present a fully functioning prototype of a 3D printed Raspberry-Pi microcontroller based paper strip reader with a rechargeable battery and consistent illumination. The device is paired with a backend Python program and a user-friendly GUI to display the results. The device also has a feature to export the results with one click to your clinician via email. APSR algorithm is 88.7% accurate and the speed of operation is ≈5s which is faster than the best speed reported in the literature yet. The future scope involves making use of Machine Learning and Deep Learning to further improve the algorithm.
Keywords: Point-of-care, medical test strip, sensors, Paper-based Analytical Device, colour detection

Chapter 1

Introduction

When you visit a hospital/clinic, apart from checking your vitals like blood pressure and heart rate, another preliminary step is to get your body fluids tested like blood or urine depending upon your symptoms. Now, because these tests are a primary and mandatory step towards diagnosis, we can imagine how critical the output accuracy would be here. Also the time pressure for getting the results out in time is something to consider. A paper strip reader is a medical test kit used to analyze the chemical concentrations (pH, glucose, protein, nitrite, etc.) and its contents in the body fluids (blood, urine, sweat, saliva, etc.) [1, 2, 3, 4, 5]. There also exist water test strips to determine various water contents like total hardness, chlorine, bromine, nitrite, etc.

1.1 Need

Due to the COVID-19 pandemic, the hospitals and clinics have a lot of load and are not able to accommodate and treat all the patients at the same time. This reduces the overall efficiency of the medical staff that can affect the health of people [6, 7]. Looking at the current scenario, people are advised not to step out of their houses until and unless it is essential. This has affected the diseased population who did not get an opportunity for early detection of the ailments. Even during normal circumstances (non-pandemic situation), general population can be unwilling to commute to the hospital and spend money on routine check up unless they experience a medical emergency. Since basic medical testing is an initial and essential part of any medical treatment, it is important to get accurate diagnosis. Many underdeveloped villages in Low and Middle-Income Countries (LMIC) lack basic medical facilities making it difficult for the residing people to avail them [8, 9, 10]. Early diagnosis of diseases is especially important for the population of LMIC namely India where 988 million people (75% of the total population) do not have a health insurance [11].

1.2 Objective

In addition to being ambient light dependent, the paper test strips also tend to fade away over time which poses a high scope for error when it comes to manual interpretation of results. In order to avoid human-induced error, the system should involve minimal human interference, and the diagnosis should be automated which will solve the issue of variation in color perception and will yield accurate, reproducible and fast results. However, better accuracy should not mean a complicated system, as the potential market for such devices include households as well. These

analytical devices must be designed in a user-friendly fashion presuming the lack of medical or technical expertise on the user end. The device must also be modular by design as to ensure easy replacement of the electronic components instead of replacing the entire device. Also, modularity will ensure easy cleaning of the device. Overall, this medical device should satisfy the ASSURED criteria levied by the World Health Organization (WHO) for medical sensors.

1.3 Approach

The use of paper test strips already satisfies the affordability criteria as they are readily available in the market at cheaper costs. There are various approaches that have been taken to address the vital aspects to make medical technology more POC and household compatible like 3D printing [12]. It drastically shortens the time frame between ideation to actuation of a concept, and delivers lightweight and portable yet sturdy products. This approach is getting a lot of attention in the scientific world as the researchers realize the importance of affordable and accessible medical technology not only for the medical field in general but also for the rural areas in undeveloped or developing countries that lack basic amenities like clinics or POC facilities. This is especially important because the people residing in such areas lead a below average standard of life and are at higher risks of attracting diseases which need timely diagnosis [13]. Therefore, it is critical to make the technology available to this population that is affordable and can help them monitor their health without the immediate need for medical professionals.

1.4 Thesis report organization

In the next Chapter 2, an elaborate literature review is presented on the different colour analysis technologies and a comparison is made based on the ASSURED criteria. Chapter 3 talks about the methods and materials of the involved hardware, software, dataset and the proposed system protocol. Chapter 4 presents all the results obtained for all the 15 water test parameters. Chapter 5 elaborates on the comparison of the proposed system with the existing technology and literature. Finally, the derived conclusions and possible advancements are elucidated on in Chapter 6. And finally, the Python codes used for the algorithm are listed in Chapter 7 i.e. Appendix.

Chapter 2

Literature review

In this chapter, the literature on colorimetric methodologies and its system level advancements have been reviewed.

2.1 History

Point-of-care colorimetric analysis is performed by existing methods by analyzing the assays or paper strips. Low-cost urine, pH and nitrite testers for monitoring human health conditions have been published in the past decade [14, 15]. The study demonstrates capturing of images of the urine reagent strip absorbing the collected urine using the charge-coupled device (CCD) camera module. In the system proposed by Luka *et al.*, once the samples and the reagents are mixed together, images are captured using a Raspberry Pi camera module hue–saturation–value (HSV) coordinates of the image are identified with the help of computer vision [15]. Few of the existing systems have ability to wirelessly transmit the results obtained to hospitals or clinics.

Many image processing techniques have been implemented with the help of Raspberry Pi [16, 17]. The vital parameters of urine such as specific gravity, pH level, blood, protein, and urobilinogen are measured by Paglinawan *et al.* by applying image processing with the help of a Raspberry Pi controller and camera [16]. Colorimetric analysis can also be done using the Raspberry Pi for determining the unknown agents in presumptive drug tests [18]. Similarly, a low-cost colorimetric analyzer has been developed by Boehle *et al.* for calculating the Michelis-Menten enzyme kinetic constants [19]. A 3D printed box has been designed to hold the Raspberry Pi controller and

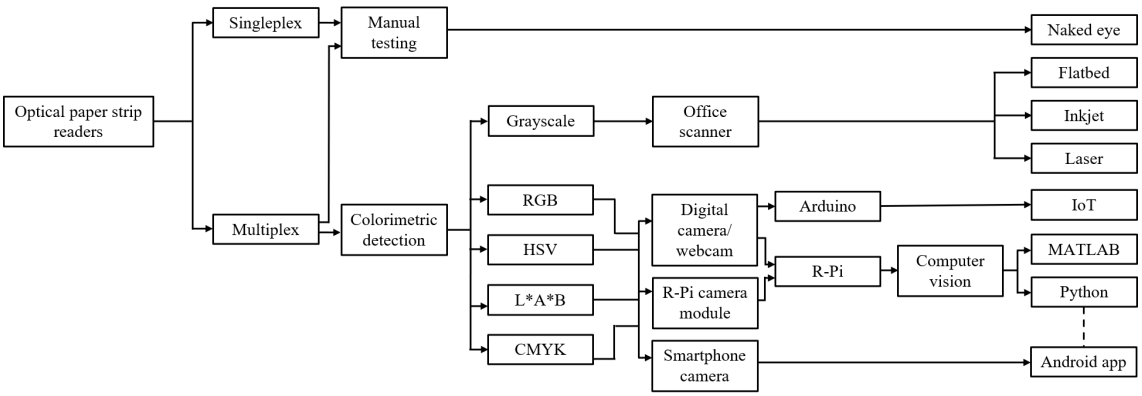


Figure 2.1: Paper strip reader technology and hierarchy.

camera in order to eliminate the ambient lighting and thereby give accurate results. Papadakis *et al.* have developed a low-cost, portable, and rapid device for real-time quantitative Loop-mediated isothermal Amplification (LAMP) of various diseases and disorders [20]. The samples are uniformly heated during analysis while the Raspberry Pi camera is used to take pictures of the samples while they are heating. The RGB values of the samples are then extracted from the images. Although the use of Raspberry Pi controller enables the existing systems to process the data and give fast results, these systems do not satisfy the ASSURED criteria for medical diagnostic devices.

The captured images are analyzed to display the results on LCD. A user-friendly paperplastic hybrid microfluidic lab-on-a-chip (LOC) device has been developed by Jalal *et al.* [21]. Once the urine analyte is inserted inside the chamber consisting of the reagent (as shown in Fig. 2.3 (b), a smartphone is used to capture its image and perform the image processing i.e. urine analysis to give the desired results. A state-of-the-art saliva test reader as reported by Carrio *et al.* has utilized a light box with an in-built illumination system to avoid the interference from the ambient light and captured images using smartphone [22]. Jang *et al.* have developed a portable and standalone smartphone-based system to detect progesterone in milk using an ELISA-based technique [23]. The pixel values of the images of samples are analyzed and the measured progesterone is displayed on the smartphone with the help of an android application. The system proposed by Papadakis *et al.* can be operated by using a smartphone and enables rapid prototyping as the device can be made using a 3D printer [20]. The device uses a Raspberry Pi zero as its computing unit and it uses a Raspberry Pi camera module to take periodic images of the samples.

Heidari-Bafroui *et al.* [24] have developed an infrared lightbox that is used to enhance the detection limits of the devices used to detect phosphate levels such as Quantofix devices [4] as well as Indigo units. Sweeney *et al.* have used an edge detection algorithm to monitor the flow rate of the blood through the developed paper microfluidic channel [25]. This technique can be used for monitoring blood coagulation with respect to the variation in the concentration of heparin and protamine which can be an important factor during surgeries. Soni *et al.* reported an optical biosensor for analyzing glucose levels present in the saliva using paper strips [26]. The study used RGB profiling on an open source image processing software called GIMP for analyzing the images captured on an office scanner. Wang *et al.* explained an image quantification model using SPSS [27]. Here they converted the colour to hue and delivered predictions for the diagnosed pH values using a reverse model. Lactoferrin has been detected by Kudo *et al.* using microfluidic paper-based analytical devices [28]. Deep Learning models such as AlexNet, GoogLeNet, Inception and ResNet were used by Tania *et al.* to classify the types of assays as ELISA or LFA [29]. Diseases like colorectal cancer, gastrointestinal diseases, intestinal inflammation, etc. can be detected with the help of a fecal occult blood test. Lin *et al.* have developed a rapid, sensitive, and linear device (as shown in Fig. 2.3 (a) for fecal occult blood (FOB) detection [30]. An FOB test is used to detect bleeding in patients with gastrointestinal diseases. The rapid test system is used to detect and calculate the concentration of the sample, so detection of the immune coloration response is more accurate in quantitative analysis.

Table 2.1 shows the comparison between existing methods that use colorimetric analysis to detect various parameters such as pH, nitrite, phosphorous, glucose, etc. The studies have been performed on different analytes for detection of different components with the help of paper strips and assays. Most of the techniques use Raspberry Pi as their controllers to perform the analysis. As mentioned earlier, the World Health Organization (WHO) stated that sensors should be "ASSURED", that is, affordable, sensitive, specific (i.e. reduction in human mediation for determining

Figure 2.2: 15 in 1 water test strip kit by Test lab [31].

the results), user-friendly, rapid & robust, equipment-free (standalone), and deliverable (portable) as a specification for diagnostic devices used in developing countries. The existing methods can detect various analytes accurately. However, the ASSURED criteria is not satisfied by all the existing methods.

As shown in fig.2.1, the medical strip reader literature is divided into different types of controllers, camera and illumination units, image processing algorithms and colour spaces and battery backups. The test strips are primarily divided into 2 types: singleplex and multiplex. Singleplex optical systems only give information about one parameter and the results are binary (yes/no), whereas, multiplex systems are semi-quantitative as it gives us the content values based on the threshold of the colour changes.

2.1.1 Naked eye detection

The multiplex systems come with vertical paper test strips and a colour chart as shown in fig.2.3(b). After introducing the paper strip to the analyte, it is allowed to dry for a few seconds (as described on the test strip box by the manufacturer) and finally compared with the colour shade chart where each colour shade corresponds to a particular quantity as shown in fig.2.2. When this comparison is done visually by an operator/clinician/user, it is referred to as naked eye testing in the present report.

2.1.2 Office scanner

The visual interpretation of colour comparison can be automated by introducing mediators for image acquisition and image processing. Office scanners were one of the first used image acquisition units after manual testing fig.2.3(a).

Flatbed

A basic flatbed scanner in combination with an open source image processing software (GIMP) can be used for detection of glucose in an analyte as shown in fig. 2.3(a) [26]. The Limit of Detection (LOD) was limited to 22.2 mg/dL glucose i.e. the biosensor was unresponsive when the glucose content in the analyte was lower [26]. Also, the lactic and ascorbic acid were common interferents in the giving accurate results thus compelling the patients to take the test on an empty stomach. The use of desktop and flatbed scanners has been quite common in the past decade [32].

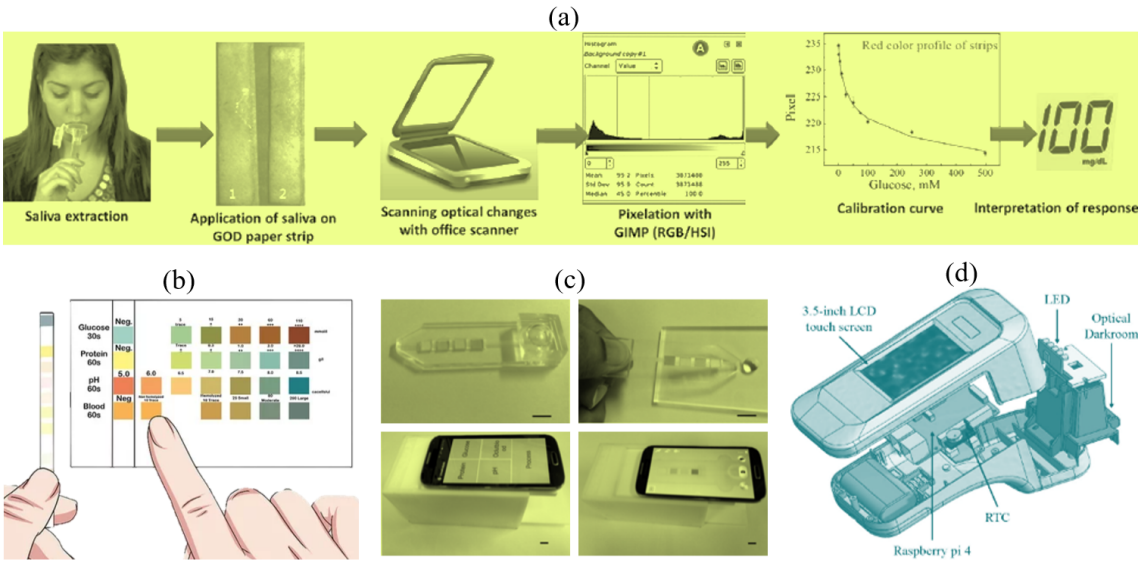


Figure 2.3: Studies from the literature. (a) Office scanner based saliva analysis [26] (b) Naked eye detection with colour matching chart. (c) Paper plastic hybrid microfluidic lab-on-a-chip (LOC) device and a smartphone to capture images for urine analysis [21]. (d) Low-cost portable device developed for the detection of fecal occult blood (FOB) using raspberry pi [30].

2.1.3 IoT

An arduino-based pH sensor was developed by Gosavi *et al.*[33]. Here, a colour sensor was employed and the output was sent to the microcontroller for processing. The final output was sent to the display screen and later to the PC via wifi module.

2.1.4 Computer vision

Computer vision has recently risen as a revolutionary technology that can be easily integrated with the microcontrollers to actuate the conceptual projects. There have been studies about determination of different body fluid concentrations like saliva [22], pH and nitrite [15] and also nitroaromatics [31] using smartphone or raspberry-pi. Keeping the controller and the image acquisition unit separate gives better results [15] as compared to an all-in-one smartphone [31].

Python

A study was done to analyze the blood coagulation in response to heparin and protamine by blood flow edge detection using Python programming [25].

2.1.5 Android app

There have been various studies proposing android applications for paper strip and assay analysis. A raspberry pi enabled infrared lightbox was reported by Heidari *et al.* that was compatible with the mobile for receiving output via the WiFi module and android/iOS app [24]. A smartphone app based microfluidic LOC was developed for urinalysis using colorimetry [21]. Similarly, another study reported the use of colorimetry but with Raspeberry pi controller and a 3D printed device insstead of LOC [20].

Table 2.1: Literature comparison chart.
ASSURED: A = Affordable, S = Sensitive, S = Specific, U = User-friendly, R = Rapid & Robust, E = Equipment-free, D = Deliverable

Analyte	**Analyte type**	**Controller**	**Image acquisition**	**Processing algorithm**	**ASSURED**	**References**
pH, Nitrite	assays	Raspberry Pi 3B	Raspberry Pi camera	OpenCV Python	A, E	[15] (UK)
Urine	urine reagent strip	Raspberry Pi zero	Raspberry Pi zero camera	-	A	[14] (Switzerland)
Urine	urine test strip	Raspberry Pi	Raspberry Pi camera	-	A, U, E	[16] (Japan)
Phosphate	phosphate test strips	Raspberry Pi 3	No infrared filter camera	MATLAB	E	[24] (USA)
Drug testing	saliva	Raspberry Pi B+	-	-	E	[18] (USA)
Urine	urine reagent strip	Smartphone	Smartphone camera	Android app	A, S, S, R, E	[21] (Korea)
Blood	paper microfluidic assay (Heparin and Protamine Solutions)	Raspberry Pi 3 B	Raspberry Pi camera	Python	-	[25] (Switzerland)
Saliva	Paper strip	Smartphone	Smartphone	Computer vision & machine learning	-	[22] (Japan)
Nitroaromatics	chemical nonselective sensors	Raspberry Pi	Microcamera	Computer vision	S, S	[17] (Spain)
Lactoferrin	Fe3+–indicator	Smartphone	Smartphone	Image processing	A, U, E	[28] (USA)
Fecal occult blood	immunochromatographic strip.	Raspberry Pi	USB camera	Image processing	A, S, S, R	[30] (Switzerland)
Saliva	-	Raspberry Pi Zero	Raspberry Pi Zero camera	Image processing and smartphone app	R	[20] (Greece)
Water pH	paper strip	-	SLR camera	SPSS	S, S	[27] (China)
-	pH paper strip	Arduino	Colour sensor	IoT - Arduino	A, E	[33] (India)
Saliva glucose	paper strip	-	Office scanner	GIMP - RGB profiling	R	[26] (India)

(a) (b)

Figure 2.4: Similar existing technology. (a) Urine Strip reader BW-300 produced by Bioway Biological Technology Co.,Ltd [5]. (b) Urine strip analyzer 'Quantofix Relay' manufactured by Macherey-Nagel [4].

2.1.6 Existing technology

Macherey-Nagel has a commercially available digital paper strip analyzer (as shown in Fig. 2.4(b) [4]. The company manufactures special pH-Fix and QUANTOFIX strips which can be used to test the quality of water samples. The paper strip is dipped into a solution and then placed in the machine. The machine analyzes the inserted sample. This product can only analyze those special strips. Dialab manufactures a commercial urine strip analyzer [34]. It can store about 2000 test results, it is portable, rapid and it can be calibrated. However, it is designed to analyze only the urine samples. Microsidd produces a commercial urine analyzer [35]. It can store the results of the previously performed 2000 tests, it can support 120 tests per hour i.e. 1 test in 2 minutes in continuous mode. It only takes one minute for strip incubation. Similar to the last product, this one is limited to urine analysis only.

As shown in Fig. 2.4 (a), Bioway Biological Technology produces an automated paper strip analyzer [5], that can analyze various parameters like glucose, nitrite, bilirubin, ketone, pH, etc. It has a large memory, a 7-inch touchscreen, uses computer vision and artificial intelligence for analyzing paper strips. Out of the above-mentioned four products, some products can only analyze certain types of paper strips like urine paper strips or specially made strips like pH-Fix for pH measurement.

2.2 Discussions

The traditional naked eye detection technique possesses a lot of potential for error. To decrease the manually induced error that leads to biased results, it is of paramount importance to automate the analysis of medical paper strips. After studying the existing and commercialized technology, it was observed that, the devices were too expensive costing more than INR 30,000 which is already not abiding by the 'A' for affordability from the ASSURED criteria let out by the WHO. The products in the market have limited functionalities like only water or only urine analysis and are not compatible with the universal test paper strips thus affecting the 'D' for deliverability from the ASSURED criteria. These devices are also quite bulky for POC or household usage. In order to develop a device for colorimetric analysis with high accuracy, it is primarily important to control the lighting conditions during image capture.

Table 2.2: Technology comparison table based on analysis time.

Method	Technique	Speed
Soni *et al.* [26]	Office scanner; RGB profiling	45 s
Heidari *et al.* [24]	Raspberry Pi; RGB profiling	60 s
Sweeney *et al.* [25]	Raspberry Pi;Python image processing	3-5 min.
Luka *et al.* [15]	Raspberry Pi; Python-HSV colour space	5 min.
Papadakis *et al.* [20]	Raspberry Pi; RGB profiling	30 min.

The literature comparison in elucidated in Table 2.1 and Table 2.2. Of all the mentioned literature, the system reported by Soni *et al.* is the fastest with 45 s of operation time for 1 test [26]. A lot of literature has reported the use of smartphone for addressing the problems arising from existing bulky devices. However, a smartphone has limitations as well. It is non-modular and the battery back up is not available for longer hours. Every camera comes with different specifications like resolution and focal length. Thus, the clarity and resolution of the image will change according to the smartphone camera. This can affect the accuracy of the algorithms that perform image processing on the images obtained using the smart phone.

It is important for the future technology to be standalone and portable to widen the LMIC market and scope of the device. Also, the User Interface (UI) of the device should be such that the need for operating expertise is eradicated. This will ensure a household-friendly device which will greatly benefit every individual, but specifically, the LMIC population. The potential ASSURED should allow the user to keep their health in check on a daily basis thereby avoiding major risks in the future. This will also help the doctors in hospitals and clinics as the rapid results given by the automated paper strip reader will enable them to speed up the treatment and further treat more patients in a shorter period of time. The low-cost portable device will allow people from low and middle-income countries to observe the health of their organs at home at an affordable rate and avoid the need to visit test laboratories. The use of computer vision for image processing on the colored paper strips will let the device predict health problems such as diabetes, urine infection, etc. accurately without the need for medical experts. The ideal device should perform more than 1 test at a time, for example ammonia, nitrate, nitrite, etc. in case of water test strips in order to increase the operational efficiency in clinics and laboratories.

3D printing technology has revolutionized the manufacturing industry as a whole as materials PLA and ABS that are used for 3D printing are reproducible as well as sturdy and robust. The time frame between ideation and actuation of the product is very small, thanks to the Computer Aided Design (CAD) supported 3D printing.

2.3 Research gap

The smartphone based medical strip readers despite being faster systems, lack modularity because the smartphone is an all-in-one device i.e. camera, controller, display screen, illumination, and battery backup. This means that even if any one of the mentioned features is damaged, the entire smartphone needs to be replaced which is rather working against the ASSURED criteria in terms of affordability and robustness. Additionally, the battery backup of the smartphone is not compatible for full day usage and will need recharging several times during the day thus slowing down the workflow of the users which will hamper the efficiency. Not every Android or IOS phone has the same camera resolution. This will impact the image processing software that is installed

for medical diagnosis. It will hamper the accuracy of the results. The existing products as well as published researches do not meet the ASSURED criteria as a whole. The ones that do are not paper strip compatible and only support assays as the analyte [20, 25, 15].

ASSURED technology is of paramount importance to empower clinics and hospitals to treat a large number of patients, especially at the time of a global pandemic like COVID-19, also to enable patients to monitor their medical condition from the comfort of their homes.

2.4 Summary

A review on the medical paper strip analyzer technology is performed in the present report. The fastest system reported in literature performs one test analysis in 45 s and it is proved on saliva paper strips for evaluating the glucose concentration in the saliva [26]. However, since the study reported the use of office and RGB profiling for image capturing and processing, there lies some scope for improvement. Especially when it comes to the image acquisition unit and the controller as different choices may ensure portability and thus abide by ASSURED criteria. The handy medical strip analyzers will not only help the hospitals and clinics boost their efficiency but will also revolutionize the health front for the less financially fortunate and rural population in the underdeveloped and developing countries.

Chapter 3

Methodology

This chapter includes detailed explanation of the methods and materials used for designing and prototyping the proposed system.

3.1 Materials

The following section elaborates on the materials used.

3.1.1 Battery

The different types of batteries to consider from were Ni-Cd, Ni-MH, Lead acid, and Li-ion. These batteries were compared based on the energy density, self-discharge rate, longevity, cost, environmental impact, size and weight. Ni-Cd and lead acid batteries have comparable characteristics in the form of low energy density, high self discharge rate, short lifetime and economical. They are both equally toxic to the environment. However, Ni-Cd batteries are slim and lightweight by structure, whereas, lead acid batteries are bulky. The energy density of Ni-MH batteries ranges from moderate to high. Its other characteristics include high self-discharge rate, short lifetime, expensive in cost and the structure ranges from slim to bulky. Ni-MH batteries also have proven to be toxic for the environment. The final type of battery under consideration was li-ion. The characteristics include very high energy density, low self discharging rate, long lifetime, economical, slim and light in weight, and it has relatively less environmental impact. After considering the above comparison. We chose li-ion battery for our device. Since the device's voltage requirement was 11-12V, we chose a 3 battery shield with each battery of 3.7V.

3.1.2 Controller and processor

After carefully considering the available processors, we chose Raspberry Pi 4B owing to its following specifications:

- Peripheral support – 26 GPIO pins
- Coding language freedom – C, C++, Java, Python
- Processor - 1.6GHz
- RAM – 8GB

- Relatively inexpensive
- Easy to use
- Popular for our application as per the literature
- Large user community
- Multiple OS support - Windows on ARM edition, Android, Linux

Other boards under consideration were PCDuino and BeagleBone. However, they were both too expensive thus nullyfying the 'A' (Affordable) criteria from ASSURED. Arduino had no OS compatibility. Orange Pi and Banana Pi had smaller user community and less OS' were compatible.

3.1.3 Operating System (OS)

After finalizing the board, next step was to select an OS. Windows 10 for RPi was just community level and not official so that was out of option. Android had its own performance limitations as discussed in section 2.3. Other Linux based OS' include Ubuntu MATE which was only available for server desktop use, Gentoo which was relatively complex and had no default GUI, and Raspbian OS which is RPi's original OS. Its characteristics include powerful processor, python support, lightweight, easy to install, pre-installed softwares like Minecraft Pi, Java, Mathematica, and Chromium, huge user community with numerous studies on OpenCV machine learning. Therefore, we chose Raspbian OS for APSR.

3.1.4 Display unit

The requirement from the display unit was that it should have an interactive yet compact display. Despite having an HD display, the stand 16*2 LCD lacked Human Machine Interface (HMI). OLED and capacitive touchscreen displays were expensive. SO we finalized a 5" resistive touchscreen as it included HMI owing to the touchscreen feature, compact and affordable thus satisfying the Affordable criteria from ASSURED.

3.1.5 Image acquisition unit

There were two options for the camera unit - RPi camera module and a standard webcam. We chose a standard webcam owing to its affordability and better availability thus satisfying A (Affordable) and D (Deliverable) criteria from ASSURED. On the other hand, RPi camera module was expensive and deliverability might have been an issue.

3.1.6 Illumination unit

In order to illumination the test paper strip, a number of light sources were considered and compared. Ambient light has high variability and thus, low accuracy; luminescent lamp has low colour rendering index and heats a lot; single LED has less brightness to illuminate an entire strip with 15 colour blocks, whereas, an LED strip is affordable, allows customizing light intensity, long-lasting and energy efficient. Hence, we chose a white LED strip.

3.1.7 Analyte

There were test paper strips available for various fluids including blood, urine and water. Due to lack of an appropriate facility, blood and urine strips were ruled out and water test strips were finalized. The water strip covered the following 15 test parameter:

- Water
- Total hardness
- Bromine
- Residual chlorine
- Total chlorine
- Iron
- Copper
- Lead
- Nitrate
- Nitrite
- Ammonium chloride
- Fluoride
- Carbonate
- Total alkalinity
- pH
- Cyanuric acid

3.2 Design concept

Based on the materials discussed in the previous section, the following concept diagrams were designed.

Figure 3.1 illustrates the block diagram of the reported device, APSR. The first block is the database which is nothing but the memory card loaded with the Raspbian OS and is inserted into the microcontroller and processor Raspberry Pi 4B. The controller board is further connected to the the 5 inch resistive TFT touchscreen via ribbon connector. Further, the webcam is connected to the controller via USB. The battery connection is given to both the controller and the LED strip for the power supply.

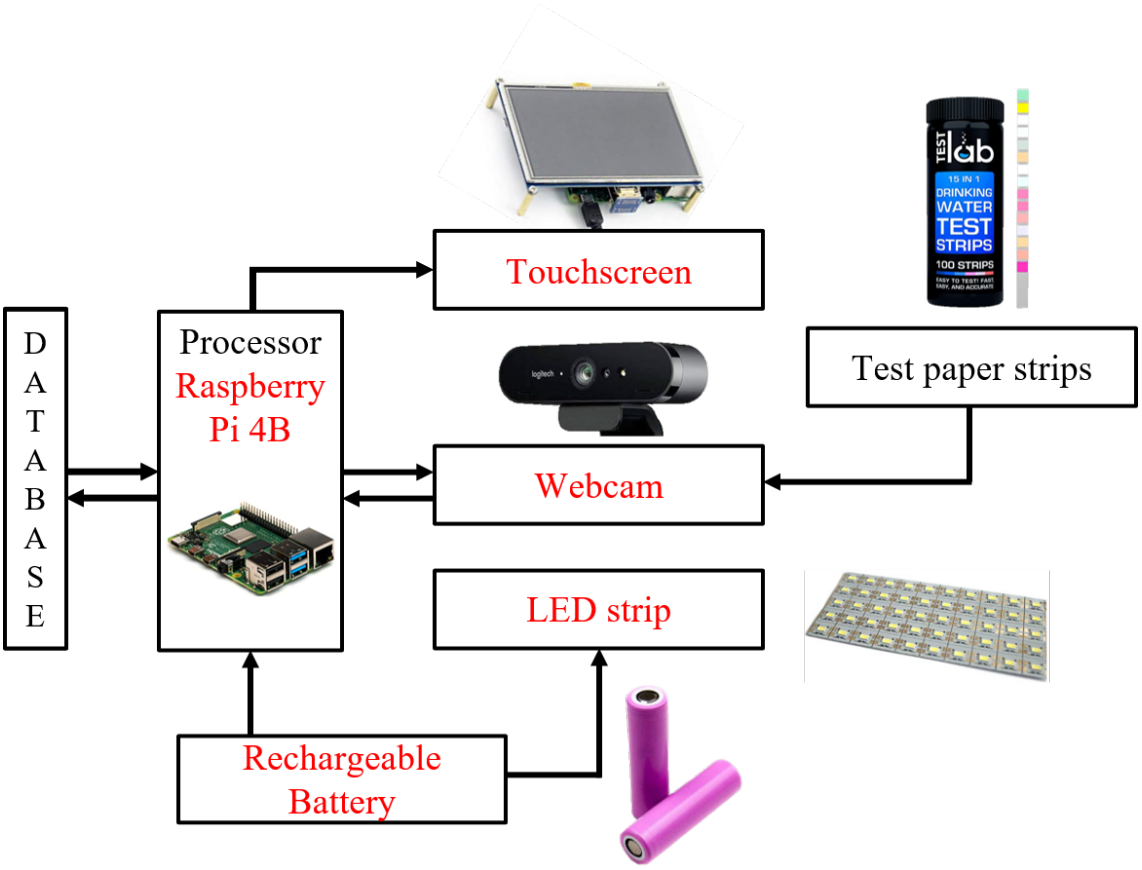
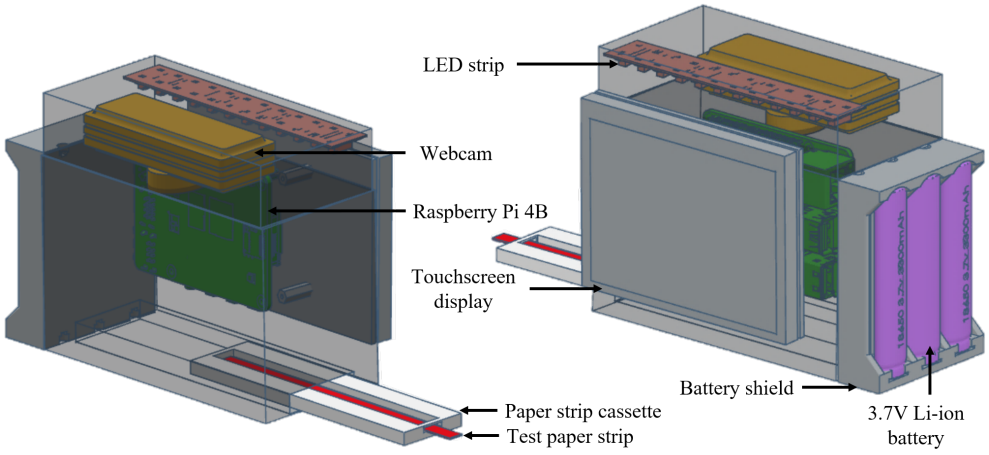


Figure 3.1: Block diagram of APSR.

Figure 3.2 illustrates the front and the back view of the system and fig. 3.3 shows an exploded view of the concept diagram inside. As discussed in Chapter 2, ambient light interference ought to introduce bias in the readings. Therefore, an enclosed system was designed that resembles a box. The box is divided into 2 parts: top and bottom.



(a) Front view (b) Back view

Figure 3.2: Concept diagram of APSR.

The components inside the enclosed box include a webcam to take a picture of the paper strip, a white LED strip to illuminate the inside of the box, the RPi processor board, and a paper strip

holding cassette to hold the strip in place. Of all these components, the webcam and the LED strip lies at the top end of the box while paper strip and the board lies at the bottom.

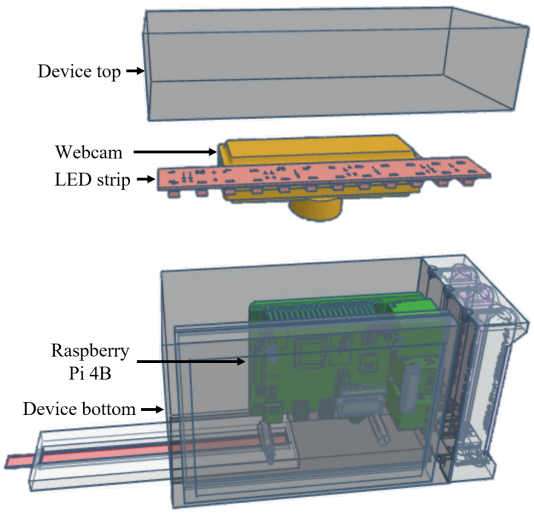


Figure 3.3: Interior view of the APSR concept diagram.

On the outside, there is a place for attaching a battery shield housing 3 batteries (3.7V each), and the touchscreen for obtaining results.

Figure 3.4 illustrates the system level assembly of the aforementioned electronic components. The positive terminal from the 11.1V battery is attached to the common terminal of the toggle switch, whereas, terminal 1 of the switch is connected in series to the positive input of the LM2596 Buck converter and LED strip. Similarly, the negative terminal of the battery shield goes to the negative terminal of the buck converter and LED strip. The buck converter is further connected to the positive and negative terminals of RPi.

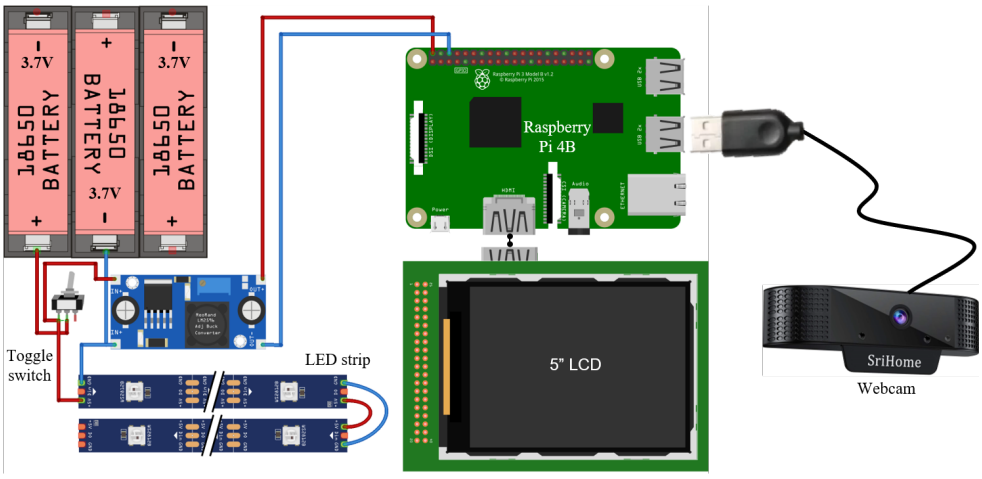


Figure 3.4: Circuit diagram of APSR.

This assembly ensures that the LED strip gets all the required 11.1V and RPi receives 5V at the input on switching ON the device. Buck converter was used due to this difference in the voltage requirement. The RPi touchscreen is simply connected to the RPi board using a ribbon connector. The USB webcam can be connected to one of the USB slots on the board.

3.3 Hardware and prototyping

Figure 3.5 showcases the prototyping phase of the bottom box, whereas, fig. 3.6 and 3.7 illustrate the final product. In order to satisfy the ASSURED criteria, we decided to use the affordable, fast and widely available 3D printing technology. The bottom box was made out foam wood first to study the impact of inner walls on the paper strips. The distance between the 2 walls (box width) was decided to be 70 mm because it successfully housed the strip holding cassette and the strip was free of any shadows from the walls. The dimensions of the box are 135x80x165 mm with the foam wood thickness of 5 mm. The box can also be 3D printed for manufacturing purposes. An experiment was performed to calculate the focal distance of the SriHome webcam. For this purpose, a Creality Ender-3 3D printer was used. The webcam was attached to the horizontal bar of the 3D printer which moves in the z-direction and the water test strip was attached to the printer bed and was secured in place with a double-sided tape (3M company). Images were taken at various z-direction values starting from z = 0mm to z = 128.425mm with an increment of 0.025, 1, and 5 mm. At z = 0mm, the distance between the cam lens and the test strip was 18mm. After analyzing the data, it was observed that the best focus was obtained at z = 55mm, and the minimum distance required between the lens and the test strip (z-value) for all the 15 test blocks to be visible was 99 mm. Therefore, z = 99mm is a prerequisite and the best focus beyond 99mm was obtained at z = 99.6mm. In order to understand the impact of LED light placement on the SriHome webcam's focal length, a prototype was made to place the 12V LED strip on the webcam such that there was one LED strip on either side of the lens. The positive and negative terminals of the strips were later soldered using connector wires and were powered using a 12V power supply. The interface was made out of foam wood and the LED strips were secured onto the foam wood with the help of a narrow double-sided tape (3M company). A 16mm diameter hole was drilled in the interface to uncover the webcam lens. Finally, the interface was placed onto the webcam with the help of a wide double-sided tape (3M company).

Figure 3.5: Prototype of the bottom half of the system made out of foam wood. (a) medical test paper strip and cassette holder, (b)-(e) foam wood prototype of the bottom box.

Figure 3.6a,b illustrate the top part of the the box which is a 3D printed enclosure to the bottom box housing a webcam and the LED strip. This part can be easily mounted on the top of the bottom box in a way that all the 15 colour blocks are visible from the webcam and the strip is within the webcam's focal length (3.6e).

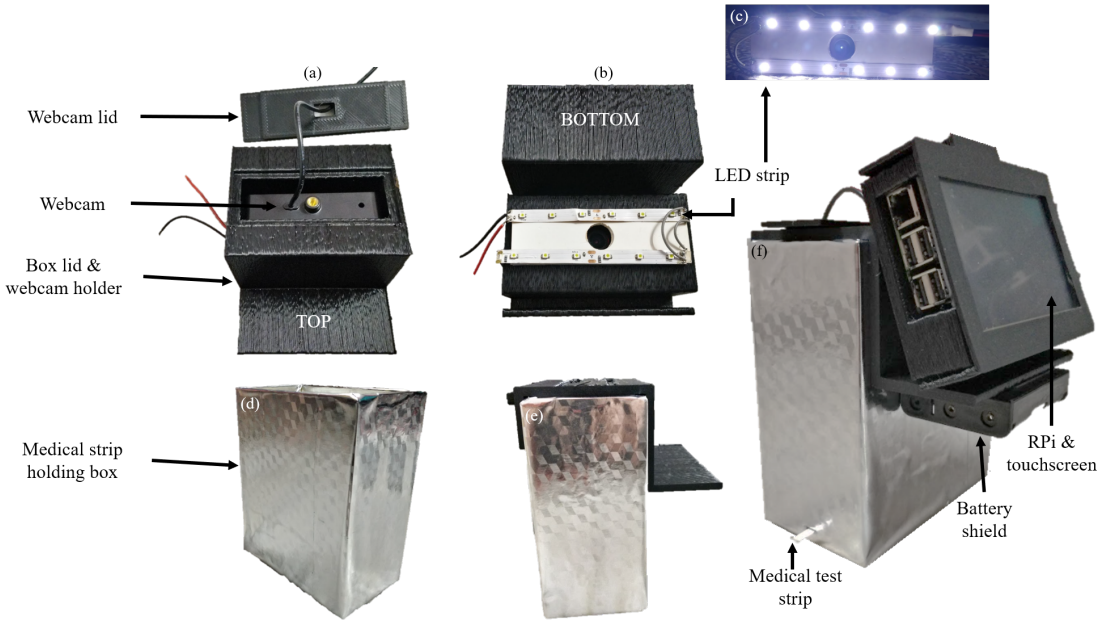


Figure 3.6: 3D printed top half of the system and fitting. (a), (b) 3D model of the webcam and LED strip holder, also acts as a lid of the device, (c) glowing LED strip at 12V, (d)-(f) combined top and bottom of the device.

Figure 3.6f represents the RPi board and screen cover. It was designed such that its peripheral pin slots are exposed for access i.e. there is an opening on the side of the case for the HDMI, USB, and Ethernet connections and another opening on the top for the power supply. The cover is also made up of 2 parts: front and back. Both these parts are connected to each other with the help of nuts, bolts and washers. This makes the cover very modular and convenient for cleaning and replacing parts. Additionally, the screen is attached at a 45°angle for ease of operating the screen when in use. As can be observed from all the hardware images, it is a minimal, modular and an efficient setup. Figure 3.7d represents the image captured with the present setup and fig. 3.5a-c illustrate the APSR's back view, top view and side view, respectively.

3.4 Software

In this section, the software building process is elucidated. The subsection 3.4.1 reports the details about how the initial dataset was prepared after integrating the hardware. And subsection 3.4.2 presents the final curve fittings used in the algorithm. The complete python codes are listed in Chapter 7 (Appendix).

3.4.1 Measurements and Image processing

The water test kit from the manufacturer Test Lab comes with a box full of paper strips and a colour chart which is stuck on the outside of the box and shown in fig. 2.2. The colour chart include 15 rows each for a water testing parameter and the column represents the different shades

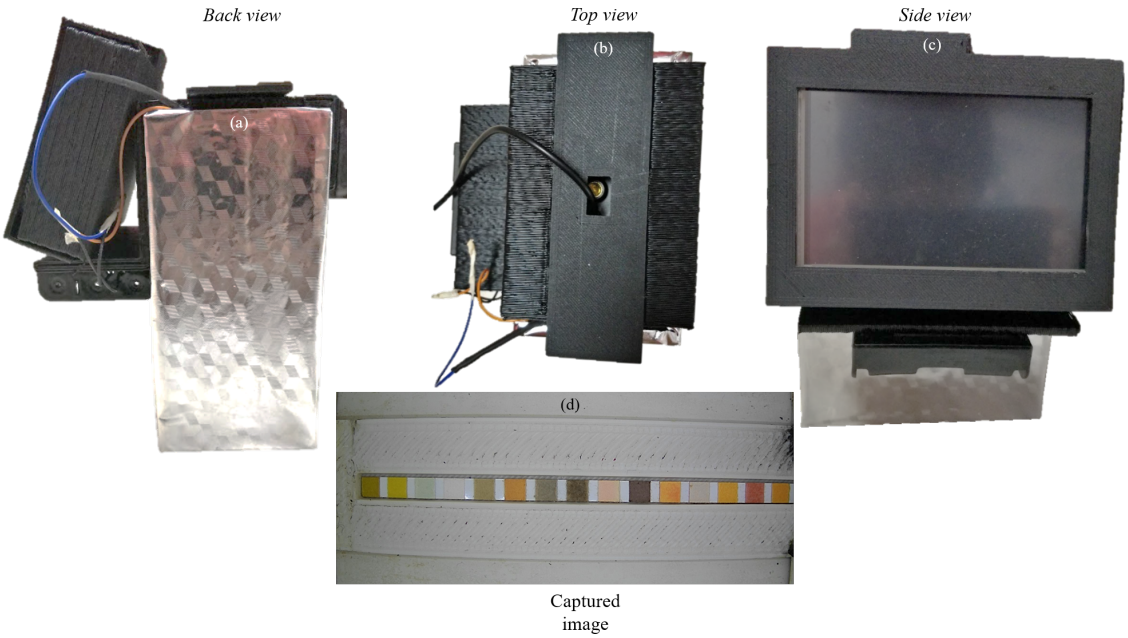


Figure 3.7: Final product. (a) back view, (b) top view, and (c) side view of the system, (d) the final captured image from the webcam.

that the paper strip can change to after dipping into the analyte. This colour change is rooted from the respective parameter concentration in the given analyte. For example, in today's date, when the clinicians are interested in analysing the water pH, they focus on the 14^th colour block of the paper strip and match it to the colour chart illustrated in fig. 3.8. Similar to pH, there rest of the parameters also their own content thresholds in the form of colour shades. The idea here was to automate to this process of manually matching the colour blocks on the paper strip to the chart so as to avoid human error.

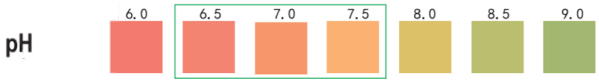


Figure 3.8: Thresholds for pH from the Test lab box chart [31].

The first step towards automation was to feed the processor with the threshold data points to later interpolate them. We did that by placing the colour chart strips individually in the APSR, clicking the pictures, and extracting and processing the obtained RGB values. We extracted RGB values for each pixel of each colour block on the chart and plotted each R, G, and B channels separately to observe a trend. We looked for the channels with linear trend so as to improve the chances of reliable colour prediction. The calibration logics behind the selected channels and their corresponding fits are explained ahead.

3.4.2 Calibration logics and curve fittings

After carefully examining individual R, G and B channels, we assigned logics for colour block of each water test strip parameter. The final calibration logics are mentioned in the second column of table 3.16.

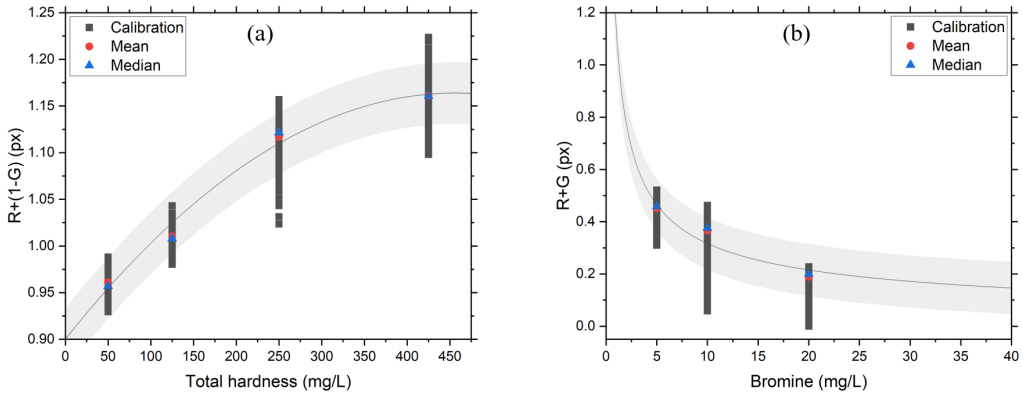


Figure 3.9: Calibration plots for a) Total Hardness and b) Bromine. x-axis represents the parameter output value and y-axis the RGB logic in pixels.

Total Hardness

For this parameter, the normalized R and G channels already had good information (plot trend) but in opposite directions so we used the logic R+(1-G) so as to get an upward trend. The curve is shown in fig. 3.9a. The nature of the calibration curve is 2^{nd} order polynomial and the formula is mentioned in table 3.1.

Table 3.1: Curve fitting formula and system calibration values for the parameter Total Hardness.

y = (0.90009) + (0.00116) * x + (-1.2651E-6) * x ^{2}	
	Calibration
Adj. R-Square	0.95747
Intercept	0.90009 ± 3.8359E-4
B1	0.00116 ± 3.98847E-6
B2	-1.2651E-6 ± 7.99625E-9

Bromine

The normalized R and G channels also already had good information so we used the logic R+G. The curve is shown in fig. 3.9b. The nature of the calibration curve and the formula is mentioned in table 3.2.

Table 3.2: Curve fitting formula and system calibration values for the parameter Bromine.

y = a*x^b	
	Calibration
Reduced Chi-Sqr	0.00258
Adj. R-Square	0.82109
a	1.14237 ± 0.00593
b	-0.55727 ± 0.00252

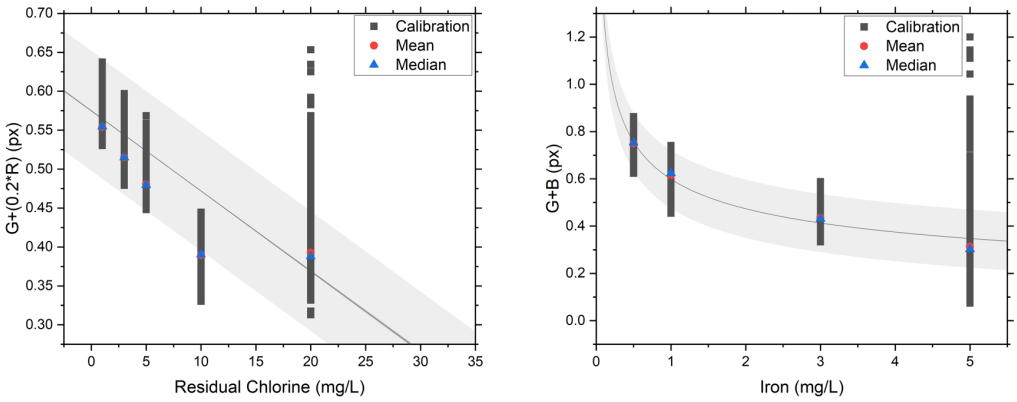


Figure 3.10: Calibration plots for a) Residual Chlorine and b) Iron. x-axis represents the parameter output value and y-axis the RGB logic in pixels.

Residual Chlorine

This parameter lacked any trend/information in all the R, G as well as B channels. In order to make sense out of the available information, we added a tuning factor of 0.2, so we used the logic G+(0.2*R). The curve is shown in fig. 3.10a. The nature of the calibration curve is linear and the formula is mentioned in table 3.3.

Table 3.3: Curve fitting formula and system calibration values for the parameter Residual Chlorine.

y = (0.57494) + (-0.0103022) * x	
	Calibration
Pearson's r	-0.85929
Adj. R-Square	0.73837
Intercept	0.57494 ± 4.10582E-4
Slope	-0.0103 ± 4.47879E-5

Iron

The normalized B and G channels also already had good information so we used the logic B+G. The curve is shown in fig. 3.10b. The nature of the calibration curve and the formula is mentioned in table 3.4.

Table 3.4: Curve fitting formula and system calibration values for the parameter Iron.

$y = a*x^b$	
	Calibration
Reduced Chi-Sqr	0.00393
Adj. R-Square	0.87767
a	0.59865 ± 5.44192E-4
b	-0.33806 ± 0.00118

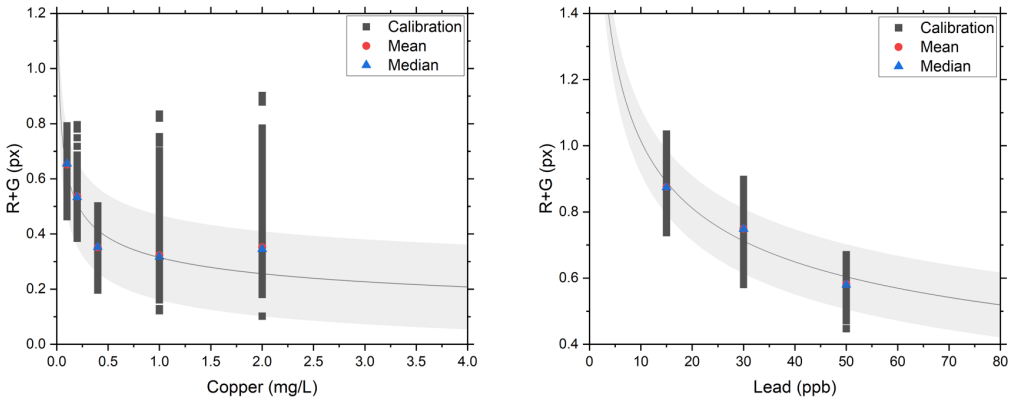


Figure 3.11: Calibration plots for a) Copper and b) Lead. x-axis represents the parameter output value and y-axis the RGB logic in pixels.

Copper

The normalized R and G channels also already had good information so we used the logic R+G. The curve is shown in fig. 3.11a. The nature of the calibration curve and the formula is mentioned in table 3.5.

Table 3.5: Curve fitting formula and system calibration values for the parameter Copper.

$y = a*x^b$	
	Calibration
Reduced Chi-Sqr	0.00616
Adj. R-Square	0.70593
a	0.31472 ± 8.84748E-4
b	-0.29873 ± 0.00162

Lead

The normalized R and G channels also already had good information so we used the logic R+G. The curve is shown in fig. 3.11b. The nature of the calibration curve and the formula is mentioned in table 3.6.

Table 3.6: Curve fitting formula and system calibration values for the parameter Lead.

$y = a*x^b$	
	Calibration
Reduced Chi-Sqr	0.0025
Adj. R-Square	0.84692
a	2.1325 ± 0.00824
b	-0.32246 ± 0.00118

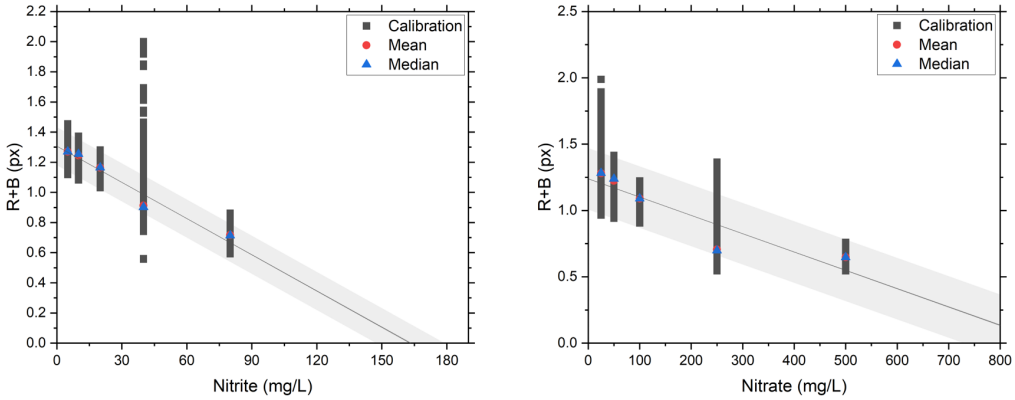


Figure 3.12: Calibration plots for a) Nitrite and b) Nitrate. x-axis represents the parameter output value and y-axis the RGB logic in pixels.

Nitrite

The normalized R and B channels had good information so we used the logic R+B (table 3.16). The curve is shown in fig. 3.12a. The nature of the calibration curve is linear and the formula is mentioned in table 3.7.

Table 3.7: Curve fitting formula and system calibration values for the parameter Nitrite.

y = (1.30685) + (-0.00800599) * x	
	Calibration
Pearson's r	-0.94514
Adj. R-Square	0.89328
Intercept	1.30685 ± 6.14421E-4
Slope	-0.00801 ± 1.79326E-5

Nitrate

The normalized R and B channels had good information so we used the logic R+B (table 3.16). The curve is shown in fig. 3.12b. The nature of the calibration curve is linear and the formula is mentioned in table 3.8.

Table 3.8: Curve fitting formula and system calibration values for the parameter Nitrate.

y = (1.23892) + (-0.00138032) * x	
	Calibration
Pearson's r	-0.89557
Adj. R-Square	0.80204
Intercept	1.23892 ± 0.00107
Slope	-0.00138 ± 4.2363E-6

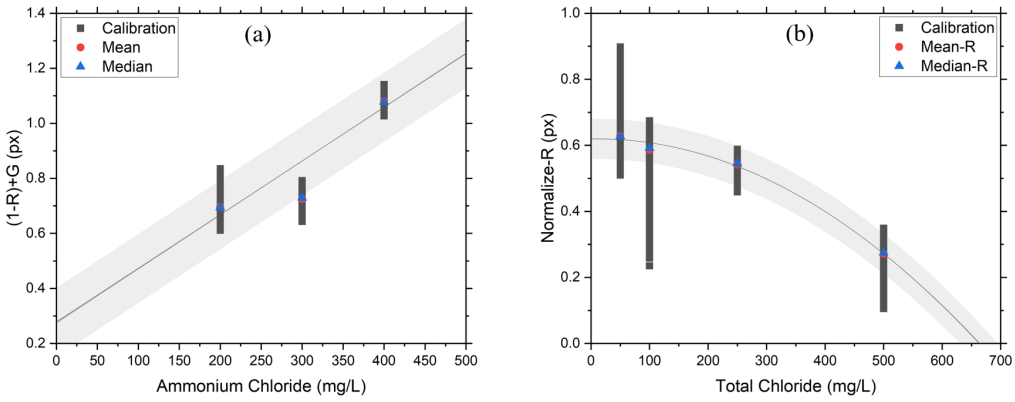


Figure 3.13: Calibration plots for a) Ammonium Chloride and b) Total Chloride. x-axis represents the parameter output value and y-axis the RGB logic in pixels.

Ammonium Chloride

The normalized R and B channels had good information but in opposite directions so we used the logic (1-R)+G (table 3.16). The curve is shown in fig. 3.13a. The nature of the calibration curve is linear and the formula is mentioned in table 3.9.

Table 3.9: Curve fitting formula and system calibration values for the parameter Ammonium Chloride.

y = (0.276681) + (0.00195505) * x	
	Calibration
Pearson's r	0.94182
Adj. R-Square	0.88701
Intercept	0.27668 ± 0.00211
Slope	0.00196 ± 6.51563E-6

Total Chloride

The normalized R channel already had good information so we used the same logic R (table 3.16). The curve is shown in fig. 3.13b. The nature of the calibration curve is 2^{nd} order polynomial and the formula is mentioned in table 3.10.

Table 3.10: Curve fitting formula and system calibration values for the parameter Total Chloride.

y = (0.61903) + (3.36811E-5) * x + (-1.45573E-6) * x^2	
	Calibration
Adj. R-Square	0.95316
Intercept	0.61903 ± 6.54784E-4
B1	3.36811E-5 ± 6.37666E-6
B2	-1.45573E-6 ± 1.10928E-8

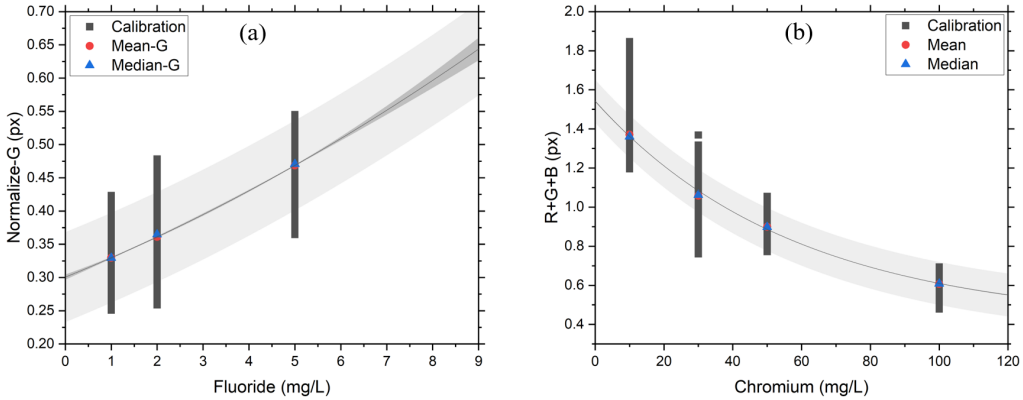


Figure 3.14: Calibration plots for a) Fluoride and b) Chromium. x-axis represents the parameter output value and y-axis the RGB logic in pixels.

Fluoride

The normalized G channel already had good information so we used the same logic G (table 3.16). The curve is shown in fig. 3.14a. The nature of the calibration curve is 2^{nd} order polynomial and the formula is mentioned in table 3.11.

Table 3.11: Curve fitting formula and system calibration values for the parameter Fluoride.

y = (0.30054) + (0.02798) * x + (0.00113) * x^2	
	Calibration
Adj. R-Square	0.76435
Intercept	0.30054 ± 0.00198
B1	0.02798 ± 0.00175
B2	0.00113 ± 2.75958E-4

Chromium

All the normalized R, G and B channels had a coherent trend, so we used the logic R+G+B (table 3.16). The curve is shown in fig. 3.14b. The nature of the calibration curve is exponential and the formula is mentioned in table 3.12.

Table 3.12: Curve fitting formula and system calibration values for the parameter Chromium.

y = a-b*c^x	
	Calibration
Reduced Chi-Sqr	0.00313
Adj. R-Square	0.95688
a	0.40762 ± 0.00346
b	-1.13464 ± 0.00268
c	0.98289 ± 1.26248E-4

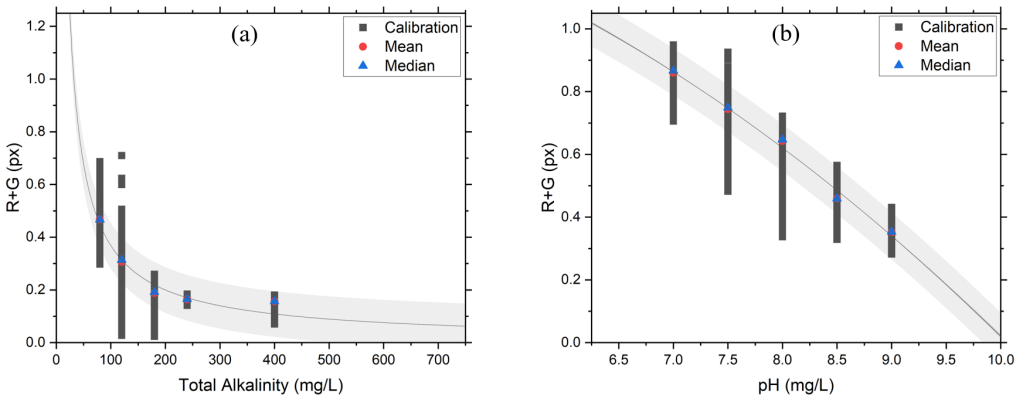


Figure 3.15: Calibration plots for a) Total Alkalinity and b) pH. x-axis represents the parameter output value and y-axis the RGB logic in pixels.

Total Alkalinity

All the normalized R and G channels had a coherent trend, so we used the logic R+G (table 3.16). The curve is shown in fig. 3.15a. The nature of the calibration curve is exponential and the formula is mentioned in table 3.13.

Table 3.13: Curve fitting formula and system calibration values for the parameter Total Alkalinity.

$y = a*x^b$	
	Calibration
Reduced Chi-Sqr	0.00195
Adj. R-Square	0.85035
a	21.37432 ± 0.28469
b	-0.8818 ± 0.00277

pH

All the normalized R and G channels had a coherent trend, so we used the logic R+G (table 3.16). The curve is shown in fig. 3.15b. The nature of the calibration curve is 2^{nd} order polynomial and the formula is mentioned in table 3.14.

Table 3.14: Curve fitting formula and system calibration values for the parameter pH.

$y = (1.47138) + (0.04886) * x + (-0.0194) * x^2$	
	Calibration
Adj. R-Square	0.95658
Intercept	1.47138 ± 0.04088
B1	0.04886 ± 0.01028
B2	-0.0194 ± 6.42041E-4

Table 3.16: RGB value logic and R^2 value chart for 15 water test parameters.

Parameters	**RGB value logic**	**R^2**
Total hardness	R+(1-G)	0.957
Bromine	R+G	0.821
Residual chlorine	G+(0.2*R)	0.738
Iron	G+B	0.877
Copper	R+G	0.705
Lead	R+G	0.846
Nitrite	R+B	0.893
Nitrate	R+B	0.802
Ammonium Chloride	(1-R)+G	0.887
Total Chloride	R	0.953
Fluoride	G	0.764
Chromium/CR(VI)	R+G+B	0.956
Total Alkalinity	R+G	0.850
pH	R+G	0.956
Cyanuric Acid	G	0.923

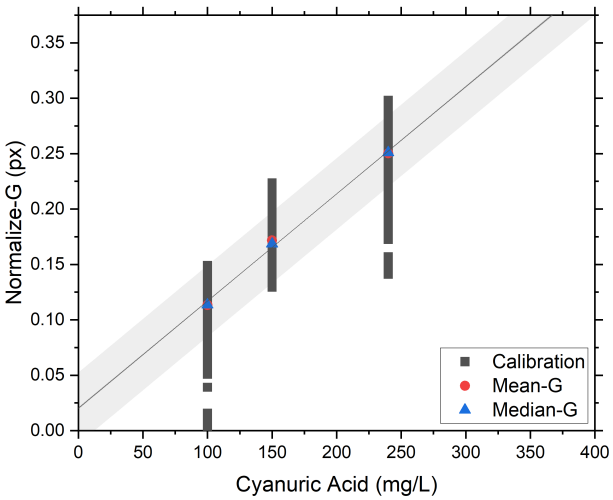


Figure 3.16: Calibration plot for Cyanuric Acid. x-axis represents the parameter output value and y-axis the RGB logic in pixels.

Cyanuric Acid

All the normalized G channels had an informative trend, so we used the same logic G (table 3.16). The curve is shown in fig. 3.16. The nature of the calibration curve is linear and the formula is mentioned in table 3.15.

Table 3.15: Curve fitting formula and system calibration values for the parameter Cyanuric Acid.

y = (0.0202479) + (0.000967136) * x	
	Calibration
Pearson's r	0.96087
Adj. R-Square	0.92327
Intercept	0.02025 ± 4.44835E-4
Slope	9.67136E-4 ± 2.55376E-6

3.5 Clinical study

A clinical study was performed to compare the performance of APSR in comparison with human participants. Four participants from different age group, gender and knowledge background had participated in the study. Chapter 4 reports a detailed comparison of the obtained results.

3.6 System protocol

This section is divided into 2 parts: step-by-step procedure for using APSR and the corresponding elements of Graphical User Interface (GUI) elements.

3.6.1 Procedure

The step-by-step procedure to use APSR is as follows:

- Turn ON the toggle switch to start supplying power to the system
- Dip the medical paper strip into the analyte and let it dry for 2 to 5s
- Insert the strip in the device through the opening and track the live camera feed to ensure proper placement
- Press "Start" on the screen to get the results
- Press "Export" on the screen to email to those results to your clinician

3.6.2 Graphical User Interface

We designed a GUI to be displayed on the RPi screen as illustrated in fig. 3.17a,b. On the left side of the screen, there is a live camera feed to track the insertion of the paper strip and there is a text box for displaying the results on the right side. The live feed stays on by default for monitoring the device interior. There are 2 buttons named "Start" and "Export" at the bottom left corner and their functions are collinear to their names. The Start button captures the image of the test strip and presents the results to the right side of the screen. Export button is meant to send the obtained results to the clinician via email. Figure 3.17c shows an example of the received email.

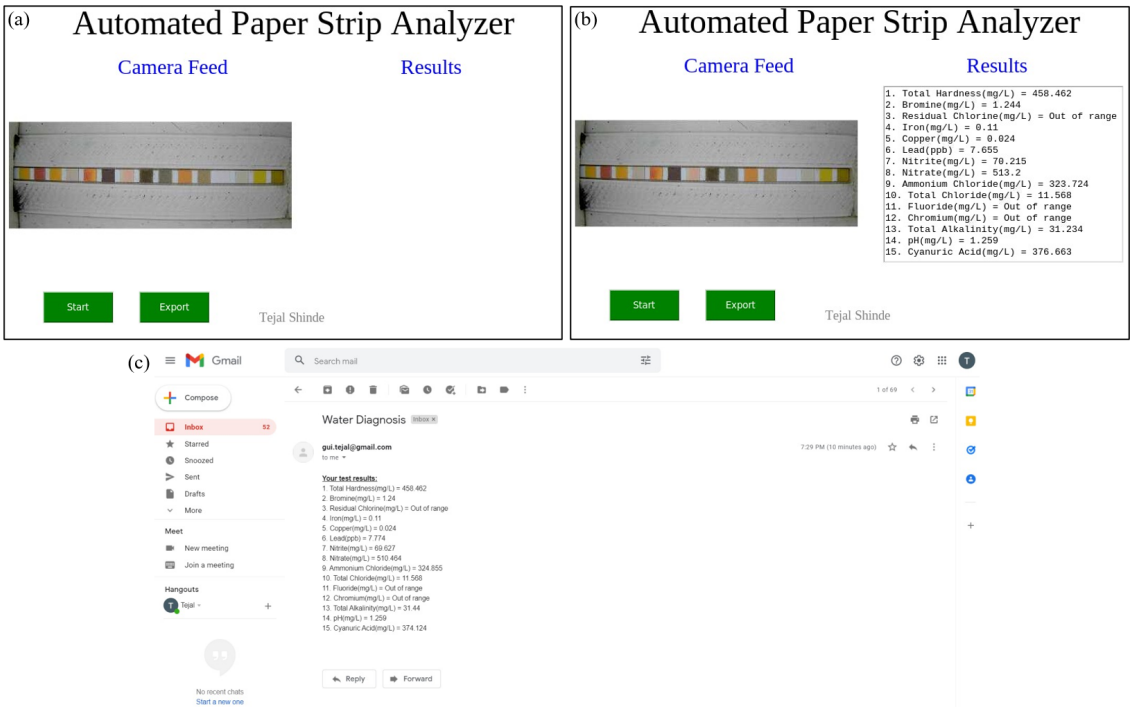


Figure 3.17: Graphical User Interface of the proposed system. (a) default screen, (b) screen after pressing "Start", (c) received results over the email on pressing "Export".

Chapter 4

Results

In this chapter, all the research goals are presented in the form of results obtained for all the 15 water test parameters.

Figure **??** illustrates the intra-sample variance in the result prediction. Higher variance (standard deviation) entails biased and thus unpredictable diagnosis. Figure **??** illustrates the true results obtained from the lab water test meter, human participants as well as from APSR. From table 4.1, the total accuracy of APSR was 88.7% and accuracy of human participants was 60.7%.

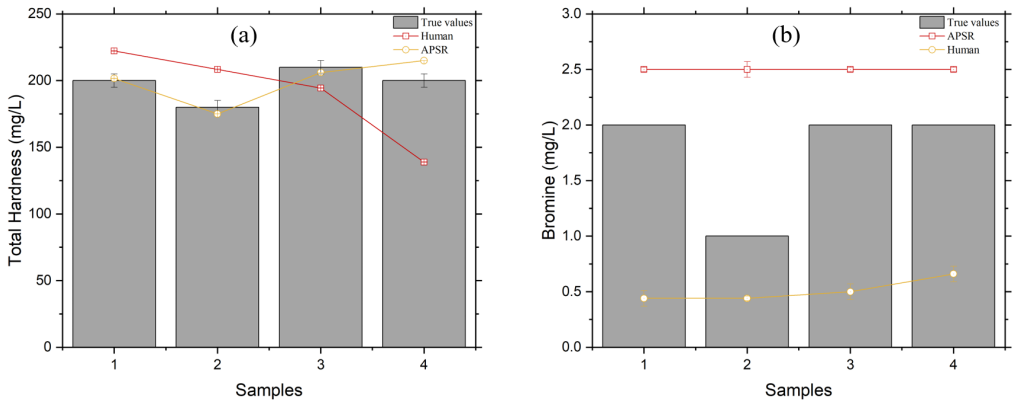


Figure 4.1: True readings for Total Hardness and Bromine obtained from laboratory deal water hardness kit, human participants from the clinical study and proposed system (APSR). x-axis illustrates the sample count and y-axis represents the obtained respective results. The red line indicates data from human participants and yellow lines stand for APSR data.

4.1 Total Hardness

Figure 4.1a and fig. 4.2a illustrates that the results obtained for the Total Hardness from human participants during the clinical study had high variance as compared to APSR. The accuracy of humans was 76% and accuracy for APSR was 95% (table 4.1) and the R^2 value of APSR calibration curve was 0.95 (table 3.16).

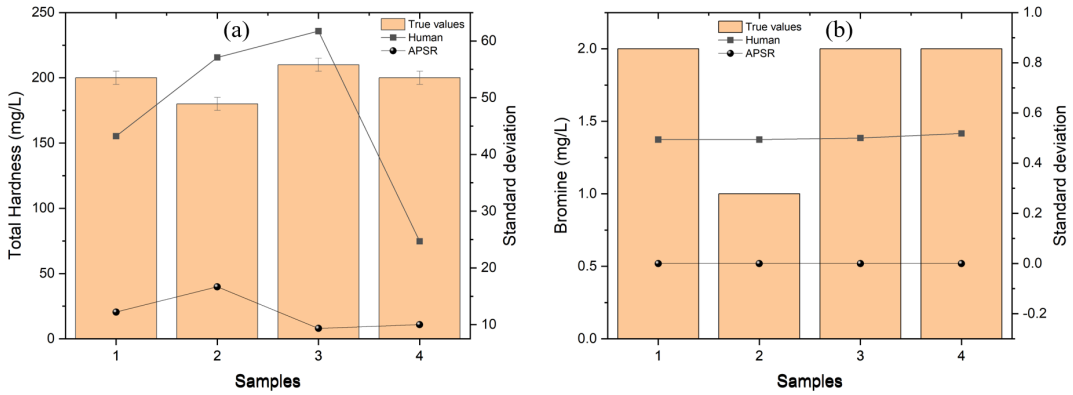


Figure 4.2: True readings for Total Hardness and Bromine from 4 lab samples vs the standard deviation of human and APSR (proposed system) predictions. x-axis illustrates the sample count, left y-axis depicts the true output of the respective tests and right y-axis represents the standard deviation scale for the variance in result prediction by humans and the proposed system (APSR). The square data points represent human data and circles represent APSR data.

4.2 Bromine

Figure 4.1b and fig. 4.2b illustrates that the results obtained for the Bromine from human participants during the clinical study very had high variance as compared to APSR. APSR gave consistent output with 97% accuracy and that for humans was 94% (table 4.1) and the R^2 value of APSR calibration curve was 0.82 (table 3.16).

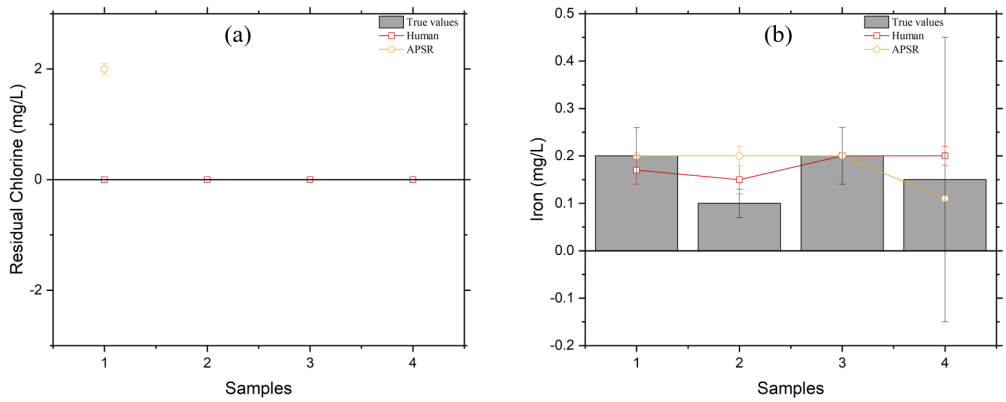


Figure 4.3: True readings for a) Residual Chlorine and b) Iron obtained from laboratory deal water hardness kit, human participants from the clinical study and proposed system (APSR). x-axis illustrates the sample count and y-axis represents the obtained respective results. The red line indicates data from human participants and yellow lines stand for APSR data.

4.3 Residual Chlorine

Figure 4.3a and fig. 4.4a illustrates that the results obtained for the Residual Chlorine from human participants during the clinical study no variance for both APSR as well as humans. In fact, APSR could not detect the output at all as the true output was 0 and values obtained by APSR algorithm

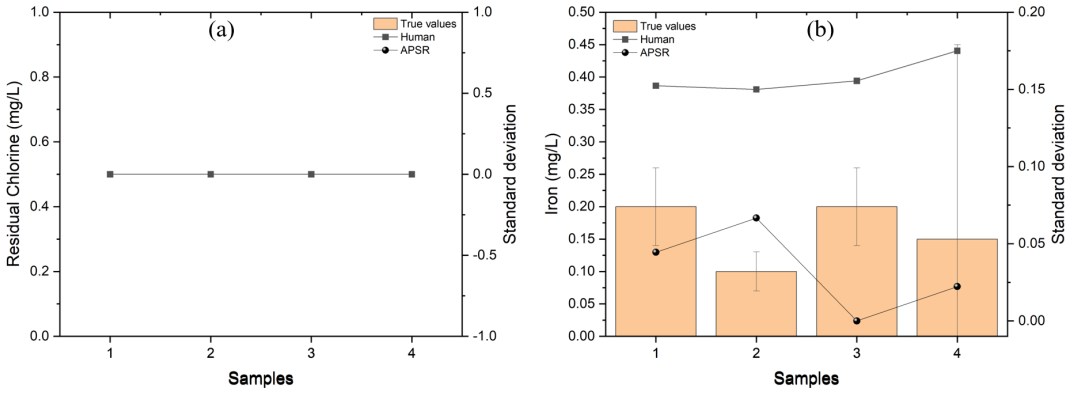


Figure 4.4: True readings for a) Residual Chlorine and b) Iron from 4 lab samples vs the standard deviation of human and APSR (proposed system) predictions. x-axis illustrates the sample count, left y-axis depicts the true output of the respective tests and right y-axis represents the standard deviation scale for the variance in result prediction by humans and the proposed system (APSR). The square data points represent human data and circles represent APSR data.

turned out to be negative which resulted in 15% accuracy and that for humans was 56% (table 4.1). The R^2 value of APSR calibration curve was 0.73 (table 3.16).

4.4 Iron

Figure 4.3b and fig. 4.4b illustrates that the results obtained for the Iron from human participants during the clinical study had high variance as compared to APSR. APSR gave a steady output. The accuracy of humans was 63% and accuracy for APSR was 100% (table 4.1) and the R^2 value of APSR calibration curve was 0.87 (table 3.16).

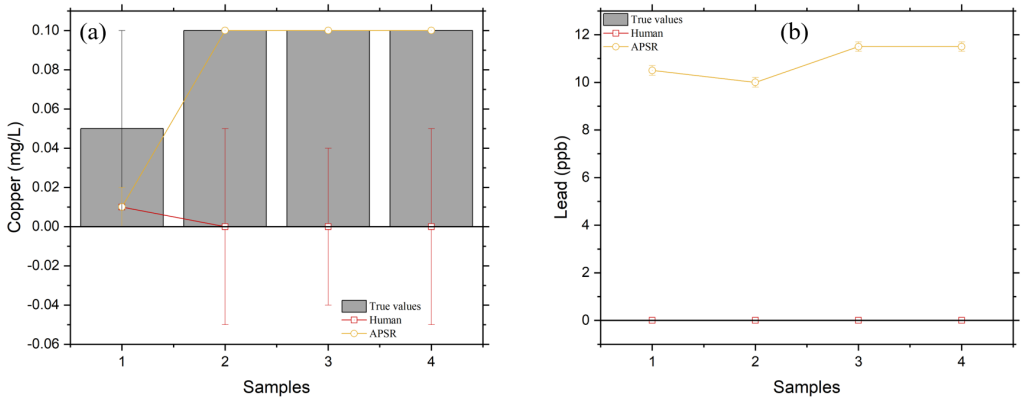


Figure 4.5: True readings for a) Copper and b) Lead obtained from laboratory deal water hardness kit, human participants from the clinical study and proposed system (APSR). x-axis illustrates the sample count and y-axis represents the obtained respective results. The red line indicates data from human participants and yellow lines stand for APSR data.

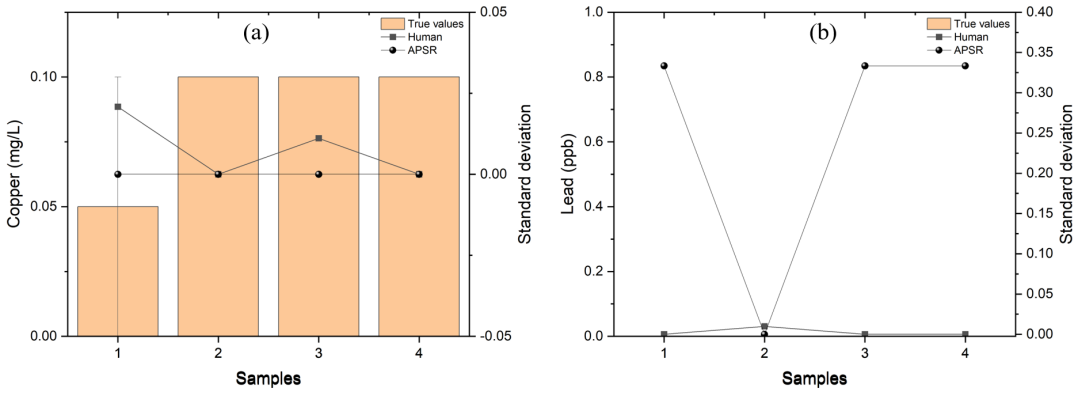


Figure 4.6: True readings for a) Copper and b) Lead from 4 lab samples vs the standard deviation of human and APSR (proposed system) predictions. x-axis illustrates the sample count, left y-axis depicts the true output of the respective tests and right y-axis represents the standard deviation scale for the variance in result prediction by humans and the proposed system (APSR). The square data points represent human data and circles represent APSR data.

4.5 Copper

Figure 4.5a and fig. 4.5a illustrates that the results obtained for the Copper from human participants during the clinical study had high variance as compared to APSR. APSR gave a steady output. The accuracy of humans was 40% and accuracy for APSR was 100% (table 4.1) and the R^2 value of APSR calibration curve was 0.70 (table 3.16).

4.6 Lead

Figure 4.5b and fig. 4.5b illustrates that the results obtained for the Lead from APSR had high variance as compared to human participants during the clinical study. Also, the accuracy of humans was 100% and accuracy for APSR was 79% (table 4.1) and the R^2 value of APSR calibration curve was 0.84 (table 3.16).

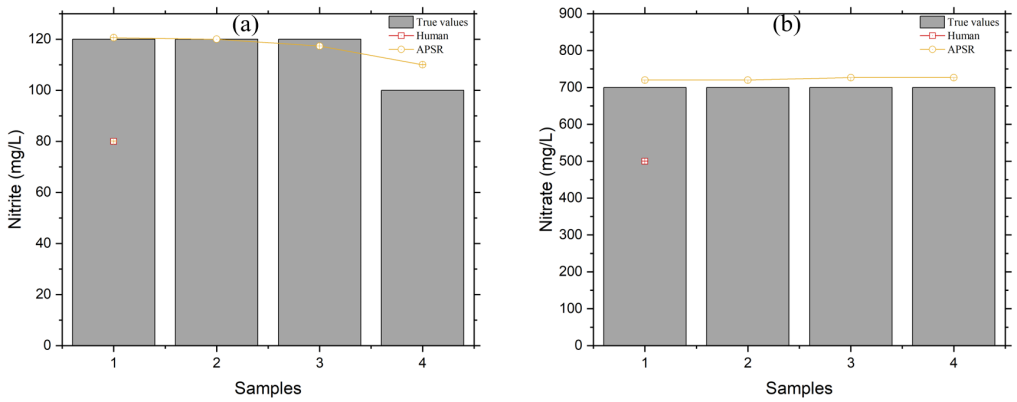


Figure 4.7: True readings for a) Nitrite and b) Nitrate obtained from laboratory deal water hardness kit, human participants from the clinical study and proposed system (APSR). x-axis illustrates the sample count and y-axis represents the obtained respective results. The red line indicates data from human participants and yellow lines stand for APSR data.

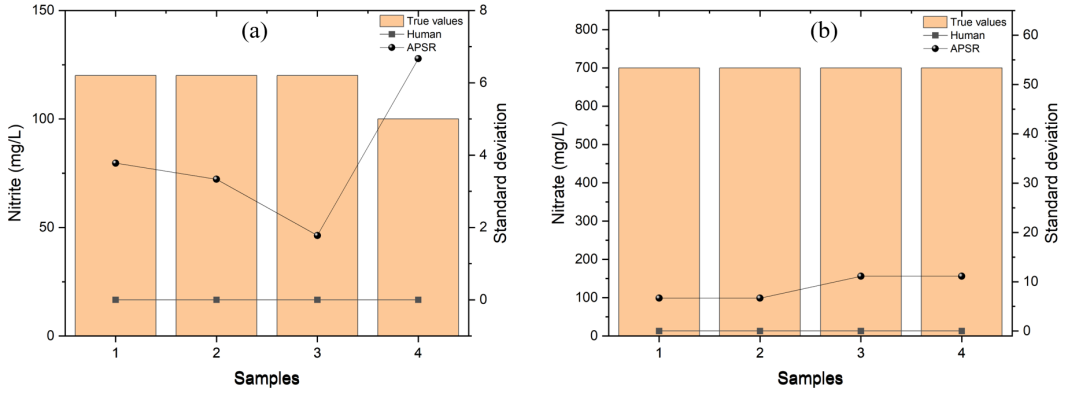


Figure 4.8: True readings for a) Nitrite and b) Nitrate from 4 lab samples vs the standard deviation of human and APSR (proposed system) predictions. x-axis illustrates the sample count, left y-axis depicts the true output of the respective tests and right y-axis represents the standard deviation scale for the variance in result prediction by humans and the proposed system (APSR). The square data points represent human data and circles represent APSR data.

4.7 Nitrite

Figure 4.7a and fig. 4.8a illustrates that the results obtained for the Nitrite from APSR had a little variance, whereas, no human participant could give any prediction. According to the cohort, the colour shade was not present on the colour chart. Therefore, the accuracy of humans was 0% and accuracy for APSR was 97% (table 4.1) and the R^2 value of APSR calibration curve was 0.89 (table 3.16).

4.8 Nitrate

Figure 4.7b and fig. 4.8b illustrates that the results obtained for the Nitrate from APSR had a little variance, whereas, no human participant could give any prediction. According to the cohort, the colour shade was not present on the colour chart. Therefore, the accuracy of humans was 0% and accuracy for APSR was 98% (table 4.1) and the R^2 value of APSR calibration curve was 0.80 (table 3.16).

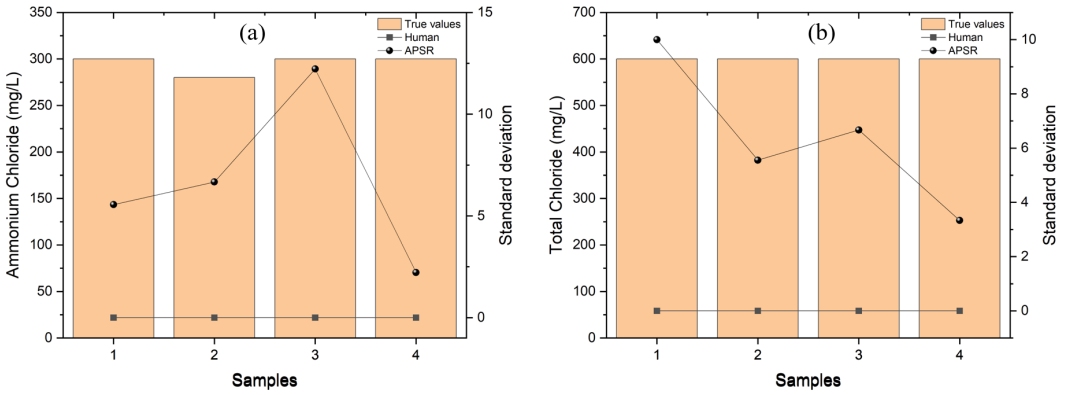


Figure 4.10: True readings for a) Ammonium Chloride and b) Total Chloride from 4 lab samples vs the standard deviation of human and APSR (proposed system) predictions. x-axis illustrates the sample count, left y-axis depicts the true output of the respective tests and right y-axis represents the standard deviation scale for the variance in result prediction by humans and the proposed system (APSR). The square data points represent human data and circles represent APSR data.

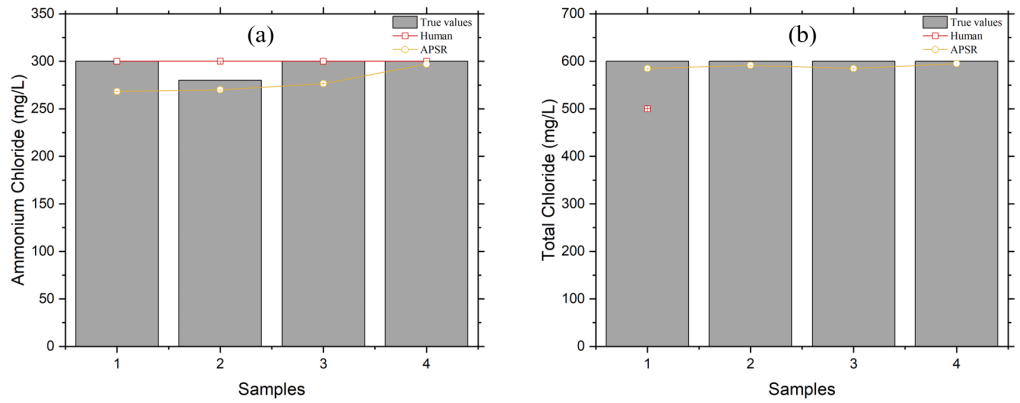


Figure 4.9: True readings for a) Ammonium Chloride and b) Total Chloride obtained from laboratory deal water hardness kit, human participants from the clinical study and proposed system (APSR). x-axis illustrates the sample count and y-axis represents the obtained respective results. The red line indicates data from human participants and yellow lines stand for APSR data.

4.9 Ammonium Chloride

Figure 4.9a and fig. 4.10a illustrates that the results obtained for the Ammonium Chloride from APSR had high variance as compared to human participants during the clinical study. The accuracy of humans was 86% and accuracy for APSR was 96% (table 4.1) and the R^2 value of APSR calibration curve was 0.88 (table 3.16).

4.10 Total Chloride

Figure 4.9b and fig. 4.10b illustrates that the results obtained for the Total Chloride from APSR had a little variance, whereas, no human participant could give any prediction. According to the

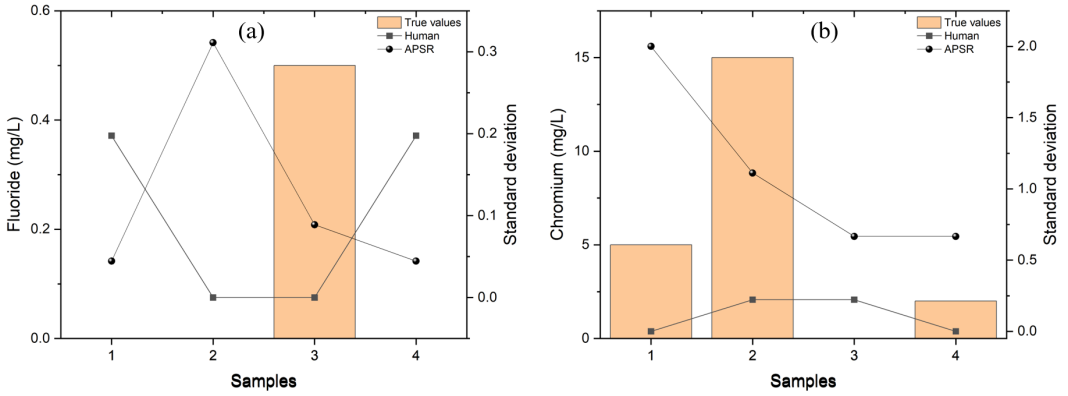


Figure 4.12: True readings for a) Fluoride and b) Chromium from 4 lab samples vs the standard deviation of human and APSR (proposed system) predictions. x-axis illustrates the sample count, left y-axis depicts the true output of the respective tests and right y-axis represents the standard deviation scale for the variance in result prediction by humans and the proposed system (APSR). The square data points represent human data and circles represent APSR data.

cohort, the colour shade was not present on the colour chart. Therefore, the accuracy of humans was 0% and accuracy for APSR was 99% (table 4.1) and the R^2 value of APSR calibration curve was 0.95 (table 3.16).

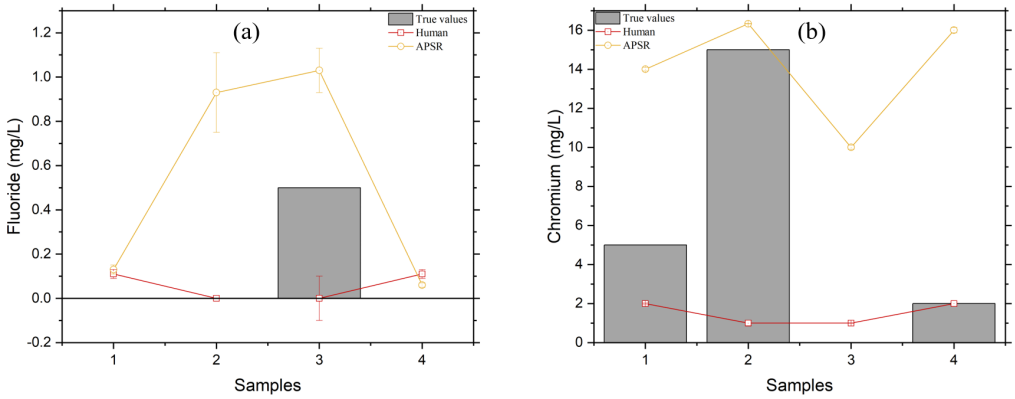


Figure 4.11: True readings for a) Fluoride and b) Chromium obtained from laboratory deal water hardness kit, human participants from the clinical study and proposed system (APSR). x-axis illustrates the sample count and y-axis represents the obtained respective results. The red line indicates data from human participants and yellow lines stand for APSR data.

4.11 Fluoride

Figure 4.11a and fig. 4.12a illustrates that the results obtained for Fluoride from APSR and human participants both had high variance. The accuracy of humans was 92% and accuracy for APSR was also 92% (table 4.1) and the R^2 value of APSR calibration curve was 0.76 (table 3.16).

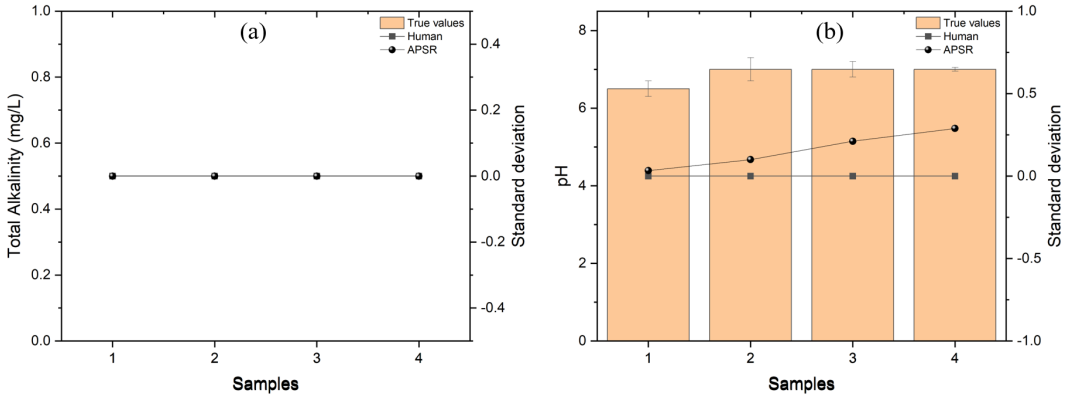


Figure 4.14: True readings for a) Total Alkalinity and b) pH from 4 lab samples vs the standard deviation of human and APSR (proposed system) predictions. x-axis illustrates the sample count, left y-axis depicts the true output of the respective tests and right y-axis represents the standard deviation scale for the variance in result prediction by humans and the proposed system (APSR). The square data points represent human data and circles represent APSR data.

4.12 Chromium

Figure 4.11b and fig. 4.12b illustrates that the results obtained for the Chromium from APSR had high variance as compared to human participants during the clinical study. However, many human participants could not match the colour at all. The accuracy of humans was 22% and accuracy for APSR was 92% (table 4.1) and the R^2 value of APSR calibration curve was 0.95 (table 3.16).

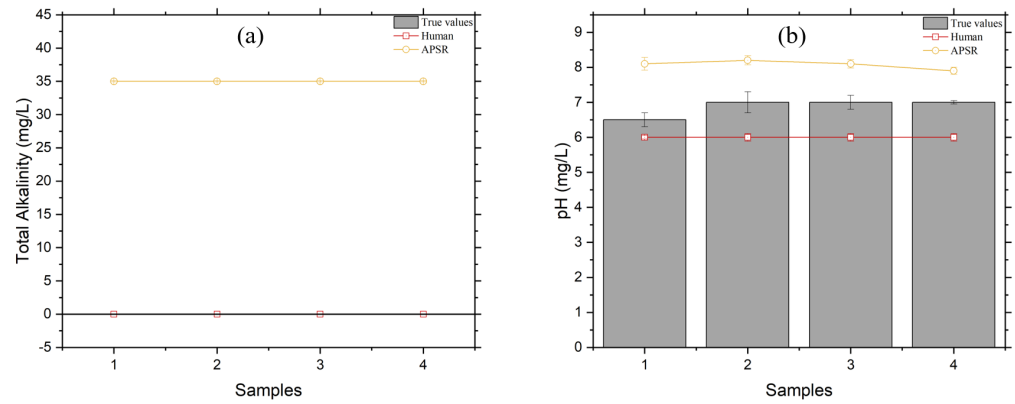


Figure 4.13: True readings for a) Total Alkalinity and b) pH obtained from laboratory deal water hardness kit, human participants from the clinical study and proposed system (APSR). x-axis illustrates the sample count and y-axis represents the obtained respective results. The red line indicates data from human participants and yellow lines stand for APSR data.

4.13 Total Alkalinity

Figure 4.13a and fig. 4.14a illustrates that the results obtained for Total Alkalinity from APSR and human participants both gave steady output with no variance. The accuracy of humans was 100% and accuracy for APSR was 92% (table 4.1) and the R^2 value of APSR calibration curve

was 0.85 (table 3.16).

4.14 pH

Figure 4.13b and fig. 4.14b illustrates that the results obtained for pH from APSR had a little higher variance as compared to human participants during the clinical study. The accuracy of humans was 91% and accuracy for APSR was 87% (table 4.1) and the R^2 value of APSR calibration curve was 0.85 (table 3.16).

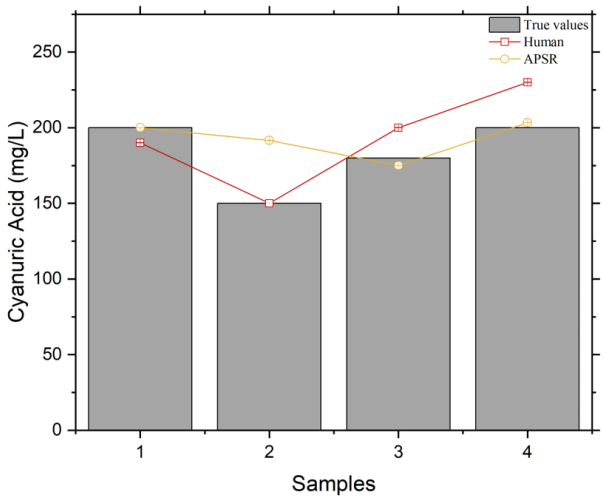


Figure 4.15: True readings for Cyanuric Acid obtained from laboratory deal water hardness kit, human participants from the clinical study and proposed system (APSR). x-axis illustrates the sample count and y-axis represents the obtained respective results. The red line indicates data from human participants and yellow lines stand for APSR data.

Table 4.1: Accuracy comparison chart between human and APSR prediction for the 15 water test parameters based on 4 clinical sample studies.

Parameters	Human accuracy (%)				APSR accuracy (%)				Total accuracy (%)	
	Sample 1	Sample 2	Sample 3	Sample 4	Sample 1	Sample 2	Sample 3	Sample 4	Human	APSR
Total hardness	76.11	71.15	73.12	83.01	94.47	91.88	94.99	96.47	76	95
Bromine	92.22	97.22	92.22	92.22	97.5	92.5	97.5	97.5	94	97
Residual chlorine	66.66	66.66	55.55	33.33	60	-	-	-	56	15
Iron	75.56	64.67	65.33	43.78	99.33	98	100	99.33	63	100
Copper	43.61	31.66	53.05	31.66	97.5	100	100	100	40	100
Lead	100	99.98	100	100	79	80	77	77	100	79
Nitrite	-	-	-	-	97.33	97.78	98.22	93.33	0	97
Nitrate	-	-	-	-	97.5	97.5	96.6	96.6	0	98
Ammonium Chloride	88.88	86.66	88.88	77.77	92.08	97.5	94.16	99.16	86	96
Total Chloride	-	-	-	-	97.5	98.6	97.5	99.16	0	99
Fluoride	97.78	100	70	97.78	97.33	81.33	89.33	98.67	92	92
Chromium/CR(VI)	21.55	19.11	33	11.11	91	98.66	90	86	22	92
Total Alkalinity	100	100	100	100	91.25	91.25	91.25	91.25	100	92
pH	94.44	88.88	88.88	88.88	81.66	86.11	87.59	89.63	91	87
Cyanuric Acid	86.98	100	86.66	88.25	96.19	88.09	93.81	88.57	91	92

4.15 Cyanuric Acid

Figure 4.15 and fig. 4.16 illustrates that the results obtained for Cyanuric Acid from APSR and human participants both had high variance. The accuracy of humans was 91% and accuracy for

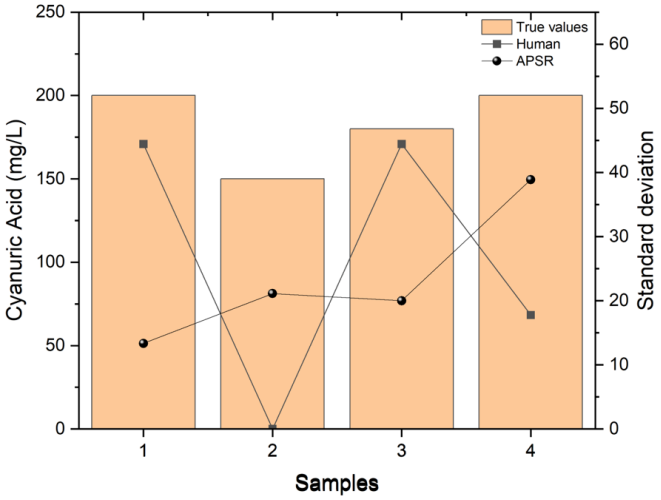


Figure 4.16: True readings for Total Hardness and Bromine from 4 lab samples vs the standard deviation of human and APSR (proposed system) predictions. x-axis illustrates the sample count, left y-axis depicts the true output of the respective tests and right y-axis represents the standard deviation scale for the variance in result prediction by humans and the proposed system (APSR). The square data points represent human data and circles represent APSR data.

APSR was 92% (table 4.1) and the R^2 value of APSR calibration curve was 0.92 (table 3.16).

The APSR algorithm was 41%, 21%, 8% and 4% less accurate than humans in predicting Residual chlorine, Lead, Total alkalinity and pH, respectively.

Chapter 5

Discussions

The clinical study results show that the diagnosis made by the human participants had high variance and low accuracy. APSR was successful in minimizing human intervention in diagnosis, which proves to be promising for solving human induced error. The system's simple and userfriendly GUI is the cornerstone in achieving the human intervention-free results. Another key interference that has the potential to disrupt the results is that from the ambient light. This is taken care of in the hardware design and is addressed by making an enclosed system. Here is an example of yet another drawback leading from human diagnosis. For example, if a clinician or a laboratory assistant is to analyze pH from a colour strip, the chart showcases prediction thresholds with an incremental interval of 0.5 (fig. 3.8. If we observe the colour shades for pH = 6.0 and 6.5 in fig. 3.8, they almost look the same to an average human. So there are two takeaways here: 1) there is no way to predict an output if the result lies in between the manufacturer mentioned thresholds, for instance, if the true pH is 6.3; 2) even in the best lighting conditions, it can be to accurately match the analyte strip to 2 similar shades. The APSR software is capable of addressing that as well with help of curve fittings and ineterpolation.

Current technologies that are already out in the market are very expensive i.e. in the range of €1000. This is not ideal for even the average income hospitals in the low-income and developing countries, let alone small clinics, labs and POC facilities. Paper being the cheapest accessory, medical test paper strips are a boon to field of diagnostics costing approximately 10 cents per strip. Apart from the cost, another drawback from the existing technology that APSR has overcome is the accuracy itself. Where today's commercial products are using photodiode and phototransistor mechanism for sensing a point in the image, APSR is capable of sensing all the 15 test parameters simultaneously thus improving the speed of the system. Fast systems are an asset for common people and hospitals as quick results leads to less waiting time, cuts down on hospital bed hoarding and in turn, helps economy.

Rural villages that experience scarcity of water or do not have access to good quality drinkable water can use this device to test the water quality before using it. The applications of APSR can be easily extended from water testing to other body fluid testing like sweat, blood, urine or saliva as the hardware of APSR is universal whereas the software algorithm can be easily modified as per the analyte. APSR also has the potential diagnose COVID if appropriate paper strips are made available. This means that upon integration of various compatible algorithms, APSR can be an all-in-one diagnostic device and can revolutionize medical technology for many countries.

Apart from pathologies and research labs, patients or healthy can also monitor their health

Table 5.1: Verification of APSR in terms of ASSURED criteria. Extended from table 2.1.

Analyte	Analyte type	Controller	Image acquisition	Processing algorithm	ASSURED	References
Water	Water test strips	Raspberry Pi 3B+	USB webcam	OpenCV Python	A, S, U, R, E, D	Proposed system

at the comfort of their home. Health camps and POC facilities can also just as much benefir from APSR. Hospitals, even though they are mostly equipped with high-end diagnostic devices like ELISA, they can use APSR during rush hours for preliminary testing. If there is one thing we learnt from the COVID-19 pandemic is that the world cannot have enough hospitals. A huge percentage of population lost their lives owing to lack of an empty bed in the hospitals. Devices like APSR can help speed up the diagnostic process thus reducing the false-positive patients and making room for needy individuals. Since this device is household friendly, majority of the people can also avoid visiting the hospital altogether and carry out the tests at home and just send the reports to the personal clinician for review by just pressing one button on the screen. APSR is not only designed for times when people are ill, it is meant to be a part of the healthy lifestyle for frequently health tracking. This will help people stay up-to-date with their organ functionality and thus can be alarmed in advance in case of a potential medical condition.

Since these strips are made out of paper, they can be easily disposed into a dustbin without them becoming a biohazard. This eliminates the time required to follow up with lengthy bio waste disposal regulations. Even in the case of blood analysis or virus infestation, the strip can be simply burnt in a well ventilated room to avoid the potential spread of diseases. Therefore, it is safe to say that APSR satisfies A, S, U, R, E, D criteria successfully.

The dataset was a laminated colour chart which may have been overexposed under the webcam thus losing some pixel information. Using non-laminated colour chart as dataset might help improve the system accuracy even further. Paper test strips are prone to ageing owing to the nature of the paper and aging bias can be introduced as the strips get older inside the box. Older strips can get a little unevenly discoloured and are also at risk of getting brown or yellowish around the edges. This can be taken care of in the post processing of the image by adjusting the white balance of the paper strip in the image. The webcam may or may not introduce focus bias where the webcam fails to focus on the strip while clicking an image. This bias can be easily addressed as mentioned in Chapter 6.

Chapter 6

Conclusions and Future scope

The conclusions and potential advancements are discussed in this chapter.

6.1 Conclusions

We know that human eyes perceive colours differently in different light settings. Extending the colour perception routine in the field of medical technology, medical test paper strips rely solely on colorimetric analysis for the prediction of body fluid contents like sweat, urine, blood and saliva. We thus developed a colour prediction device that uses a constant intensity light source in an enclosed environment to avoid any interference from the ambient light. The software for APSR has been handcrafted for each of the 15 parameters. The proposed system is currently compatible with water testing and gave a total accuracy of 88.7%, whereas, in the same clinical study, the humans participants were 60.7% accurate. This satisfies the Sensitive criteria 'S' from ASSURED for our device. Accuracy in body fluid content prediction is of paramount importance because these test results act as a preliminary indicator of potential medical conditions. The interpolation technique proved to be an important asset in predicting the results that lied in between the manufacturer mentioned thresholds. The use of less expensive and easily available electronic components makes APSR an Affordable 'A' ($\approx$ ₹12000/ €150) and Deliverable 'D' device. The minimal and easy to understand GUI, and fast ($\approx$ 5s) operating algorithm satisfies the User-friendly 'U' and Rapid 'R' criteria, respectively. APSR is faster than the fastest system reported in the literature (2015) that took 45s to give the output using RGB profiling on an office scanner (table 2.2). The sturdy, lightweight and portable 3D printed assembly that is powerbank compatible makes APSR robust and Equipment-free 'E'. APSR can already be used in POC facilities in rural areas of underdeveloped and developing countries.

6.2 Future Scope

The applications of APSR can be easily extended from water testing to other body fluid testing like sweat, blood, urine and saliva as the hardware of APSR is universal whereas the software algorithm can be easily modified as per the analyte. Machine learning and deep learning can be used to find better fitting curves to improve accuracy of the system. The algorithm can also be further developed to make it compatible with email id input feature for the "export" function. That way, the user will be able to change the destination email id as per their will, for instance

in situation of changing your general practitioner. The APSR algorithm was 41%, 21%, 8% and 4% less accurate than humans in predicting Residual chlorine, Lead, Total alkalinity and pH, respectively. Therefore, there are 3 approaches towards further improving the accuracy: 1) the dataset images can be recaptured whilst ensuring that there is no overexposure from the LED light on the laminated colour chart; 2) use a non-laminated colour chart by printing it on a plain paper with chart dimensions; 3) the curves can be reassessed for better fittings (by using machine learning, for instance). Although the device is well within the focal length of the webcam, adding a physical delay before clicking an image can help with better focusing. Also, the device can be programmed to click three images on pressing "Start" and the image with zero variance with respect to the other two can be chosen for further image processing. This will remove the webcam focus bias, if any. Different colour spaces can be investigated for better accuracy including CMYK, HSV and LAB. LAB colour space is the most promising one. Underexposing the paper strip by reducing the LED light brightness may also help retain the pixel information in the image that can be later recovered in the post-processing.

Chapter 7

Appendix

7.1 Backend code

```
import cv2
import numpy as np
import math

def imageCropper(img):

    coords = [[72,625,142,699],
              [177,634,244,704],
              [283,637,352, 703],
              [399, 639,468, 714],
              [518, 642,589, 717],
              [638, 644,715, 721],
              [763, 646,842, 723],
              [892, 650,974, 727],
              [1024, 652,1105, 730],
              [1154, 651,1241, 731],
              [1286, 654,1362, 731],
              [1417, 659,1497, 731],
              [1546, 659,1621, 733],
              [1665, 661,1741, 737],
              [1787, 662,1856, 736],]

    scaled = []
    for c in coords:
        temp=[]
        for i in c:
            temp.append(int(i/3))
        scaled.append(temp)

    coords = scaled


    slides = []

    for coord in coords:

        cropped = img[coord[1]:coord[3],coord[0]:coord[2]]
        slides.insert(0,cropped)

    return slides


def rgbExtractor(slides):
```

```
44
45      Counter = 1
46      finalOutputs = []
47      for img in slides:
48
49          rows, cols, channels = img.shape
50          vals = []
51
52          for i in range(rows):
53              for j in range(cols):
54                  pixel = img[i, j]
55                  vals.append([pixel[2], pixel[1], pixel[0]])
56
57          npvals = np.array(vals, np.uint8)
58          npvals=npvals/255
59
60
61          scratchpad = []
62
63          if Counter==1:
64              for pixel in npvals:
65                  scratchpad.append(pixel[0]+1-pixel[1])
66
67              # finalOutputs.insert(1,[np.mean(scratchpad),np.median(scratchpad)])
68              y = np.mean(scratchpad)
69              x = (-0.00116/(2*(-1.2651*(10**-6)))) + (((0.00116**2)+(4*(-1.2651*(10**-6)
70
71
72              if x<0:
73                  x= "Out of range"
74              else:
75                  x=round(x,3)
76              finalOutputs.insert(0,x)
77              # print(finalOutputs)
78
79
80
81
82          elif Counter==2:
83              for pixel in npvals:
84                  scratchpad.append(pixel[0]+pixel[1])
85
86              # finalOutputs.insert(2,[np.mean(scratchpad),np.median(scratchpad)])
87              y = np.mean(scratchpad)
88              x = (y/1.14237)**(1/(-0.55727))
89              if x<0:
90                  x= "Out of range"
91              else:
92                  x=round(x,3)
93
94              finalOutputs.insert(1,x)
95
96
97
98          elif Counter==3:
99
100             for pixel in npvals:
101                 scratchpad.append((0.2*pixel[0])+pixel[1])
102
103             # finalOutputs.insert(3,[np.mean(scratchpad),np.median(scratchpad)])
104
105             # data['G+(0.2)*R'] = (0.2*data["Normalize-R"])+data["Normalize-G"]
106
```

```
            # for y in element:
            y = np.mean(scratchpad)
            x = (0.57494-y)/0.0103022
                # scratchpad.append(x)
            if x<0:
                x= "Out of range"
            else:
                x=round(x,3)

            finalOutputs.insert(2,x)


        elif Counter==4:
            for pixel in npvals:
                scratchpad.append(pixel[2]+pixel[1])

            # finalOutputs.insert(4,[np.mean(scratchpad),np.median(scratchpad)])
            y = np.mean(scratchpad)
            x = (y/0.59865)**(1/(-0.33806))
                # scratchpad.append(x)

            if x<0:
                x= "Out of range"
            else:
                x=round(x,3)

            finalOutputs.insert(3,x)
            # data["G+B"] = data["Normalize-G"]+data["Normalize-B"]



        elif Counter==5:
            for pixel in npvals:
                scratchpad.append(pixel[0]+pixel[1])

            # finalOutputs.insert(5,[np.mean(scratchpad),np.median(scratchpad)])
            y = np.mean(scratchpad)
            x = (y/0.31472)**(1/(-0.29873 ))
                # scratchpad.append(x)
            if x<0:
                x= "Out of range"
            else:
                x=round(x,3)

            finalOutputs.insert(4,x)

            # data["R+G"] = data["Normalize-G"]+data["Normalize-R"]


        elif Counter==6:
            for pixel in npvals:
                scratchpad.append(pixel[0]+pixel[1])

            # finalOutputs.insert(6,[np.mean(scratchpad),np.median(scratchpad)])

            y = np.mean(scratchpad)
            x = (y/2.13252)**(1/(-0.32246))
                # scratchpad.append(x)
            if x<0:
                x= "Out of range"
            else:
                x=round(x,3)
            finalOutputs.insert(5,x)
```

```
170             # data["R+G"] = data["Normalize-G"]+data["Normalize-R"]
171
172
173         elif Counter==7:
174             for pixel in npvals:
175                 scratchpad.append(pixel[0]+pixel[2])
176
177             # finalOutputs.insert(7,[np.mean(scratchpad),np.median(scratchpad)])
178             y = np.mean(scratchpad)
179             x = (1.30685-y)/0.00800599
180                 # scratchpad.append(x)
181             if x<0:
182                 x= "Out of range"
183             else:
184                 x=round(x,3)
185
186             finalOutputs.insert(6,x)
187             # data["R+B"] = data["Normalize-B"]+data["Normalize-R"]
188
189
190         elif Counter==8:
191             for pixel in npvals:
192                 scratchpad.append(pixel[0]+pixel[2])
193
194             # finalOutputs.insert(8,[np.mean(scratchpad),np.median(scratchpad)])
195             y = np.mean(scratchpad)
196             x = (1.23892-y)/0.00138032
197                 # scratchpad.append(x)
198             if x<0:
199                 x= "Out of range"
200             else:
201                 x=round(x,3)
202             finalOutputs.insert(7,x)
203             # data["R+B"] = data["Normalize-B"]+data["Normalize-R"]
204
205
206         elif Counter==9:
207
208             for pixel in npvals:
209                 scratchpad.append(1-pixel[0]+pixel[1])
210
211             # finalOutputs.insert(9,[np.mean(scratchpad),np.median(scratchpad)])
212             y = np.mean(scratchpad)
213             x = (y-0.276681)/0.00195505
214                 # scratchpad.append(x)
215
216             if x<0:
217                 x= "Out of range"
218             else:
219                 x=round(x,3)
220             finalOutputs.insert(8,x)
221             # data["(1-R)+G"] = data["Normalize-G"]+(1-data["Normalize-R"])
222
223
224         elif Counter==10:
225             for pixel in npvals:
226                 scratchpad.append(pixel[0])
227
228             y = np.mean(scratchpad)
229             x=(-(3.36811*(10**-5))/(2*(-1.45573*(10**-6)))) + ((((3.36811*(10**-5)**2)+
230
231                 # scratchpad.append(x)
232             if x<0:
```

```
233                 x= "Out of range"
234             else:
235                 x=round(x,3)
236             finalOutputs.insert(9,x)
237             # finalOutputs.insert(10,[np.mean(scratchpad),np.median(scratchpad)])
238             # data["Mean-R"] = np.mean(data["Normalize-R"])
239             # data["Median-R"] = np.median(data["Normalize-R"])
240
241
242
243         elif Counter==11:
244             for pixel in npvals:
245                 scratchpad.append(pixel[1])
246
247             # finalOutputs.insert(11,[np.mean(scratchpad),np.median(scratchpad)])
248             y = np.mean(scratchpad)
249             # x = (y-0.276681)/0.00195505
250             x=(-0.02798/(2*0.00113)) + (((0.02798**2)+(4*0.00113*y)-(4*0.00113*0.30054)
251                 # scratchpad.append(x)
252             if x<0:
253                 x= "Out of range"
254             else:
255                 x=round(x,3)
256             finalOutputs.insert(10,x)
257             # data["Mean-G"] = np.mean(data["Normalize-G"])
258             # data["Median-G"] = np.median(data["Normalize-G"])
259
260
261
262         elif Counter==12:
263             for pixel in npvals:
264                 scratchpad.append(pixel[0]+pixel[1]+pixel[2])
265
266             # finalOutputs.insert(12,[np.mean(scratchpad),np.median(scratchpad)])
267             y = np.mean(scratchpad)
268             # x = (y-0.276681)/0.00195505
269             x=(math.log((0.40762-y)/(-1.13464 ))/math.log(0.98289))
270                 # scratchpad.append(x)
271             if x<0:
272                 x= "Out of range"
273             else:
274                 x=round(x,3)
275             finalOutputs.insert(11,x)
276             # data["R+G+B"] = data["Normalize-G"] + data["Normalize-R"]+data["Normalize
277
278         elif Counter==13:
279             for pixel in npvals:
280                 scratchpad.append(pixel[0]+pixel[1])
281
282             # finalOutputs.insert(13,[np.mean(scratchpad),np.median(scratchpad)])
283
284             y = np.mean(scratchpad)
285             # x = (y-0.276681)/0.00195505
286             x = (y/21.37432)**(1/(-0.8818))
287                 # scratchpad.append(x)
288             if x<0:
289                 x= "Out of range"
290             else:
291                 x=round(x,3)
292             finalOutputs.insert(12,x)
293             # data["R+G"] = data["Normalize-G"]+data["Normalize-R"]
294
295
```

```
296         elif Counter==14:
297             for pixel in npvals:
298                 scratchpad.append(pixel[0]+pixel[1])
299 
300             # finalOutputs.insert(14,[np.mean(scratchpad),np.median(scratchpad)])
301 
302             y = np.mean(scratchpad)
303             # x = (y-0.276681)/0.00195505
304             x=(-0.04886/(2*(-0.0194)) + (((0.04886**2)+(4*(-0.0194)*y)-(4*(-0.0194)*1.4
305                 # scratchpad.append(x)
306             if x<0:
307                 x= "Out of range"
308             else:
309                 x=round(x,3)
310             finalOutputs.insert(13,x)
311             # data["R+G"] = data["Normalize-G"]+data["Normalize-R"]
312 
313 
314         elif Counter==15:
315             for pixel in npvals:
316                 scratchpad.append(pixel[1])
317 
318             # finalOutputs.insert(15,[np.mean(scratchpad),np.median(scratchpad)])
319             y = np.mean(scratchpad)
320             # x = (y-0.276681)/0.00195505
321             x = (y-0.0202479)/0.000967136
322                 # scratchpad.append(x)
323             if x<0:
324                 x= "Out of range"
325             else:
326                 x=round(x,3)
327             finalOutputs.insert(14,x)
328             # data["Mean-G"] = np.mean(data["Normalize-G"])
329             # data["Median-G"] = np.median(data["Normalize-G"])
330         Counter+=1
331     print(finalOutputs)
332     return finalOutputs
333 
334 
335 if __name__ =="__main__":
336     img = cv2.imread("/Users/anishpawar/dev/IP_New'21/IP_New21/Sample/1.10.jpg")
337 
338     slides = imageCropper(img)
339     Outputs = rgbExtractor(slides)
```

7.2 GUI code

```
1  import tkinter as tk
2  from tkinter import *
3  import cv2
4  from PIL import Image, ImageTk
5  from tkinter import filedialog, Text
6  import numpy as np
7  import smtplib, ssl
8  from email.mime.text import MIMEText
9  from email.mime.multipart import MIMEMultipart
10 
11 from Backend import imageCropper,rgbExtractor
12 
13 
14 
15 width, height = 600, 350
16 cap = cv2.VideoCapture(0)
```

```
17 cap.set(cv2.CAP_PROP_FRAME_WIDTH, width)
18 cap.set(cv2.CAP_PROP_FRAME_HEIGHT, height)
19
20 # Frame Vars
21 frame1 = []
22 clickStat = False
23 OpString1 = ""
24
25 #Tk Start
26 root = tk.Tk()
27 root.overrideredirect(True)
28 root.overrideredirect(False)
29 root.attributes('-fullscreen',True)
30
31 canvas = tk.Canvas(root,height=480,width = 800,bg="white")
32 canvas.pack()
33
34 frame = tk.Frame(root,bg="white")
35 frame.place(relwidth=1,relheight=1)
36
37 viewfinder = tk.Frame(frame,bg="white")
38 viewfinder.place(width=400,height=240,relx=0.01,rely=0.35)
39
40
41 # Hide Camera
42 #viewfinder1 = tk.Frame(frame,bg="black")
43 #viewfinder1.place(width=400,height=240,relx=0.01,rely=0.25)
44
45 # results = tk.Frame(frame,bg="#d6dba7")
46 # results.place(relwidth=0.25,relheight=0.7,relx=0.7,rely=0.25)
47
48 text_box = tk.Frame(frame,bg="white")
49 text_box.place(width=340,height=250,relx=0.555,rely=0.25)
50
51 lmain = tk.Label(viewfinder)
52 lmain.pack()
53
54 # Function Definitions
55 def show_widget(widget):
56     widget.pack()
57
58 def hide_widget(widget):
59     widget.pack_forget()
60
61 def show_frame():
62     global OpString1
63     if not clickStat:
64         global frame1
65         # frame = np.zeros((1920,1080))
66         x, frame1 = cap.read()
67
68
69         cv2image = cv2.cvtColor(frame1, cv2.COLOR_BGR2RGBA)
70         ar = cv2image.shape[0]/cv2image.shape[1]
71
72         cv2image = cv2image[int(0.17*(cv2image.shape[1])):int(0.9*(cv2image.shape[1])),
73         # cv2.imshow("A",cropped)
74         cv2image = cv2.resize(cv2image,(400,150))
75
76
77
78         img = Image.fromarray(cv2image)
79         imgtk = ImageTk.PhotoImage(image=img)
```

```
80          lmain.imgtk = imgtk
81          lmain.configure(image=imgtk)
82          lmain.after(10, show_frame)
83
84      else:
85          cv2image = cv2.cvtColor(frame1, cv2.COLOR_BGR2RGBA)
86          ar = cv2image.shape[0]/cv2image.shape[1]
87          #cv2image = cv2.resize(cv2image,(350,int(350*ar)))
88          cv2image = cv2image[int(0.17*(cv2image.shape[1])):int(0.9*(cv2image.shape[1])),
89          cv2image = cv2.resize(cv2image,(400,150))
90          img = Image.fromarray(cv2image)
91          imgtk = ImageTk.PhotoImage(image=img)
92          lmain.imgtk = imgtk
93          lmain.configure(image=imgtk)
94
95          print (frame1.shape)
96          slides = imageCropper(frame1)
97          Outputs = rgbExtractor(slides)
98
99
100         OpString = """1. Total Hardness(mg/L) = {} \n2. Bromine(mg/L) = {} \n3. Residua
101         OpString1 = """1. Total Hardness(mg/L) = {} <br>2. Bromine(mg/L) = {} <br>3. Re
102         show_txt_btn_clicked(OpString)
103
104
105
106 def dummyFunc():
107     global clickStat
108     print("I work")
109     print(clickStat)
110
111     clickStat = not(clickStat)
112
113
114     # show_txt_btn_clicked()
115     show_frame()
116
117
118
119 def show_txt_btn_clicked(text):
120     global text_box
121     scrollbar = Scrollbar(text_box)
122     scrollbar.pack(side=RIGHT, fill=Y)
123     ocr = Text(text_box, wrap=WORD, yscrollcommand=scrollbar.set,bg = 'white')
124     ocr.insert('1.0',text)
125
126     ocr.place(relx=0,rely=0,relheight = 1,relwidth = 1)
127
128     scrollbar.config(command=ocr.yview)
129
130 def imagePicker():
131     global OpString
132
133     filename= filedialog.askopenfilename(initialdir="/Users/anishpawar/Robocon/Robocon_
134     print(filename)
135
136     global og_img
137     og_img = cv2.imread(filename)
138
139     cv2.imshow('image',og_img)
140     slides = imageCropper(og_img)
141     Outputs = rgbExtractor(slides)
142     # print(Outputs)
```

```
143
144     OpString = """1. Total Hardness(mg/L) = {} \n2. Bromine(mg/L) = {} \n3. Residual Ch
145
146
147     show_txt_btn_clicked(OpString)
148
149 def mail():
150     global OpString1
151
152     sender_email = "gui.tejal@gmail.com"
153     receiver_email = "tejal.tudelft@gmail.com"
154     password = "gui.tejal2021"
155
156     message = MIMEMultipart("alternative")
157     message["Subject"] = "Water Diagnosis"
158     message["From"] = sender_email
159     message["To"] = receiver_email
160
161     text = OpString1
162     html = """\
163     <html>
164     <body>
165         <p><strong><u>Your test results:</u></strong><br>
166         {}<br>
167
168         </p>
169     </body>
170     </html>
171     """.format(OpString1)
172     part1 = MIMEText(text, "plain")
173     part2 = MIMEText(html, "html")
174
175     message.attach(part1)
176     message.attach(part2)
177
178     context = ssl.create_default_context()
179     with smtplib.SMTP_SSL("smtp.gmail.com", 465, context=context) as server:
180         server.login(sender_email, password)
181         server.sendmail(
182             sender_email, receiver_email, message.as_string()
183         )
184
185
186 # GUI
187 label = tk.Label(frame,text='Automated Paper Strip Analyzer',fg= 'black',bg='white',fon
188 label.place(relx=0.12,rely=0.0)
189
190 label = tk.Label(frame,text='Tejal Shinde',fg= 'gray',bg='white',font =('Times',14))
191 label.place(relx=0.45,rely=0.9)
192
193 label = tk.Label(frame,text='Results',fg= 'blue',bg='white',font =('Times',22))
194 label.place(relx=0.7,rely=0.15)
195
196 label = tk.Label(frame,text='Camera Feed',fg= 'blue',bg='white',font =('Times',22))
197 label.place(relx=0.2,rely=0.15)
198
199 # Buttons
200 start = tk.Button(frame,text="Start",padx=10,pady=10,fg="white",bg="green",command=dumm
201 start.place(relx=0.07,rely=0.85,height=45,width=100)
202
203 export_btn = tk.Button(frame , text = 'Export' , fg = 'white',bg = 'green', padx = 10 ,
204 export_btn.place(relx = 0.24, rely = 0.85, height=45,width=100)
205
```

```
206 # # Dummy
207 # start1 = tk.Button(frame,text="Dummy",padx=10,pady=10,fg="white",bg="green",command=i
208 # start1.place(relx=0.8,rely=0.9,height=45,width=100)
209
210 show_frame()
211 root.mainloop()
```

Bibliography

[1] Chen-zhong Li et al. "Paper based point-of-care testing disc for multiplex whole cell bacteria analysis". In: *Biosensors and Bioelectronics* 26.11 (2011), pp. 4342–4348.

[2] Woo-Ri Shin et al. "Aptamer-based paper strip sensor for detecting Vibrio fischeri". In: *ACS combinatorial science* 20.5 (2018), pp. 261–268.

[3] Supattra Arsawiset and Siriwan Teepoo. "Ready-to-use, functionalized paper test strip used with a smartphone for the simultaneous on-site detection of free chlorine, hydrogen sulfide and formaldehyde in wastewater". In: *Analytica Chimica Acta* 1118 (2020), pp. 63–72.

[4] *Test strip reader QAUNTOFX Relax.* URL: `https://www.mn-net.com/test-strip-readers-information`.

[5] *BW-300 - Urine strip reader by Bioway Biological Technology Co.,Ltd: MedicalExpo.* URL: `https://www.medicalexpo.com/prod/bioway-biological-technology-co-ltd/product-122544-936195.html`.

[6] Devon E McMahon et al. "Global resource shortages during COVID-19: Bad news for low-income countries". In: *PLoS neglected tropical diseases* 14.7 (2020), e0008412.

[7] Hagai Rossman et al. "Hospital load and increased COVID-19 related mortality in Israel". In: *Nature communications* 12.1 (2021), pp. 1–7.

[8] Vojo Djukanovic, Edward Paul Mach, World Health Organization, et al. *Alternative approaches to meeting basic health needs in developing countries: a joint UNICEF/WHO study.* World Health Organization, 1975.

[9] Pascaline Dupas. "Health behavior in developing countries". In: *Annu. Rev. Econ.* 3.1 (2011), pp. 425–449.

[10] Natasha Potgieter. *The Relevance of Hygiene to Health in Developing Countries.* IntechOpen, 2019.

[11] IndiaSpend@IndiaSpendBookmark et al. *988 Million Indians Do Not Have Life Insurance.* 2019. URL: `https://www.bloombergquint.com/business/988-million-indians-do-not-have-life-insurance`.

[12] Ho Nam Chan, Ming Jun Andrew Tan, and Hongkai Wu. "Point-of-care testing: applications of 3D printing". In: *Lab on a Chip* 17.16 (2017), pp. 2713–2739.

[13] Shankha Chakraborty, Chris Papageorgiou, and Fidel Pérez Sebastián. "Diseases, infection dynamics, and development". In: *Journal of Monetary Economics* 57.7 (2010), pp. 859–872.

[14] Hyun-Kyung Lee and Jeong-Hyeon Bae. "Design of Appropriate Technology-Assisted Urine Tester Enabling Remote and Long-Term Monitoring of Health Conditions". In: *Sustainability* 12.12 (2020), p. 5165.

[15] GS Luka et al. "Portable device for the detection of colorimetric assays". In: *Royal Society open science* 4.11 (2017), p. 171025.

[16] Arnold C Paglinawan et al. "Measurement of Specific Gravity, Urobilinogen, Blood, Protein and pH Level of Urine Samples Using Raspberry Pi based Portable Urine Test Strip Analyzer". In: *Proceedings of the 2020 10th International Conference on Biomedical Engineering and Technology*. 2020, pp. 58–63.

[17] Nuria López-Ruiz et al. "Computer Vision-Based Portable System for Nitroaromatics Discrimination". In: *Journal of Sensors* 2016 (2016).

[18] Saurabh Khatiwada, Kelly M Elkins, et al. "CSI-Pi: A novel automated secure solution to interpret on-site colorimetric tests". In: *Colonial Academic Alliance Undergraduate Research Journal* 5.1 (2015), p. 2.

[19] Katherine E Boehle et al. "Single board computing system for automated colorimetric analysis on low-cost analytical devices". In: *Analytical Methods* 10.44 (2018), pp. 5282–5290.

[20] George Papadakis et al. "Real-time colorimetric LAMP methodology for quantitative nucleic acids detection at the point-of-care". In: *bioRxiv* (2020).

[21] Uddin M Jalal, Gyeong Jun Jin, and Joon S Shim. "Paper-plastic hybrid microfluidic device for
smartphone-based colorimetric analysis of urine". In: *Analytical chemistry* 89.24 (2017), pp. 13160–13166.

[22] Adrian Carrio et al. "Automated low-cost
smartphone-based lateral flow saliva test reader for drugs-of-abuse detection". In: *Sensors* 15.11 (2015), pp. 29569–29593.

[23] Hyunwook Jang, Syed Rahin Ahmed, and Suresh Neethirajan. "GryphSens: A smartphone-based
portable diagnostic reader for the rapid detection of progesterone in milk". In: *Sensors* 17.5 (2017), p. 1079.

[24] Hojat Heidari-Bafroui et al. "Portable infrared lightbox for improving the detection limits of paper-based phosphate devices". In: *Measurement* 173 (2021), p. 108607.

[25] Robin E Sweeney et al. "Flow rate and raspberry PI-based paper microfluidic blood coagulation assay device". In: *IEEE sensors journal* 19.13 (2019), pp. 4743–4751.

[26] Anuradha Soni and Sandeep Kumar Jha. "A paper strip based non-invasive glucose biosensor for salivary analysis". In: *Biosensors and Bioelectronics* 67 (2015), pp. 763–768.

[27] Ying Wang et al. "Quantification of combined color and shade changes in colorimetry and image analysis: water pH measurement as an example". In: *Analytical methods* 10.25 (2018), pp. 3059–3065.

[28] Hiroko Kudo et al. "Microfluidic paper-based analytical devices for colorimetric detection of lactoferrin". In: *SLAS TECHNOLOGY: Translating Life Sciences Innovation* 25.1 (2020), pp. 47–57.

[29] Marzia Hoque Tania et al. "Intelligent image-based colourimetric tests using machine learning framework for lateral flow assays". In: *Expert Systems with Applications* 139 (2020), p. 112843.

[30] Kai-Wen Lin and Yu-Chi Chang. "Embedded Immunodetection System for Fecal Occult Blood". In: *Biosensors* 11.4 (2021), p. 106.

[31] *15 in 1 Drinking Water Test Kit Strips - Water Quality Test - Well Water and Tap Water - Low Level ranges for Lead, Fluoride, Iron, Copper & Mercury : Amazon.in: Garden & Outdoors*. https://www.amazon.in/Drinking-Water-Test-Kit-Strips/dp/B086Q3JNZX. (Accessed on 09/08/2021).

[32] Kyriaki Glynou et al. "Oligonucleotide-functionalized gold nanoparticles as probes in a dry-reagent strip biosensor for DNA analysis by hybridization". In: *Analytical Chemistry* 75.16 (2003), pp. 4155–4160.

[33] Ajit P Gosavi et al. "IoT Based pH Reader". In: ().

[34] *Urine Strip Reader 40*. URL: https://www.dialab.at/en/products/instruments/urine-strip-analysers/urine-strip-reader-40/.

[35] All. *Urine Analyzer U120*. URL: https://microsidd.com/en/mission-urine-analyzer\?gclid=CjwKCAjwiLGGBhAqEiwAgq3q_in5A3m4oyCSDfKNic\\9Xdj9vfDu1ljK94H3DacpWKEkkQye2_WSKBBoC\\SGIQAvD_BwE.