FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

# Cross-platform application for determination of sulfonamides in water using digital image colorimetry

**Fábio Alexandre Matos Azevedo**


Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Hélder Filipe Pinto Oliveira

Second Supervisor: Pedro Henrique Moreira Queirós Carvalho

July 19, 2021

# Cross-platform application for determination of sulfonamides in water using digital image colorimetry

**Fábio Alexandre Matos Azevedo**

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Prof. Luís Teixeira
External Examiner: Prof. Paula Viana
Supervisor: Prof. Hélder Oliveira

July 19, 2021

# Abstract

Currently, the usage of antibiotics in the treatment of humans and animals is recurrent and with an increasing trend. One of its many side effects is that the presence of these compounds in aquatic environments is increasing, causing the emergence of antibiotic-resistant bacteria, which will make intractable infectious diseases more likely to appear.

Sulfonamides are an important group of antibiotics that are frequently found in environmental water. Their determination is vital since they kill the non antibiotic-resistant bacteria, allowing the others to proliferate. Furthermore, these compounds seem to be quite resistant to biodegradation in surface water which can benefit contamination of aquatic environment.

One of the most common methods to analyse and determine the concentration of sulfonamides in water consists in high-performance liquid chromatography coupled with mass spectrometry (HPLC-MS/MS), which is an expensive, reagent consuming process and not suitable for an in situ analysis strategy. When sulfonamides are mixed with the colorimetric reagent *p-dimethylaminocinnamaldehyde* a color is obtained. From that color is possible to extract the concentration of the compound in the sample.

As we may need to carry out an analysis at locations where the nearest laboratory is miles away then on the field analysis are essential to act quickly so the solution must be fully mobile. We cannot deny that smartphones are powerful, in fact, some of them are even more powerful than old and new generation laptops. With this in mind, the solution we are working on consists of a mobile application that leverages the computing power and ease of use of these devices. The proposed application must be able to let the user take a new photo of the solution or load an existing one. It should also save the taken photos in an ordered way, for posterior analysis. After that the images are going to pass through a color correction algorithm returning the estimated concentration of the sample. This approach will enable the ability of in situ analysis, without requiring an Internet connection or specified heavy equipment.

Within the context where this dissertation is inserted, there is already an application that aims to solve the problem in question, but it is exclusive to Android which nowadays is a big turnoff since iOS devices are very popular and used. The proposed solution intends to be transversal to the OS mentioned above and to be able to use manual input to help the algorithm in a friendlier way when compared to the original.

**Keywords**: Application, Android, iOS, Mobile, Color, Bacteria, Colorimetry, Sulfonamides, Water.

**Areas**: CCS → Computing Methodologies → Artificial Intelligence → Computer Vision

# Resumo

Atualmente, o uso de antibióticos no tratamento de humanos e animais é uma prática recorrente e em crescimento. Um dos seus muitos efeitos colaterais é o facto de que a presença destes compostos em ambientes aquáticos está a crescer, causando o surgimento de bactérias resistentes a antibióticos, o que torna doenças infeciosas intratáveis mais propícias a aparecer.

As Sulfonamidas são um grupo importante de antibióticos frequentemente encontrados em águas no meio ambiente. A sua determinação é vital, uma vez que eles matam as bactérias não resistentes a antibióticos, permitindo que as outras proliferem. A resistência destes compostos à biodegradação em águas é um fator que contribui para a sua contaminação.

Um dos métodos mais comuns para analisar e determinar a concentração de sulfonamidas em água consiste em cromatografia líquida de alto desempenho associada a espectrometria de massa (HPLC-MS / MS), que é um processo caro, que consome reagente e não é adequado para uma aplicação in situ. Quando as sulfonamidas são misturadas com o reagente colorimétrico *p-dimethylaminocinnamaldehyde*, uma cor é obtida. Dessa cor é possível extrair a concentração do composto na amostra.

As análises no terreno tornam-se essencias para uma resposta rápida perante a necessidade de realizar uma análise num local onde o laboratório mais próximo fique a quilómetros de distância pelo que a solução deve ser totalmente móvel. Não pode ser negado que os smartphones são poderosos, na verdade, alguns deles são ainda mais poderosos do que muitos computadores de última geração. Com isso em mente, a solução proposta consiste numa aplicação móvel que aproveite o poder de computação e a facilidade de uso destes dispositivos. A aplicação proposta deve permitir que o utilizador tire uma foto da solução ou carregue uma existente. Devem também ser guardadas as fotos tiradas de forma ordenada, para posterior análise. Em seguida as imagens passarão por um algoritmo de correção de cores retornando a concentração estimada da amostra. Esta abordagem permitirá a capacidade de análise in situ, sem a necessidade de uma conexão com a Internet ou equipamento pesado especifico.

Dentro do contexto onde esta dissertação se encontra já existe uma solução que procura dar resposta ao problema em questão contudo é exclusiva para Android, o que hoje em dia é uma grande barreira de acessibilidade, uma vez que os dispositivos iOS são muito populares e usados. A solução proposta visa ser transversal a ambos os sistemas operativos mencionado acima, permitindo também que o utilizador auxilie de forma manual o algoritmo.

**Keywords**: Aplicação, Android, iOS, Movél, Cor, Bactéria, Colorimetria, Sulfonamidas, Água.

**Àreas**: CCS → Metodologias de Computação → Inteligência Artificial → Visão de Computador

# Acknowledgements

First, this work was financed by the ERDF - European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e Tecnologia within project POCI-01-0145-FEDER-031756. A special thank you to all collaborators of this project, namely to Professor Hélder Oliveira and researchers Patrícia Peixoto and Henrique Carvalho.

I am grateful to have had my family by my side on this journey, without them I would have finished my dissertation much less sane. A big thank you to my mother, father, sister, uncle and godmother for all their support and encouragement.

My second family, also known as friends, played a very important role in my life these past few months, putting up with me and supporting me in the daily Discord calls

I've been blessed with a fantastic group of people and I'm so grateful for that.

Fábio Azevedo

*"It always seems impossible*
*until it's done."*

Nelson Mandela

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| AI | Artificial Intelligence |
| CPU | Central Process Unit |
| DNA | Deoxyribonucleic acid |
| GPU | Graphics Processing Unit |
| HDR | High Dynamic Range |
| HLPC-MS | High-performance Liquid Chromatograpy tandem Mass Spectometry |
| iOS | Apple Operative System |
| Kg | Kilogram |
| ML | Machine Learning |
| MP | Mega Pixels |
| OS | Operative System |
| RGB | Red Green and Blue |
| SAs | Sulfonamides |
| SMS | Short Message Service |
| UI | User Interface |
| WHO | World Health Organization |

# Chapter 1

# Introduction

Overuse of antibiotics for treatment of humans and farm animals is causing the environment to become polluted [4]. Antibiotic resistance is becoming more common, which is a serious threat to the health of everyone in the world. The treatment of infections becomes troublesome when the microorganisms responsible are resistant to antibiotics, with raise in treatment time as well as risking the spread of the infection and, ultimately, death [25]. Sulfonamides are an important group of antibiotics that are often found in environmental water. Wastewater discharges, contaminated manure and aquaculture are some of the ways through which these coumpounds get into the environment mentioned above. Since they kill the non antibiotic-resistant bacteria and allows the others to proliferate, their determination is vital.

Antibiotic resistance can affect any person, at any stage of life, but people receiving health care or those with weakened immune systems are often at higher risk of getting an infection [11]. By 2019 it was possible to estimate that, on USA, each year at least 2.8 million people get infected and more than 35.000 dies as a result of the infection. New forms of resistance emerge and can spread with remarkable speed between continents through people, goods, and animals [11]. This threat continues to grow both in seriousness and extension and is the reason why monitoring programs for the determination of certain compounds, such as Sulfonamides, must be continuosly developed and improved.

When sulfonamides are mixed with the colorimetric reagent *p-dimethylaminocinnamaldehyde* a color is obtained. From that color is possible to extract the concentration of the compound in the sample. Therefore, digital image colorimetry is a key method for sulfonamides detection on water samples.

## 1.1 Motivation

In the last few years great advancements were made in the field of digital image colorimetry. In a similar way smartphones have evolved and are now similar in processing power to a laptop.

Having both these tools molded together, it is possible to produce new solutions that target the issue mentioned above as well as improve existing ones. These new solutions will be cheaper and easier to use when compared to the traditional ones.

Currently there is a fully mobile solution [9] capable of quantifying the concentration of sulfonamides on water environments. However, this solution is attached to a singular OS (Android) which hinders other mobile users, such as iOS, to utilize the solution. To summarise, the approach is incomplete and not available to the majority of the potential users.

## 1.2 Objectives

Regarding this research there are several key objectives:

- **Multi OS support:** There are currently 3.5 billion smartphone users worldwide. Considering the total number of people using phones globally is at 4.8 billion, that means that nearly 73 percent of them are users of smartphones [19]. By October 2020, Android was in the lead position for the most used mobile operating system with 72.92 percent share. Apple's operating system, iOS, had 26.53 percent share. Google's Android and Apple's iOS jointly possess almost 99 percent of the global market share [23], which justifies why it is so important that the application runs on both OS.

- **Local implementation of the algorithm:** The machine learning algorithms that will be applied on this solution can only be executed on a computer and as we aim to use this solution in places where there may be no internet connection of any kind it is not ideal to be external server dependent, so with this in mind, the algorithm should be translated and ported in a way that it can be executed in the mobile phone.

- **User input to assist the algorithm:** In case of failure of the automatic version it is important that the touch capabilities of smartphones are used to let the user manually assist the algorithm on specific sections such as cropping the image to just contain the sample. The color palette is an element present in all images that the algorithm processes and it is extremely important that the identification of patches is done correctly, manually selecting a specific group of patches from the color palette is also crucial since it helps to improve the algorithm accuracy. Figure 2.2 represents an example of a image that is sent to the algorithm for analysis, color palette is present as stated before.


Figure 1.1: Example Image to be segmented, containing the color palette and the sample [4].

## 1.3 Contributions

The solution proposed on this thesis consists in a mobile application capable of screening concentrations of Sulfonamides in water after analyzing the color of the sample with an automatic image processing algorithm. That said, in the end of the dissertation we have:

- A fully functional mobile application compatible with both Android and iOS devices;
- The algorithms done by Inês Rocha [29] running on Android and iOS.
- An interface capable of letting the user assist the analysis process;
- An option for the user to select a specific group of patches among the 24 of the color chart. This option was developed for purposes of optimization and it will be, later on this dissertation, explained how.

## 1.4 Document Structure

The monography structure will be as follows:

- **Chapter 2 - State of the Art:** Here the current state of sulfonamides detection will be analysed in-depth. Previous solution will be in-depth analysed in this chapter as well. There are three section: background (exploring sulfonamides), exploring mobile phones and previous solution analysis;
- **Chapter 3 - Methodology:** Here will be described the developed solution, the main architectural decisions and described the information flow;
- **Chapter 4 - Results:** Here will be detailed every screen developed and justified the design choices and usability;

- **Chapter 5 - Conclusion:** Conclusion of the dissertation and reflection about future work to be done.

# Chapter 2

# Literature Review

In this chapter we explore the background associated with colorimetry and the detection of sulfonamides in aquatic environments, as well as, the history of mobiles phones and their cameras and exemplifying how they can be useful in medicine and, more specifically, in SAs detection using digital image colorimetry. An example of a fully functional colorimetry algorithm that serves as one entry point for this dissertation is also detailed and explained.

## 2.1 Background

### 2.1.1 Sulfonamides

Since their appearance, in 1968, sulfonamides positioned themselves among the most widely used antibiotics in the world. One of the main uses of sulfonamides in medicine is for respiratory and urinary tract infections [6]. This compound has high effectiveness against bacteria which, in conjunction with being inexpensive and non toxic to the patients, justifies its widespread consumption. Unfortunately, the appeal of the consumption of antibiotics is leading to a clearly negative phenomenon called overprescription. Using these drugs on treatment of livestocks also contributes to increase the appearance of sulfonamides in environmental water.

Sulfonamides kill the non antibiotic-resistant bacteria allowing the others to proliferate which makes their determination vital.

#### 2.1.1.1 Detection Methods

The already existing methods for measure of the compounds concentrations can be divided into three groups: quantitative, screening and confirmatory methods.

Quantitative methods are the hardest ones to execute. They demand heavy laboratory equipment and high trained professionals as well as complex sample-preparation procedures [31], which makes them expensive and extremely time-consuming. They are mostly based on chromatography and electrokinetic separation methods.

Low cost and percentage of false complaints, as well as ease of use and short analysis time are some of the advantages of screening methods. Allowing for the reliable checking of samples is one the main objectives of these kind of methods, therefore only samples that indicate the presence of the analyte should be selected. These methods are able to detect chemical substances providing a semi-quantitative result.

Although they have the capability of reliably identify a compound, confirmatory methods are expensive and have a lot of complexity involved and should only be used on two conditions:

1. Precise measurements are required;

2. A screening process has been done before.

Among these three groups here are some of the mainly used methods for determinaton of SAs concentrations:

- **High-performance liquid choromatography tandem mass spectometry (HLPC-MS):** This method can be divided in two distinct stages. The first one consists in separating the individual components with liquid chromatograpy, after this, in the second stage, mass spectometry is used to analyse the mass structure of each element [12]. HPLC-MS is nothing more than combine liquid choromatograpy with mass spectometry. This methos is not only used for the detection os SAs, but also with other pluri-molecular organic compounds.

- **Immunoassay:** Having in mind that antibodies are capable of binding to a particular target structure, Li et al. [18], created an antibody to bind the structure of sulfonamides. Back in 2000, Muldon et al. [22] made possible the detection of eight sulfonamides by using an antibody created by them. Four years later, in 2004, Korpimäki et al. [16] produced a specific antibody with the capability of detecting sulfonamides under different levels of concentrations.

- **Capillary electrophoresis:** HLPC-MS method can be affected by low efficiency during the separation process, which is crucial for the analysis [15]. An alternative to the liquid chromatography separation is the capillar electrophoresis, which consists in separating the components according to their size and charge. A capillary is a small tube where the injector travels until reaching the destination vial.

- **Gas chromatography:** Another method subdivided in two phases. Initially the compounds are separated using gas chromatography, after which atomic emission is used to determine the sulfonamides. This method was developed by Chiavarino et al. [5].

In Figure 2.1 it is possible to see the percentages corresponding to the usage of some methods. HPLC-MS(/MS) occupies the lead position being the most used analytical method.

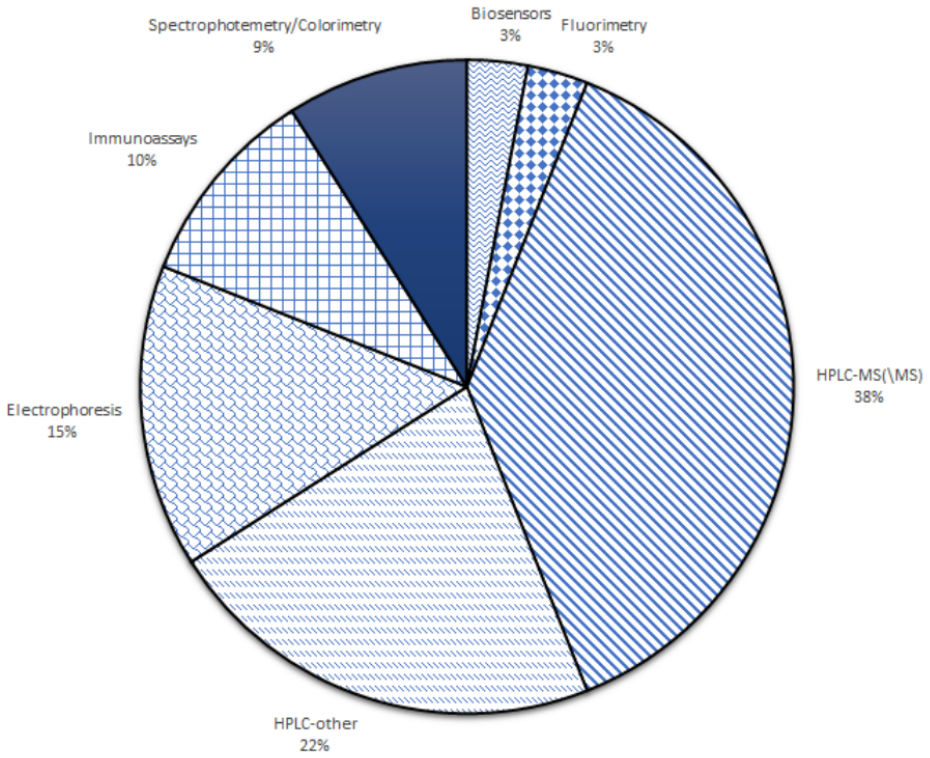


Figure 2.1: Methods used to determination of sulfonamides [8].

### 2.1.2 Colorimetry

The art of physically quantify the human color perception is called colorimetry [24]. Colorimetry positions itself only on the visible region of the electromagnetic spectrum as is somewhat similar to spectrophotometry [9]. Colorimetry can only be performed on devices capable of capturing the wavelengths of visible light and is uses the following tools:

- **Colorimeter:** The colorimeter is an instrument tipically used to ascertain the concentration of a solution based on the amount of absorbed light of a specific wavelength [30].
- **Spectrophotometer:** Capture and evalute color are the main capabilitites of this device [39].

As this devices require technical handling they are out of reach to the average user. However, colorimetry is already being used in devices that the great majority of us have: smartphones. The topic of colorimetry on smartphones (mobile phones as medical devices) will be adressed more deeply in the next section.

## 2.2 Algorithm to detect sulfonamides in water

Colorimetry and chemical analysis are two concepts that work together adequately and we will demonstrate that in this subsection. In 2019, Carvalho et al. [4] developed an algorithm with the

purpose of estimating sulfonamides concentration in a more inexpensive and usable way without requiring to send a water sample to a laboratory to be analyzed. Their main motivation was the fact that traditional methodologies were somewhat impractical and too much expensive, therefore digital image colorimetry was the chosen approach to extract the result from a sample.

This new approach only required a photo of the sample taken from a smartphone camera and a computer to run the algorithm. Despite there was not the need of laboratory equipment, the team had the need to use a specific reagent called *p-dimethylaminocinnamaldehyde* to give color to the sample. From that, the process subdivides itself in three major sequential steps:

1. **Photographs database:** Each photograph consists in two regions of interest - the sample and the color chart - set up as shown in Figure 2.2. The database is made of 24 photographs for each sulfonamide concentration as a result of being taken by three different mobile phones. This was made so that they could determine the impact that the device and camera variations can have on the results. To guarantee more accuracy, the samples were made in a laboratory and the SAs concentration was known *a priori*.

Figure 2.2: Example database photograph containing the color palette and the sample. [4].

2. **Segmentation:** Both the sample and the color chart need to be segmented. In order to do that, a segmentation algorithm is applied and both regions of interest are isolated, starting with the color chart, followed by the 24 patches and then the disc with the sample. The process is as follows:

   (a) **Chart:** The color chart used on this experiment consists in 24 patches with a black frame and, before trying to segment the patches, it is necessary to locate the chart on its own. Starting by converting the image to grey scale and using a multilevel Otsu thresholding [4] to generate 3 thresholds, the color chart is detected. Afterwards, the detection the image is split in two and only the side with the patches is selected. Figure 2.3 shows the chart segmentation process.

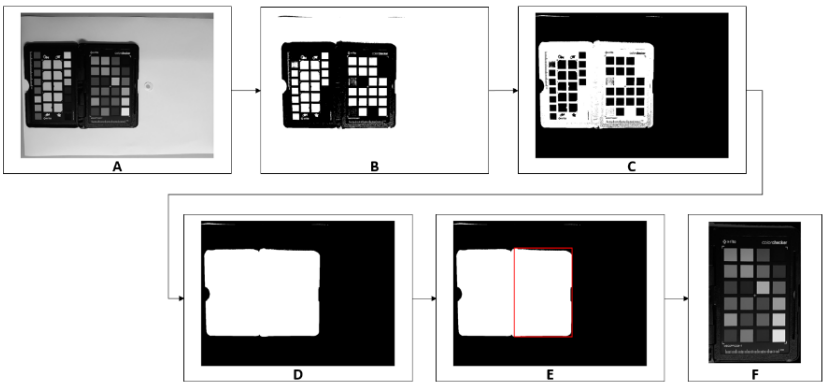


Figure 2.3: Color chart segmentation process. A: Grayscale convertion of original image. B: Result of A binarisation. C: B's complement. D: Hole filling process result. E: Bounding box around target. F: Color chart grayscaled. [4]

(b) **Patches:** At the end of the previous process, there should be an image containing only the side of the chart with the 24 patches. The image is binarised and a new Otsu threshold [4] is applied so all the 24 patches can be isolated. One thing that should have been kept in mind is that, it is an hard task for a threshold to binarise all the patches since they encompass a large range of colors. In order to fix this, the missing spots are completed using the centroids of the detected patches (we can observe this process on D in Figure 2.4).

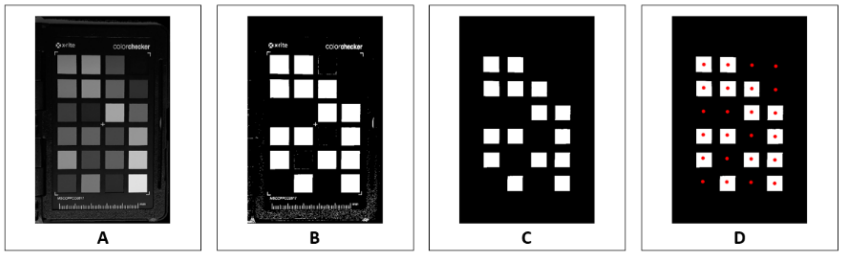


Figure 2.4: Patches segmentation process to estimate the 24 patches centroids. A: Grayscale color chart. B: A's binarisation. C: Isolated patches. D: Centroids estimation. [4]

One last thing needs to be done before the process can be completed. It is important to mitigate the existence of outlier pixels and, in order to do that, a bounding box is added to each patch, with its center being the identified point. After that, the median value is extracted. Figure 2.5 demonstrates the final result of the described process.

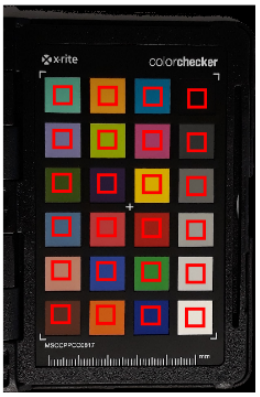


Figure 2.5: Bounding boxes algorithm result [4].

(c) **Disk:** The disk is detected by using a Canny edge (image B in Figure 2.6), to ensure the connection between the edges of the disk a dilation is applied. The last step consists on performing a filling in order to cover the middle of the disk (image C in Figure 2.6). On Figure 2.6 is shown the initial edge detection. dilation and fill results and disk crop.

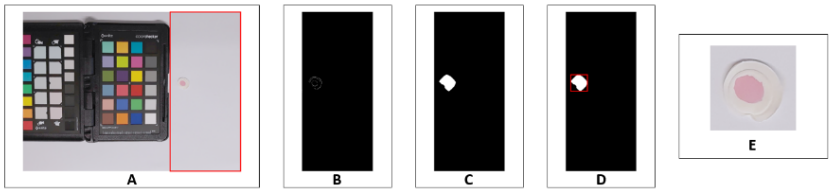


Figure 2.6: Disk detection process. A: Disk region detected (witouth the color chart). B: Detection of the edge. C: Edges dilated and filled. D: Bounding box. E: Cropped disk. [4]

(d) **Sample:** To get a more precise detection of the disk, the circular Hough transform is applied (image A in Figure 2.7). For relatively big concentrations of sulfonamides, it is easy to distinguish the color from the rest of the disk, however, when we talk about small concentrations it can be quite troublesome. The solution found is to convert the image to grayscale and normalize it between 0 and 1. Afterwards, a histogram equalisation (image D in Figure 2.7) followed by a binarisation is done (image E in Figure 2.7).


Figure 2.7: Color extraction steps. A: Hough transform applied for disk detection. B: Cropped Disk. C: Grayscale B. D: Result of C normalisation and histogram equalisation. E: Binarisation of D. F: Sample after detection. [4]

3. **Color correction:** With the color path information obtained, one last step was required: a method for color correction needed to be chosen. With that in mind, the following color correction methods were compared by Carvalho et. al [4].

   - Weighted Grey Edge (WGE);
   - Illuminant Estimation Using White Patch (White);
   - Illuminant Estimation Using Achromatic Patches (Neutral);
   - Color Correction Matrix RGB to RGB;
   - Color Correction Matrix RGB to XYZ.

   There were used images of four different concentrations and in order to guarantee that the comparison is independent of the segmentation process, the sample was performed and the patches were annotated manually.

   Figure 2.8 show the comparison results between the methods, using as metric the standard deviation.

| Colour Component | WGE | White | Neutral | RGB-RGB | RGB-XYZ |
|---|---|---|---|---|---|
| R | 0.058 | 0.055 | 0.054 | 0.040 | **0.016** |
| G | 0.067 | 0.068 | 0.068 | 0.039 | **0.021** |
| B | 0.056 | 0.057 | 0.057 | 0.047 | **0.023** |
| L | 0.057 | 0.056 | 0.056 | 0.034 | **0.017** |
| a* | 0.022 | 0.027 | 0.027 | 0.024 | **0.018** |
| b* | 0.021 | 0.024 | 0.023 | 0.025 | **0.017** |
| X | 0.077 | 0.076 | 0.075 | 0.072 | **0.025** |
| Y | 0.080 | 0.081 | 0.081 | 0.072 | **0.027** |
| Z | 0.069 | 0.072 | 0.072 | 0.081 | **0.034** |
| H | **0.011** | 0.013 | 0.013 | 0.013 | 0.013 |
| S | 0.039 | 0.043 | 0.044 | 0.032 | **0.026** |
| V | 0.058 | 0.055 | 0.054 | 0.040 | **0.016** |
| Average | 0.051 | 0.052 | 0.052 | 0.043 | **0.021** |

Figure 2.8: Comparison between color correction methods [4].

The author concluded that the best results are obtained, when the color check is mapped from the RGB color space to the XYZ color space.

## 2.3 Mobile Phones

Nowadays there is something that we humans cannot live without: the mobile phone. Phones started with the single purpose of making phone calls, but as the time passed more features were

added and now is used for almost everything. The great majority of people could not live without their mobile phones, but, in fact mobile phones as we know them today have only been around in the last 20 years [37].

A quick look to the numbers show us that there are 4.8 billion people using a phone and 3.5 billion of them are smartphone users [19]. Almost 62 percent of people in the world uses a phone and 73 percent of them have a smart one. In terms of OS (Operative System), by October 2020 the numbers told us that Android was leading with 72.92 percent share against the 26.53 percent share of iOS [23]. These two OSs combined make up about 99 percent of the global market share. Figure 2.9 represents the evolution of the mobile phone over time.


Figure 2.9: Evolution of the Mobile Phone [38].

### 2.3.1 First mobile phone

Our inseparable companions, usually known by smartphones, are relatively new. However, the first US Patent for a wireless telephone was issued back at 1908, in Kentucky.

Mobile phones appeared for the first time a little later, in the 1940s by the hands of engineers working at AT&T developing cells for mobile phone base stations [37]. However, these phones were not mobile at all since they were nothing more than two-way radios that are used by services such as emergency ones to communicate. The mobile phone network at that time consisted in a single powerful base station that covered a very wide area [37]. Nowadays there are multiple base stations with separated cells that communicate with each other.

On 3 April 1973, Motorola started producing in mass the first handheld mobile phone, a device that weighted 1.1kg and is know as the 0G (Zero Generation) mobile phone (see Figure 2.10). This prototype had a 10 hours recharge time and a talk time of about 30 minutes. The first mobile telephone call from an handheld equipment was made by Dr. Martin Cooper, an engineer at Motorola, who called his old rival Bell Labs [38].

Figure 2.10: DynaTac the first mobile phone, developed by Motorola in 1973 [14].

### 2.3.2 Landmarks in mobile history

During the last 50 years there were a lot of events that we can consider as landmarks of mobile history.

- **1973:** Using a 1.1Kg device, Dr. Martin Cooper from Motorola made the first mobile phone call ever.
- **1985:** This year the first "public" phone call was made. It was in UK and was made by the comedian Ernie Wise.
- **1992:** A developer for a telecom contractor called Neil Papworth, aged 22, sent the world's first ever SMS message.
- **1998:** The arrival of the rigntone, the first downloadable contet sold to mobile phones, launched by Finland's Radiolinja.
- **1999:** Emoticons were finally replaced for pictures, the so known and used emojis. They were invented in Japan by Shigetaka Kurita. This year was also the year when Blackberry released their first phone.
- **2000:** The famous Nokia 3310 was released and 126 million units were sold.
- **2003:** The era of mobile internet arrived as the 3G standard started to be adopted. Smartphone were a step closer.
- **2007:** Arrival of the first iPhone from Apple.
- **2008:** The first Android phone was released. Android Market (later renamed to Google Play Store) and Apple's app store debuted starting a industry evaluated in 77 billion dollars.

### 2.3.3 Mobile phones cameras

Nowadays smartphones are very different from what they were back in 1973 or even 2000 if we compare with Nokia 3310. They are smaller, lighter, fastest and obviously better.

We currently have professional dual-lens cameras on the majority of smartphones as well as all-screen displays or face-scanning technology that let us unlock our phone just by looking at it.

The first appearance of a mobile camera dates back to 2000 when J-Phone was released with 0.11 megapixels camera. This settled a precedent for the smartphone industry. 6 years passed and we reached 2006 the year were Nokia N93 arrives at the market with an incredible 3.15 megapixels camera and the ability to flip over into a camcorder style format [3].

The year of 2009 was great for the mobile cameras enthusiasts. In this year, Sony released Sony Erickson S006 with a 16MP camera as a result of a race between the tech giants for megapixels. As techonology evolved, more of it was added to cameras: HDR, panoramas, time-lapse, slow-motion and even AI (Artificial Intelligence) to improve a photograph.

Since the color accuracy is critical for the final result of colorimetry we may have a big problem with the new features that are being added to mobile cameras, as two similar hardware-level cameras can produce two very distinct images.

## 2.4 Mobile phones as biomedical devices

### 2.4.1 Sugar Application

Medical devices are equipment's that are traditionally used in the diagnosis of a disease or medical condition [1]. Medical devices, such as spirometers and X-Ray machines, are expensive and demand a specialized usage so they can only be used in clinical environments, such as hospitals, and by trained clinicians [1]. With such limitations it is more than obvious that the waiting time for patients will be long which can be extremely negative for their health.

Calling and texting was the main purpose of a smartphone but as time goes by more features started to appear and some of them could be used to help diagnose certain medical conditions. The microphone is one of them, the user's audio signal that is captured can be processed to detect cough and allergies [17]. Another example are the high-resolution cameras that are now standard on smartphones, videos recorded by these cameras can be analysed to estimate the user's heart rate [28].

Medicine is a field that could be greatly improved by the use of the previous mentioned smartphone features. By 2019 there were 318,000 mobile health apps available and over 60 percent of people have downloaded them [21].

One of the most relevant features of a smartphone for this dissertation is the colorimetry available in smartphone cameras. One good example of a smartphone camera being used on a medicinal context is *Sugar application* developed by Agu et al. [1] for the analysis of diabetes wound healing. Approximately 7.8 percent of Americans have diabetes and about 5 millions have wounds that requires them to go to the hospital frequently. Sugar was developed with the aim of letting the users track blood glucose level as well as other physical variables such as weight, physical activity and nutrition. Moreover, this application have an extra module associated: a module for the users to assess their diabetic wounds healing status [2]. To achieve this a series of processing steps are applied to pictures of the patient's wounds taken by their smartphones camera. From the relative

size of red, yellow and black tissues within the wound is possible to infer the wound healing status and how well it is healing. The execution flow is simple:

1. A photo of the wound is taken by the patient (see Figure 2.11);
2. The picture is decompressed;
3. The image segmentation algorithm is applied in order to detect wound areas boundaries;
4. Color segmentation algorithm is applied to demarcate red, yellow and black regions of the wound (see Figure 2.12).

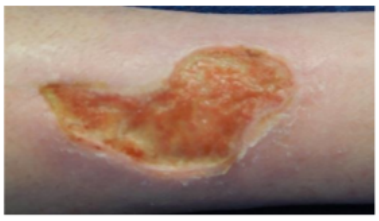

Figure 2.11: Original Wound Image [2].

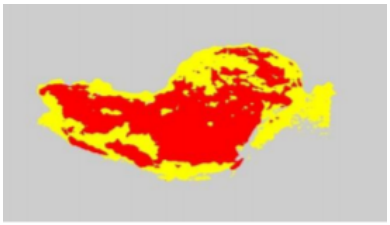

Figure 2.12: color separated wound image [2].

### 2.4.2 Android application for determination of sulfonamides concentration using digital image colorimetry

Traditional methods for quantifying sulfonamides in water require the researcher to collect a water sample in the location and send it to a laboratory to be analyse; ergo, these methods can be quite expensive and lack mobility as well as being remarkably time consuming for the researchers.

In an attempt to overcome these difficulties, a mobile application for determination of sulfonamides concentration on water based on digital image colorimetry was developed by Pedro Reis [9], in 2019. This application is fully functional, however, for this dissertation's context, it lacks features in order to be a more robust application. Moreover, this original implementation's execution is limited to Google's Android, which impedes other mobile users from utilizing it.

The following features are covered by Reis' solution [9]:

- **Photo selection:** The application allows the user to take a new photo of the sample or load one from the smartphone gallery. The photo should contain the color palette and the disk with the sample. After the photo is taken or selected, it is possible to associate a location to it. Figure 2.13 shows the sample menu screen after the photo selection.

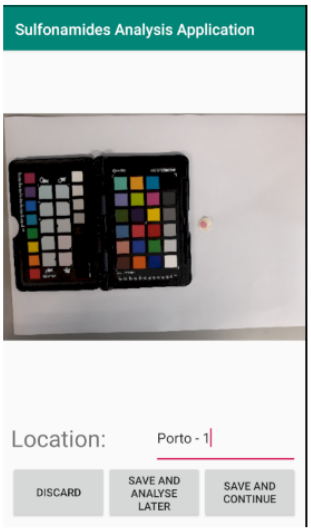


Figure 2.13: Sample Menu where the user can add a location to a photo and analyse it [9].

- **Manual Input to assist the algorithm:** The developed solution provides the ways for the detection of the color checker, patches, disk and sample. Everything can be done automatically (the most conventional way), however for each step of the algorithm the user has the possibility of manually adjust the result of the automatic mode:

  - **Checker Detection:** Figure 2.14 and 2.15 shows the automatic and manual detection of the color checker, respectively.

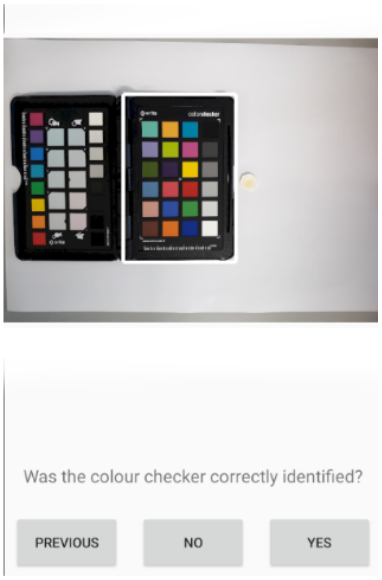


Figure 2.14: Automatic Checker Detection [9].

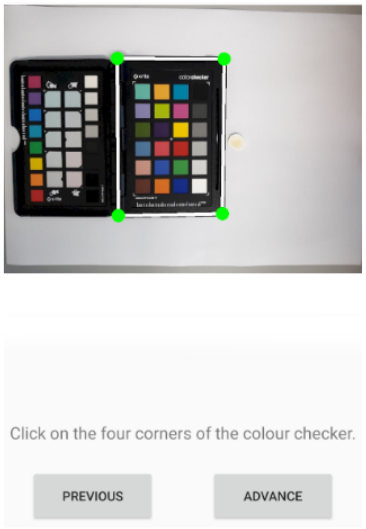


Figure 2.15: Manual Checker Detection [9].

– **Patches Detection:** Figure 2.16 and 2.17 shows the automatic and manual detection of the patches, respectively.


Figure 2.16: Automatic Patches Detection [9].


Figure 2.17: Manual Patches Detection [9].

– **Disk Detection:** Figure 2.18 and 2.19 shows the automatic and manual detection of the disk, respectively.

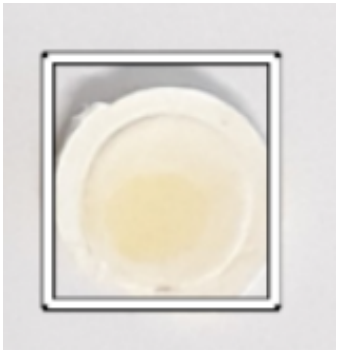

Figure 2.18: Automatic Disk Detection [9].

Figure 2.19: Manual Disk Detection [9].

– **Sample Detection:** Figure 2.20 and 2.21 shows the automatic and manual detection of the sample, respectively.

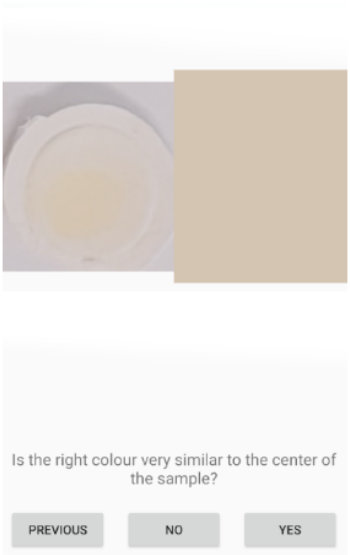


Figure 2.20: Automatic Sample Detection [9].

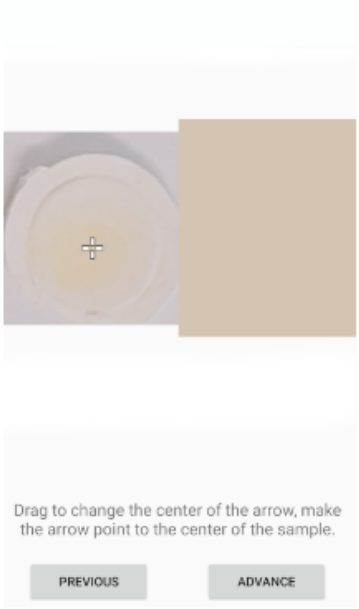


Figure 2.21: Manual Sample Detection [9].

- **Manual selection of the patches to use:** Traditionally, the algorithm will use all of the twenty-four patches available at the color checker, however, in order to improve accuracy, it is possible to use only the n-closest one's to the sample color. With that in mind, there is a settings menu where the user can manually select the n-patches to be used. As illustrated in Figure 2.22, this feature is not intuitive as the user can not visually distinguish each patch, which makes this setting pretty useless since, without distinguishing them, it is impossible to correctly select the thirteen closest to the sample color.

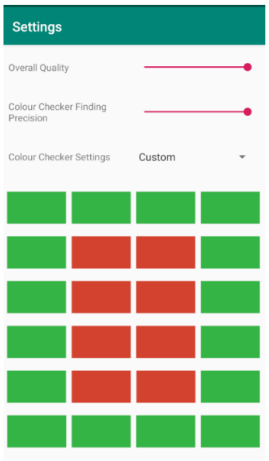


Figure 2.22: Patches Selection Menu [9].

- **Run algorithm and extract result:** After all the previous segmentation process, the application runs an algorithm developed by Carvalho et al. [4]. The results are extracted from the

algorithm and showed as illustrated in Figure 2.8.


Figure 2.23: Results Screen where the estimated concentration is displayed [9].

## 2.5 Summary

Along this chapter a full background on sulfonamides was given. History of mobile phones and their cameras have been explored and their usage as biomedical devices explained. Previous work such as the algorithm of colorimetry and the mobile app was exposed and discussed.

Some of the benefits of using a mobile phone as a biomedical device can be summarized as:

- **Easy and inexpensive deployment:** Nowadays, smartphone app markets such as Google Play Store or Apple's app store provides the developers of medical apps a way to make a low cost deployment to a large amount of people.

- **Availability without barriers:** Smartphone are with us all the time. According to a recent empirical studies smartphones are keep in the same room as their owners over 90 percent of the time [7]. This data makes clear that the owners can diagnose their ailments whenever they want, same thing does not happen when we talk about medical appointments.

- **Frequent hardware upgrade:** We live in a world where we have a new smartphone model each 3 months, with new CPU and GPU capabilities which will make medical apps developed for older phones run better and faster without the need of a lot of modifications.

Side by side with the benefits there are the challenges of turning a smartphone into a biomedical device:

- **They are good but not good enough:** While it is true that smartphones are truly powerful in terms of computational power it is also true that some complex task such as machine learning are still far from running smoothly in a mobile device.
- **The battery:** Battery consumption is a clear barrier and one of the biggest constraints to mobile computing applications [26]. In term of medical apps there is usually an excessive battery usage due to the need of image processing. If no energy efficient techniques are available make the battery last a little longer it is likely that the users will reduce the usage of the medical apps in order to retain battery.
- **Security:** As the number of apps available at the mobile markets increases it also increases the amount of malware that is designed to perform malicious operations namely: unauthorized financial transactions and accessing user sensitive and private information [10]. Medical apps are not free of this problem and have been recently affected by them [32].

A previous approach to the problem was mentioned and discussed on this chapter. This approach will serve as basis for this dissertation.

# Chapter 3

# Methodology

The problem this dissertation aims to solve is the need for a easy and inexpensive way to determine the concentration of SAs in environmental waters. Traditional solutions requires send a sample to a laboratory which consists in a very time consuming and expensive method. With this dissertation we aim to develop a mobile application that through digital image colorimetry can extract the concentration of the compounds mentioned above. This approach will make *in situ* analysis a reality.

An approach like this needs a well-structured methodology and an in-depth study of the technologies to be used, namely in two fundamental topics:

- Cross-Platform application development;
- Machine Learning for Computer Vision applied to mobile devices.

The information flow is also a key information to be defined and described.

Throughout this chapter the previous topics will be addressed and the founded solution explained and justified. An overview of the main requirements of the application along with the chosen information flow will also be described.

## 3.1 Requirements

The first aspect that need to be sorted out before starting the development of the application are the main requirements that it should answer and allow the user to fulfill. This initial step is crucial because it is the vital piece of information when the time for the design and technological decisions arrives.

One of the biggest advantages of this application is that it turns *in-situ* analysis a reality. With this use case in mind, it is possible to identify several functional requirements such as:

1. The user must be able to take a photo or load an already taken one from the gallery to be analysed;
2. The user must be able to define the geographic location of the selected photo;

3. The user must be able to save the selected photo in a way that it can access it later and perform the analysis;

4. The user must be able to see all the previous experiments results as well as the photo that the user saved for future analysis and start the analysis of that photos whenever the user wants;

5. The user must be able to adjust specific steps of the ML algorithms when they are not precise enough;

6. The user must be able to choose which and how many patches he wants to be used in the color correction algorithm;

7. The user must be able to see the final result of the analysis (concentration);

Regarding the requirement number one, the user must select the photo that the user wants to analyse. There are two ways the user can do that, the first is one is by accessing the device gallery and load one of the pictures presented there, for the second option the user can launch the device camera and take a new photo. Figure 3.1 displays the interface that allows this to happen.

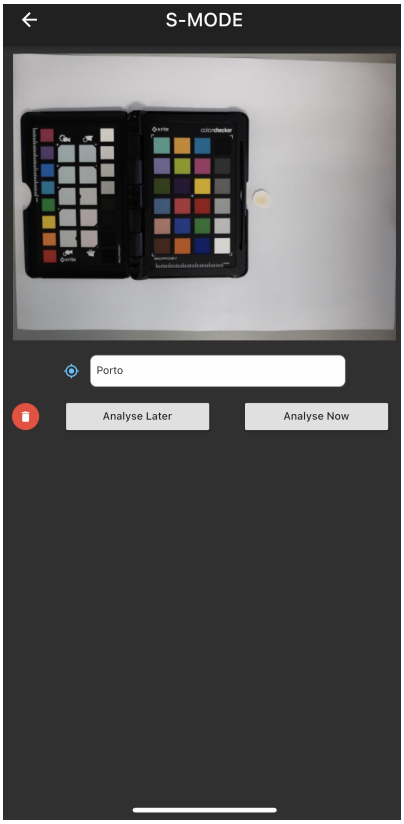


Figure 3.1: Photo screen where the user can select a photo and add a location.Two buttons are displayed, one for storage the photo for later analysis and the other one for immediate analysis of the sample.

In Figure 3.1 is displayed the interface where the user perform the requirement mentioned above. In the interface the user views the image selected for analysis and a location is required for identification means. The user can choose to analyse immediately the sample or analyse it later, either way the picture is stored in the device memory (it will be displayed on the device gallery) and locally in the application so it can be later displayed when the user access the registry of past experiments from where he can analyse all the samples that he chooses to store and view the results of the ones that are already analysed. The location where each photo was taken and the date when it was taken is also stored and displayed in the registry screen. Thus, the interface shown in Figure 3.1 handle the first three requirements mentioned on the list above.

The fourth requirement listed above concerns the possibility for the user to see all the experiments previously analysed as well as the ones that were stored for future analysis. Along with this requirement there is a need for the user to be able to start an analysis on the experiments that have been stored without being previously analysed. To answer both of this needs, a screen, were all the stored experiments are displayed sorted by location, was developed. This screen will be detailed later on this dissertation.

Another important requirement for this application to be considered complete and functional is the possibility to select which and how many patches to use into the color correction algorithm. Although the original algorithm was designed to use all the 24 patches, Carvalho et al. [13] found that better results can be achieved when a specific subset of 13 patches is selected for the color correction matrix. With this in mind, there are three different ways for the user to select patches:

1. A template for all the 24 patches;

2. A template for the 13 best patches to improve the performance;

3. Manually inserting the desired patches.

A complete flowchart describing the flow between all the applications screens and functionalities is displayed in the Figure 3.2

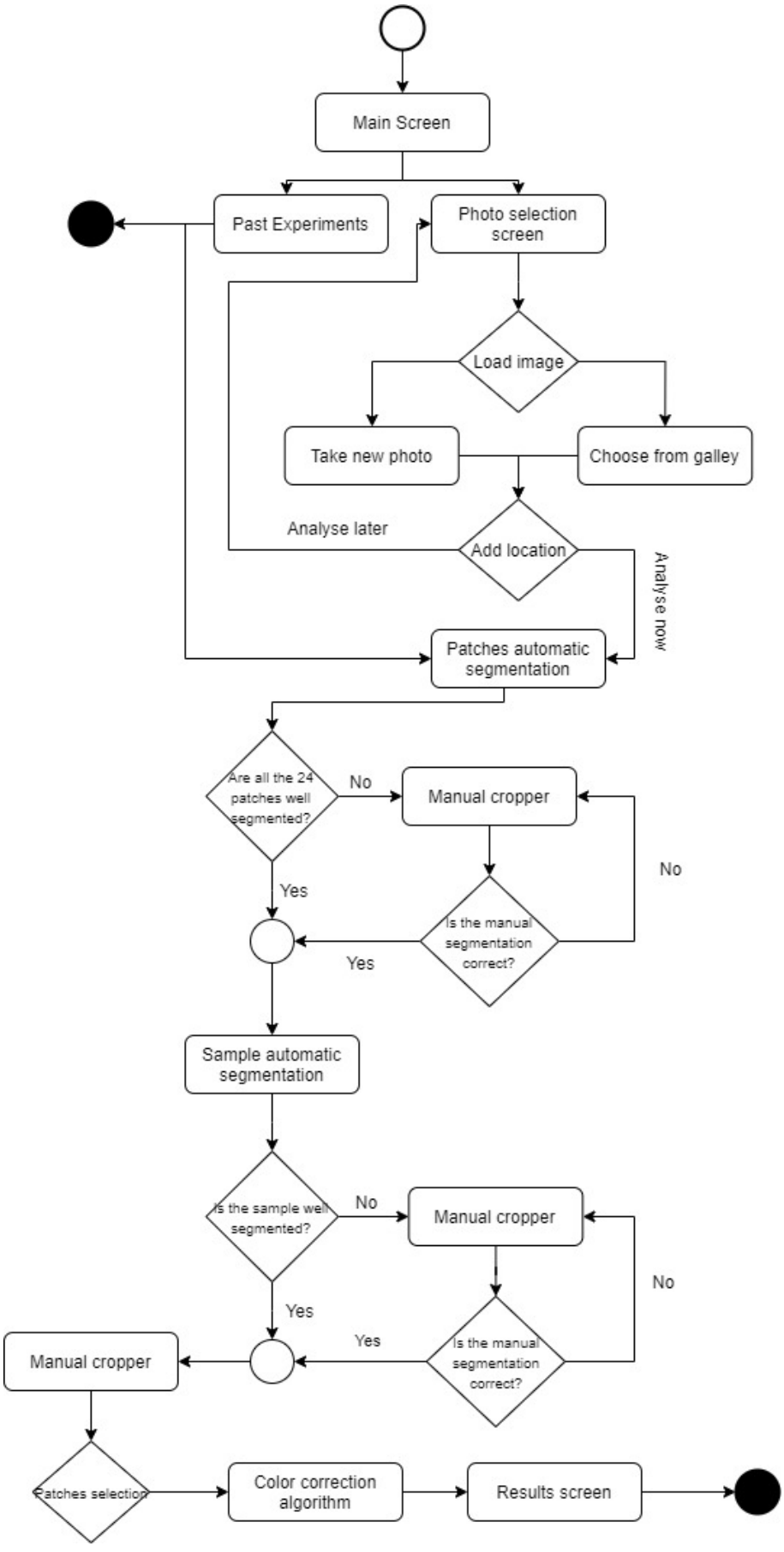


Figure 3.2: Application flowchart

## 3.2 Architecture decisions

### 3.2.1 Flutter

It was clear since the beginning that the application should be runnable on both iOS and Android devices without compromising performance, and to achieve that we choose to work with Flutter.

Flutter is an open-source cross-platform mobile app development framework which allows to build a native application for both iOS and Android with a single codebase. Flutter allows faster development of cross-platform applications and reduce their cost of production. In terms of differentiation, Flutter moves away from the other frameworks since it does not require web browser technology using instead its own high-performance rendering engine improving the overall performance of the applications. Flutter apps are developed using Dart programming language, which is an object-oriented and class-based programming language created by google, capable of performing a great number of function at the same time.

In the list below are enumerated and explained the main reasons why we decided to use Flutter to develop this mobile application:

- **Highly Productive:** Since it is a cross-platform framework, with Flutter is possible to use just one base code to build applications for both iOS and Android which can save a lot of time to the developers;
- **Simple and quick when answering to changes:** This is one of the best qualities of flutter, it can answer any change in the code very quickly without the need of restart which is very helpful for bug fixing and allows to create UIs without pause;
- **Performance:** As it does not need any Java Script Bridge it is higly reliable and can achieve great scores in terms of performance;
- **Easy handling:** Flutter becomes of easy handling as it does not requires bundles of code, rules or any kind of regulations;
- **Compatibility with TensorFlow Lite models:** As both the color correction and the segmentation algorithm consist of TensorFlow models, there was a need to find a framework that could handle them. Flutter has two different libraries capable of handling these models, ensuring maximum performance.

### 3.2.2 TensorFlow Lite

The algorithms that are needed to be implemented on this application consists in ML (Machine Learning) and they must be directly implemented on the target devices without requiring external services and any kind of internet communication.

TensorFlow Lite is a deep-learning open source framework which allows to convert machine learning algorithm into models that are runnable in mobile devices in a complete offline way. It can help solve some common problems such as:

- **Image Classification:** Given an image the model will identify lots of "*objects*" including people, animals, plants and even places. Figure 3.3 demonstrates an example where an object is classified as a banana with 98.82 percent of probability;

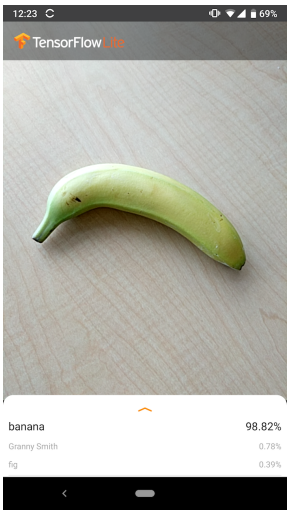


Figure 3.3: Image classification example where the object in the photograph is classified as a banana [35].

- **Object Detection:** Animals, objects and even people are detected with bounding boxes. Figure 3.4 demonstrates a running example of the algorithm where a banana and an apple are detected and a bounding box is applied to each one of them.

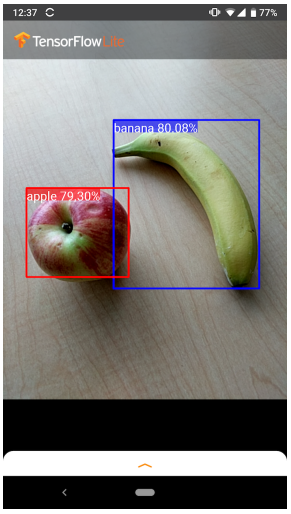


Figure 3.4: Object detection example where an apple and a banana are detected and surrounded by a bounding box [36].

- **Question Answering:** A model that use a state-of-the-art natural language is capable of answering questions based on the content of a given passage of text with BERT [34]. Figure 3.5

consists on an example where the algorithm answers the question "What is tensorflow?" by highlighting the answer in the text.

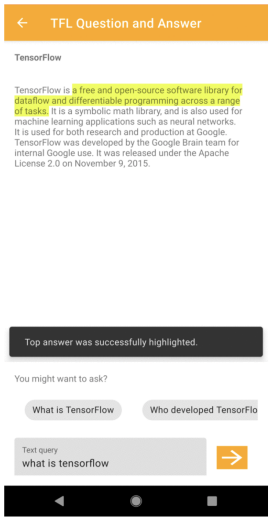


Figure 3.5: Question answering example where the question to be answered is "What is TensorFlow?" and the answer is highlighted within the text [34].

The process attached to this framework consist in four steps:

1. **Select the model:** Choose an already existing model or train a new one;
2. **Convert the model:** With the help of the TensorFlow Lite Converter a TensorFlow model is converted into a fixed buffer;
3. **Implementation:** After creating the models they are sent to the devices within a compact .tflite file. Frameworks like Flutter can easily handle this files which is very helpful for the this project specifically;
4. **Optimization:** TensorFlow Lite supports running the models in the GPU or converting the 32-bit floats to 8-bit integers which improves the efficiency.

## 3.3 Photographs

Before delving into the algorithms themselves it is important to clarify how the photos to be analysed should be structured.

Each photo will have two major points of interest:

- **The color chart (including the patches);**
- **The sample.**

These main points must be placed side by side (horizontally) and the sample can be on the right or left of the color chart. Figures 3.6 and 3.7 are example of how the sample and the chart should be placed before taking the photography. In Figure 3.6 the sample is on the right of the chart and in Figure 3.7 it is placed on the left. The photo must be taken in the landscape direction, that is, horizontally. Taking the photo vertically will produce inaccurate results, as the algorithms are designed to receive inputs where the photos are wider than they are tall.


Figure 3.6: Example photo where the sample is placed on the right of the color chart.

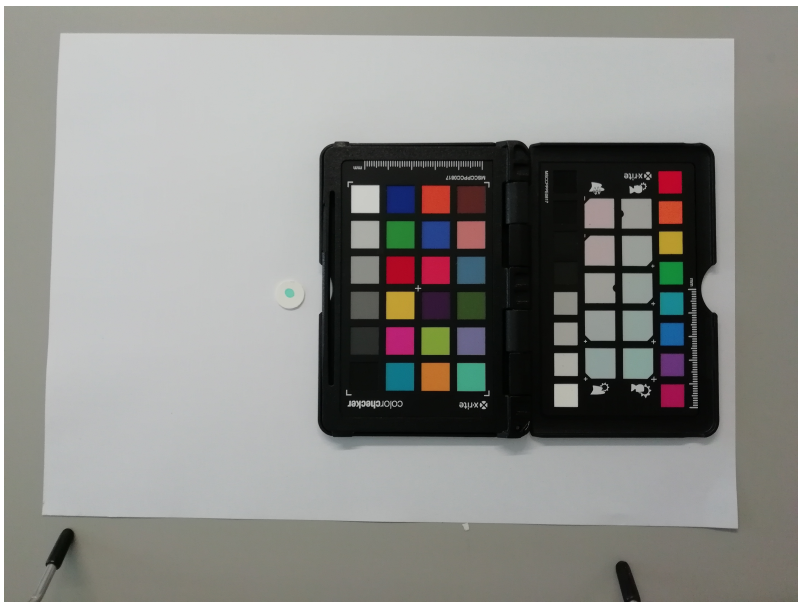

Figure 3.7: Example photo where the sample is placed on the left of the color chart.

## 3.4 Machine Learning Algorithms

First, the algorithms implemented in this dissertation were all developed within the scope of another dissertation, leaded by Inês Rocha [29], that is also associated with the InescTec S-MODE project. This thesis and the thesis responsible for the algorithms complement each other as required by the S-MODE project.

The algorithms were provided in a TensorFlow model format (.tflite files) and, in this project, we provide them with a specific input and an output is returned, which is interpreted and displayed graphically.

Two types of algorithms were provided:

- **Segmentation:** The segmentation algorithm is used for the segmentation of the patches, within the color chart, and the sample, returning bounding boxes for each one.
- **Color correction:** The color correction algorithm is used to correct the color of the sample by using the colors extracted from the patches.

A more detailed explanation about the algorithms will be given in the next sections.

### 3.4.1 Segmentation

#### 3.4.1.1 Patches

Regarding the segmentation of the patches the goal is to obtain a bounding box for each and from that extract its color. For that a SSD MobileNet V2 model is used.

The following changes have been applied to the original algorihtm in order to detect the patches:

- **Preprocessing**
  - The number of classes to be detected was changed to 24;
  - The input images resolution was changed from 300×300 to 512×384 ($width$×height).
- **Training**
  - Batch size changed from 512 to 32;
  - The fine tune checkpoint type was changed from classification to detection.
- **Pos processing**
  - Max detections per class was changed from 100 to 1 (only one detection for patch).

The model was trained using an image dataset with images similar to Figure 3.6 and Figure 3.7 with the resolution set to 512×384 pixels.

As input the algorithm receives a three-channel image of 512×384 resolution. The input tensor has the shape [1, 384, 512, 3] with values for the RGB (Red, Green and Blue) channels in [0, 1].

The output consists in the following elements:

- **Bounding boxes:** A tensor with shape [1, 24, 4] containing the coordinates for the twenty-four bounding boxes in the order: [ymin, xmin, ymax, xmax];

- **Detection score:** A tensor with shape [1, 24] containing the detection scores for each bounding box.

- **Classes:** A tensor with shape [1, 24] containing the class identified for each bounding box.

Graphically speaking, the patch segmentation algorithm results in the image that can be seen in Figure 3.8.


Figure 3.8: Result of the patch segmentation algorithm. For each one of the twenty four patches a bounding box is displayed.

After the segmentation process is complete it is possible to extract the color of each patch and then create a 80×120 resolution image with twenty-four 20×20 squares each one filled with a color belonging to one of the patches from the color chart. Figure 3.9 is a graphically visualization of the image previously described. The color extraction for each patch is done in four major steps:

1. Find the center coordinates of the corresponding bounding box;

2. Calculating a 5×5 square with the center being the one mentioned in the previous step;

3. Extract the RGB channels for each pixel belonging to the previous square;

4. Obtain the RGB channel values for the patch color by averaging between the RGB channel values of the pixels obtained in the previous step.

Figure 3.9: 80×120 resolution image where each 20×20 square is filled with the color of one of the patches of the color chart.

#### 3.4.1.2 Disk and sample

Once the detection and segmentation of the patches is finished, it is time to segment the sample and extract its color. This process is splited in two parts:

- Disk detection;
- Sample color detection and extraction.

The algorithm used for this step consists on a SSD Mobilenet V2 Object detection model with FPN-lite feature extractor, shared box predictor and focal loss [33]. The model was created using the TensorFlow Object Detection API [33] and when trained on COCO 2017 dataset it produces an output as shown in Figure 3.10 where we could see that every detected object is surrounded by a bounding box and a probability value is assigned to each one of them.

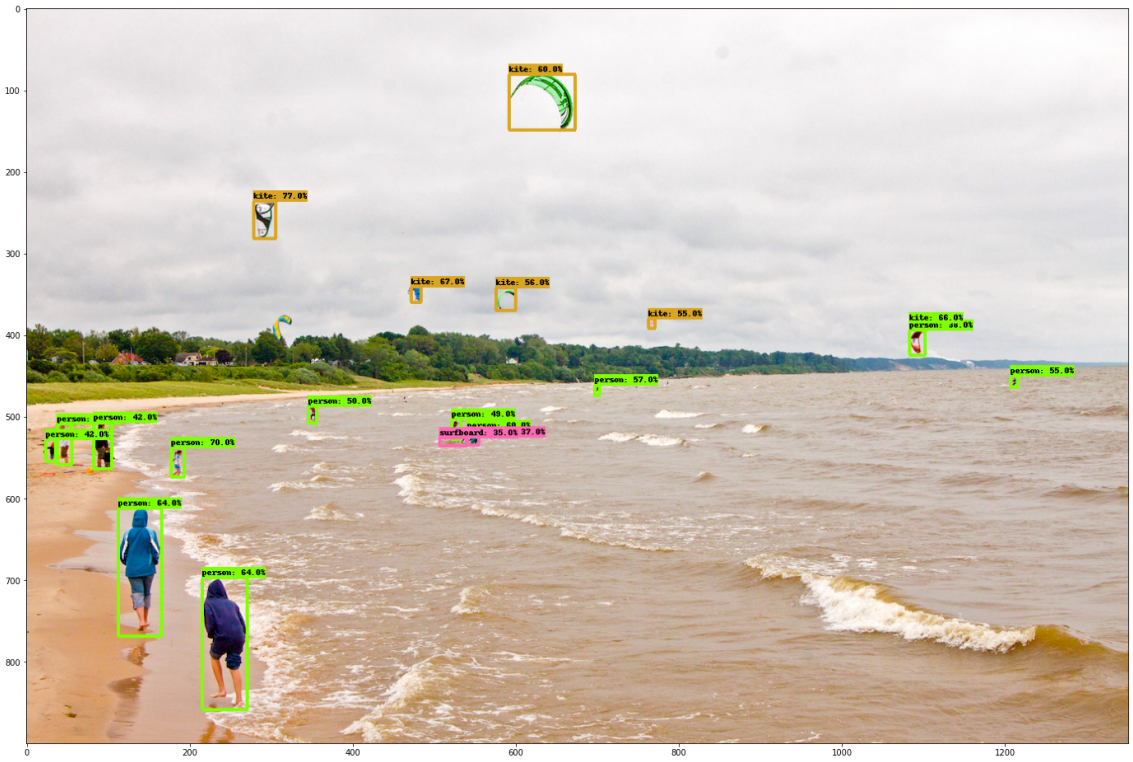


Figure 3.10: Output of the SSD MobileNet FPN-lite algorithm trained with images from COCO 2017 dataset. Boundig boxes are assigned to each detected object.

The following changes were made to the algorith in order to detect the sample and the disk:

- **Preprocessing**
  - The number of classes to be detected was changed to 2;
  - The input images resolution was changed from 300×300 to 256×384 (width * height).
- **Training**
  - Batch size changed from 512 to 32;
  - The fine tune checkpoint type was changed from classification to detection.
  - Added a data augmentation option which consists in a random vertical flip of the image.
- **Pos processing**
  - Max detections per class was changed from 100 to 1 and the max total detections from 100 to 2 (one detection for each of the two classes - the sample and the disk).

For disk and sample detection purposes, this algorithm was trained using images scaled to 256×384 resolution that are the result of cutting the image presented in Figures 3.6 and 3.7 in half. Figure 3.11 represents an example of the described image.

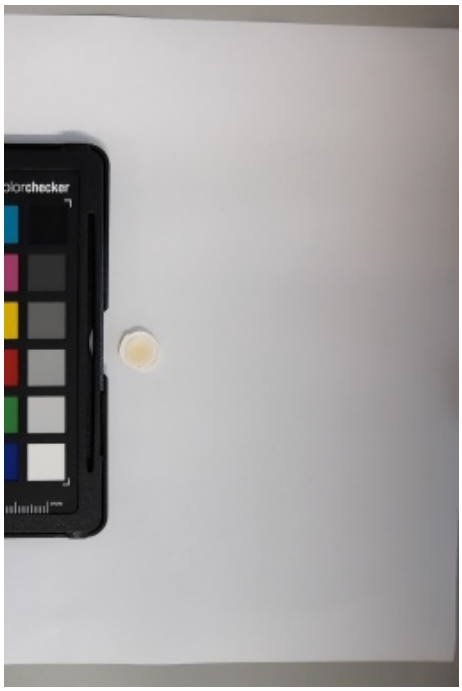

Figure 3.11: Example of 256×384 resolution image used to train the sample segmentation algorithm.

As input the algorithm receives a three-channel image of 256×384 resolution. The input tensor has the shape [1, 384, 256, 3] with values for the RGB (Red, Green and Blue) channels in [0, 1].

In terms of output the algorithm returns the following:

- **Bounding boxes:** A tensor with shape [1, 2, 4] containing the coordinates for the two bounding boxes (the disk with the sample and the sample itself) in the order: [ymin, xmin, ymax, xmax];

- **Detection score:** A tensor with shape [1, 2] containg the detection scores for both bounding boxes.

Figure 3.12 shows the result of the algorithm applied to the image represented on Figure 3.11. A bounding box has been returned and displayed for both the disk and the sample.

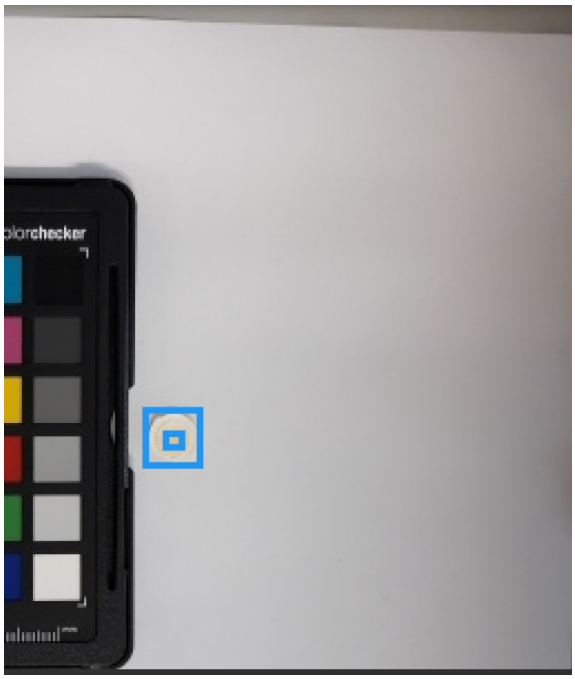

Figure 3.12: Result of the sample segmentation algorithm, the disk and the sample have been detected and a bounding box returned for each one of them.

After the bounding boxes are returned it is possible to finally extract the sample color applying the following steps:

1. Find the center coordinates of the inner bounding box (the one corresponding to the disk);

2. Calculate a 5×5 square with the center being the pointed obtained in step 1;

3. Extract the RGB channels of each pixel that belongs to the square calculated in step 2;

4. Obtain the RGB channel values for the sample color by averaging between the RGB channels values of the pixels obtained in step 3;

5. Create a 20×20 resolution image with each pixel filled with the color obtained in step 4 (displayed in Figure 3.13).

Figure 3.13: Sample color extracted using the method described above.

### 3.4.2 Color Correction

The last machine learning algorithm to be applied refers to the color correction. This step is needed because lightning has an inevitable impact in the colors when a photo is taken. Color correction aims to correct colors by eliminating ray interference, bringing them closer to their true value.

The algorithm chosen by Inês Rocha [29] is a variation of color constancy by deep learning. Color constancy by deep learning aims to estimate the color of the light source [20]. This technique consists in a convolutional neural network with a structure as the one show in Figure 3.14. The activation is done by Rectified Linear Unit activation function also known as ReLU Layer. In terms of optimization the Adam algorithm was the chosen one. Adam optimization consists in a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments.

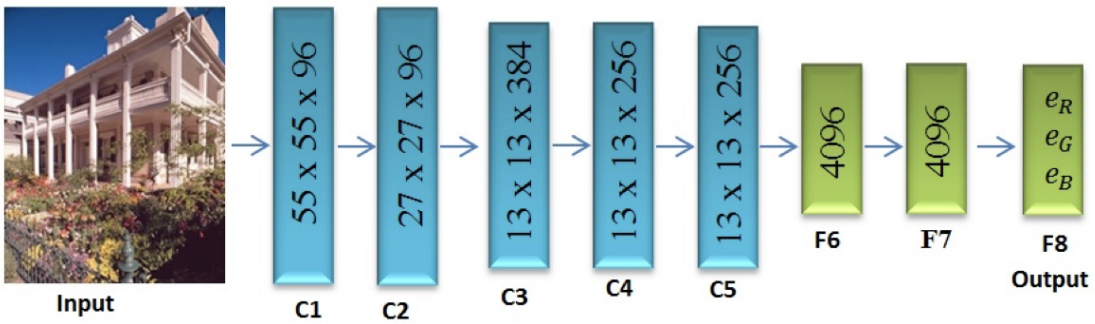


Figure 3.14: The architecture of the convolutional neural network. From the total of eight layers, the first five are Conv2D layers and the last three are Dense layers [20].

In order to create a model that could be used on a mobile phone, the neural network layers had to be reduced in size:

- C1 changed from 96 to 32;

- C2 changed from 96 to 64;
- C3 changed from 384 to 256;
- C4 and C5 changed from 256 to 196;
- F6 changed from 4096 to 1024;
- F7 changed from 4096 to 128.

In terms of training, the model is trained with 100×100 resolution images that are composed by 25 20×20 resolution squares, being the first (the top left one) filled with the sample color to be corrected and the remaining 24 filled with the colors segmented of the color chart. An example of this image can be seen in the Figure 3.15.

Figure 3.15: Image with 100×100 resolution, to be corrected, where the top-left 20×20 square is filled with the sample color to be corrected and the remaining 24 with the color chart patches.

In the Figure 3.16 it is displayed the color of the sample before applying the color correction and in the Figure 3.17 it is displayed the sample color after going through the color correction model.

Figure 3.16: Sample before color correction.



Figure 3.17: Sample after color correction.

As input the algorithm receives a three-channel image of 100×100 resolution. The input tensor has the shape [1, 100, 100, 3] with values for the RGB (Red, Green and Blue) channels in[0, 1]. The output is an identical tensor with the RGB channels fixed to the correct colors.

## 3.5 Validation

Usability has become a key factor in determining the acceptability and consequent success of any kind of software from programs to applications [27]. Users are less and less willing to accept applications with unattractive and unintuitive interfaces, which demonstrates that these are factors to take into account during the development phase and that should be evaluated before the final release of the application.

Usability is a non-measurable concept, but it is related to several parameters that can be measured. Measurable usability parameters are divided into two groups:

- **Objective performance measures:** The ones that measure how users are able to use the system;
- **Subjective measures of user preference:** The ones that measure the user opinion.

Effectiveness, efficiency and satisfaction can be seen as critical criteria that influence usability. To evaluate these criteria, it is necessary to break down into sub-criteria [27].

Therefore, a questionnaire was carried out in which the questions aimed to assess the usability of the application's interfaces. As mentioned above, one of the usability criteria is player satisfaction, so this questionnaire allowed us to observe some satisfaction sub-criteria. The questions were the following:

1. **Is it appropriate for the main task?**
2. **Does the user have control over the interface?**
3. **Is it explicit?**
4. **Is it consistent?**
5. **Does it fulfill the user expectations?**
6. **Is it easy to use?**

Each question can be rated on a scale from 1 to 5, where 5 is the most positive possible and 1 the most negative. The questionnaire can be found in Annex A.1.

## 3.6 Summary

This chapter presents all the information related to the chosen methodology. The user requirements have been described and detailed. The main architectural decisions on what platforms to use for developing the application have been announced and justified. An overview on how the machine learning algorithm works and how they are executed on a mobile device is presented in this chapter as well. The validation method chosen to validate the usability and acceptance of the application is also detailed. In the next chapter the results obtained by applying the methodology described in this chapter will be demonstrated.

# Chapter 4

# Results

This chapter is split-ed in two main sections. The first corresponds to the application and there are described and displayed all of the screens of the application as well as the flow between them. The second is about the validation process and is where the results of the usability questionnaire are described and analysed.

## 4.1 Application

In this section the following screens of the application are described and displayed:

- Main screen;
- Photo selection screen;
- Patches segmentation screen;
- Sample segmentation screen;
- Patches selection screen;
- Results screen;
- Past experiments screen.

### 4.1.1 Main screen

The application main screen is very simple, having a button at the center of the screen with a wave effect around it. This screen is represented in Figure 4.1. From here the user can navigate to:

- **Photo selection screen:** By pressing the central button with the text "ANALYSE";
- **Past experiments screen:** Pressing the menu button in the AppBar followed by selecting the option "Past Experiments" will take the user to the past experiments screen.


Figure 4.1: Application main screen with a button at the center that leads to the photo selection screen and a menu icon button in the AppBar that allows the user to access the "Past Experiments".

### 4.1.2 Photo selection screen

The photo selection screen allows the user to select the photo of the sample he wants to analyse, add a location to it and start an analysis immediately or saving it for later. This screen consists in the following elements:

- **Camera Button:** The button with the camera logo when pressed allow the user to choose a photo from the gallery or take a new one using the mobile phone camera. After the has been taken or selected this button is replaced with it;
- **Location InputTextBox:** In this text the user must add a location where the photo belongs;
- **Analyse Now Button:** When pressed, an analysis of the selected photo starts instantly. If no photo is selected or the location field is empty, the user is warned and the analysis does not start.
- **Analyse Later Button:** When pressed the selected photo and location are saved and sent to the past experiments to be analysed later;
- **Discard Button:** The selected photo and location are deleted.

Figure 4.2 shows the photo selection screen when neither the photo or the location have been added and Figure 4.3 shows the photo selection screen after adding both of them.

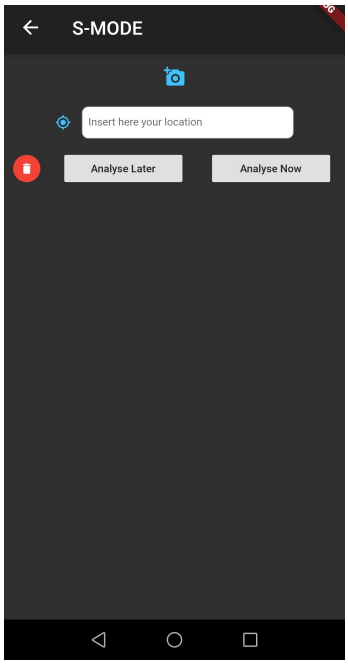


Figure 4.2: Photo selection screen where the user can select a photo from the gallery or take a new one, add a location and start an analysis or save it for later analysis.

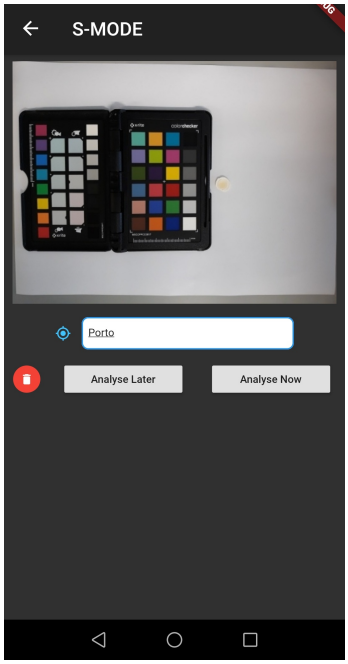


Figure 4.3: Photo selection screen with a photo selected and a location added.

### 4.1.3 Patches segmentation screen

The first step of the entire process is the segmentation of the patches. The segmentation is made by a ML model and the result is displayed in the form of bounding boxes around the patches of the color chart. Figure 4.4 shows the patches segmentation screen where is possible to see the selected photo with the bounding boxes around the 24 patches.

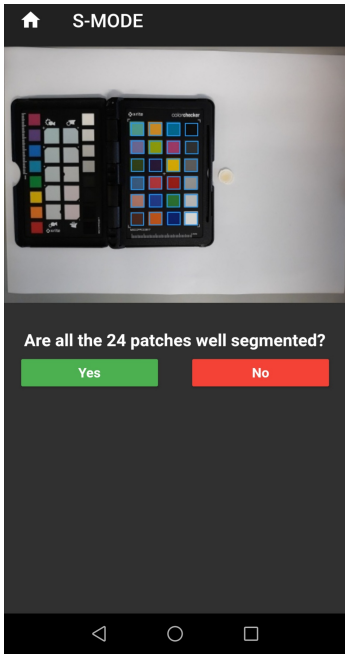


Figure 4.4: Result of the patch segmentation algorithm. For each one of the twenty four patches a bounding box is displayed.

Despite the high accuracy and precision of the ML model sometimes some of the patches are not well detected, so the user is asked if the segmentation is correctly done. When the segmentation is wrong and the user answers accordingly a cropper tool is displayed and the user can manually crop the patches. Figures 4.5 and 4.6 represent the cropper aspect in iOS and Android, respectively.

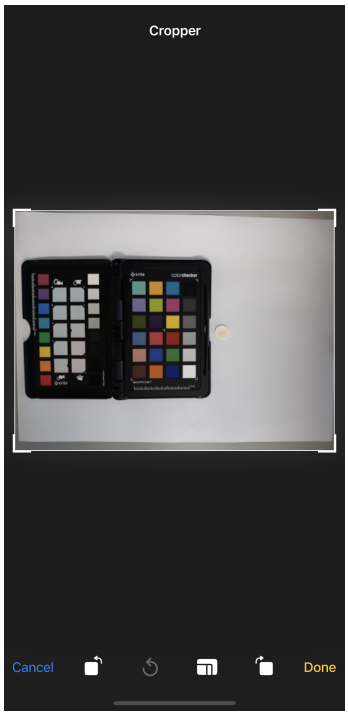


Figure 4.5: Patches manual cropper in iOS devices where the user can crop the patches as much accurate as possible.


Figure 4.6: Patches manual cropper in Android devices where the user can crop the patches as much accurate as possible.

Before the cropper appears, a dialog box is displayed giving the user guidance on how to crop

the patches. This dialog box can be seen in the Figure 4.7 for Android and in the Figure 4.8 for iOS.

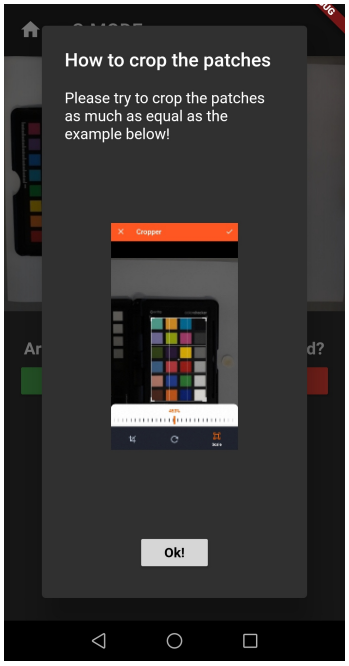


Figure 4.7: Dialog box on Android devices with guidance on how to crop the patches.

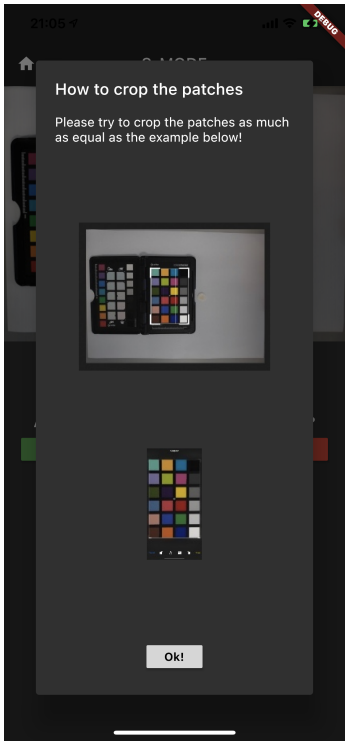


Figure 4.8: Dialog box on iOS devices with guidance on how to crop the patches.

After the user manually crops the patches, a new screen appears in order to the user verify the result of the manual crop. The user is asked if the crop and color extraction is correctly done and if the user selects *'no'* then the user will be redirect to the previous screen and asked to crop the patches again. Figure 4.9 shows the mentioned screen.

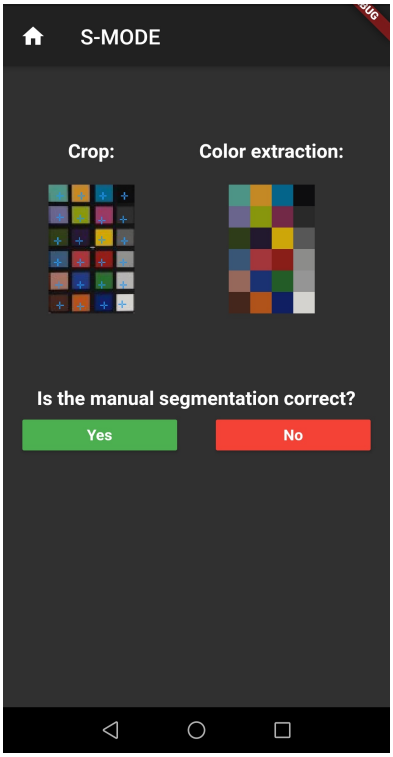


Figure 4.9: Patches crop verification screen where the result of the crop and of the color extraction are displayed to the user.

### 4.1.4 Sample segmentation screen

Once the patches have been segmented, it's time to do the same with the sample. The sample segmentation screen is identical to the patches segmentation screen and the user can choose to manually segment the sample, just as he can with the patches. In the Figure 4.10 it is possible to see the screen layout. Figures 4.11 and 4.12 shows the cropper for Android and iOS devices, respectively. The dialog box with guidance on how to crop the sample are displayed in the Figures 4.13 and 4.14 for Android and iOS devices, respectively.

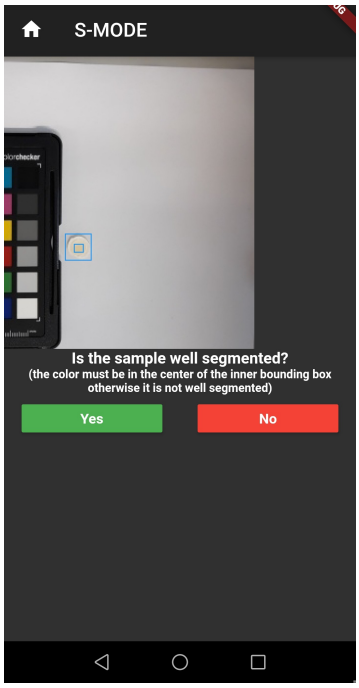


Figure 4.10: Sample segmentation screen where the disk and the sample have been detected and a bounding box applied to each one of them. The user can accept the result or choose to manually crop the sample.


Figure 4.11: Sample manual cropper in Android devices where the user can crop the sample as much accurate as possible.

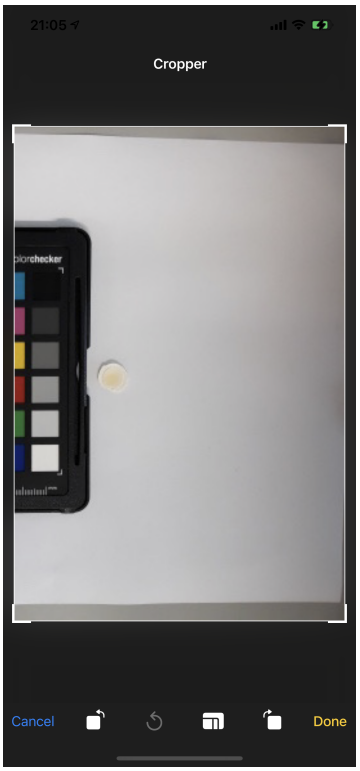


Figure 4.12: Sample manual cropper in iOS devices where the user can crop the sample as much accurate as possible.

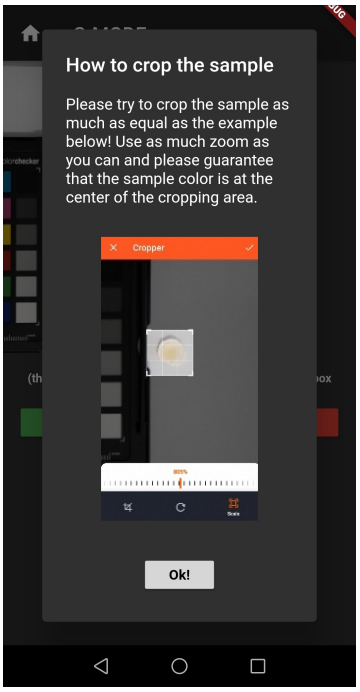


Figure 4.13: Dialog box on Android devices with guidance on how to crop the sample.

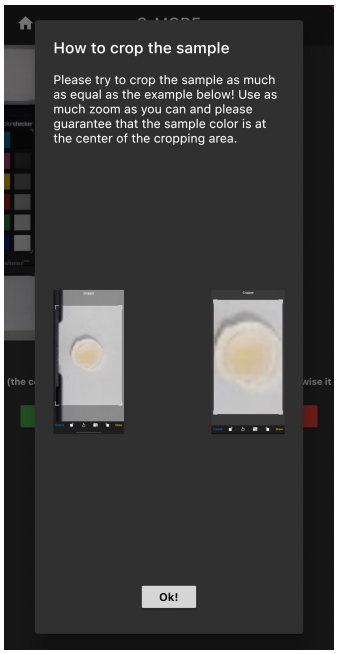


Figure 4.14: Dialog box on iOS devices with guidance on how to crop the sample.

After the user manually crops the sample, a new screen appears in order to the user verify the result of the manual crop. The user is asked if the crop and color extraction is correctly done and if the user selects *'no'* then the user will be redirect to the previous screen and asked to crop the sample again. Figure 4.9 shows the mentioned screen.

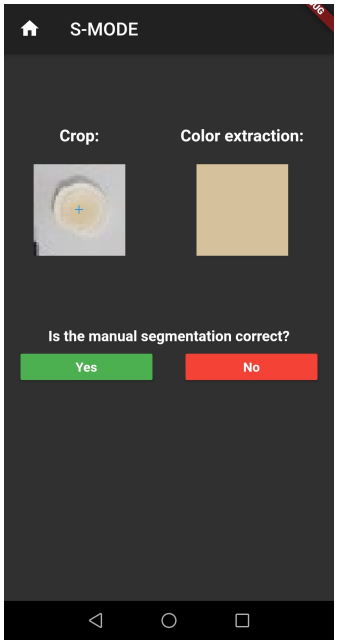


Figure 4.15: Sample crop verification screen where the result of the crop and of the color extraction are displayed to the user.

### 4.1.5 Patches selection screen

The color correction algorithms uses by default all the 24 patches of the color chart, however, it can also work with any n-patches that the user wants. Moreover, Inês Rocha [29] based on the work done by Carvalho et al [4] proved that a specific group of 13 patches could achieve the best results in terms of color correction. This said, in the patches selection screen the user has the possibility to choose how he wants the algorithm to work choosing one of the following options:

- **Templates:**
  - **T24 or t24:** Template used when the user wants to send all the 24 patches to the color correction model;
  - **T13 or t13:** Template used when the user wants to send a pre-defined set of the best 13 patches.
- **Manual insertion:** If the user does not wants a pre-defined template he can insert manually the patches he wants to be used. The patches are numbered according to the grid visible in Figure 4.16.

The user selection will be saved and the next time the user arrives in this screen that selection will already be displayed in the input box. From this screen the user navigates to the results screen where the calculated concentration will be displayed.

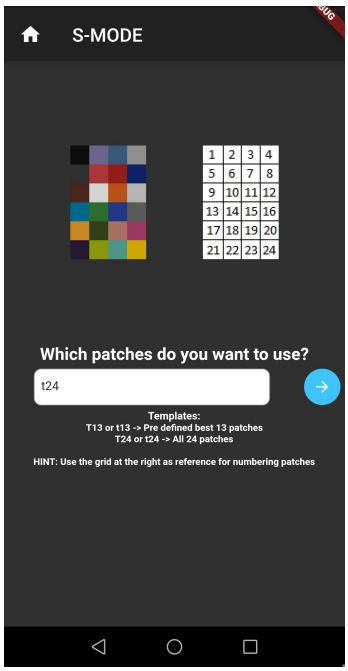


Figure 4.16: Patches selection screen where the 24 patches are displayed side by side with a grid that helps to numerate them. In the input box the user inserts the patches he wants to use or a pre-defined template.

### 4.1.6 Results screen

The results screen is where the sample concentration is displayed to the user along with the final color of the sample after applying the color correction ML model. This is the final screen in the application flow and from here the user saves the analysis and go back to the main screen. Figure 4.17 shows the described screen.

The concentration value is obtained by applying the following formula, which, according with the work done by Carvalho et al [4] and Pedro Reis [9] in the S-MODE projetc, relates the hue value of the corrected color of the sample to the concentration: $concentration = 0.034 \times (sampleColorHue/360)^{-9.124}$

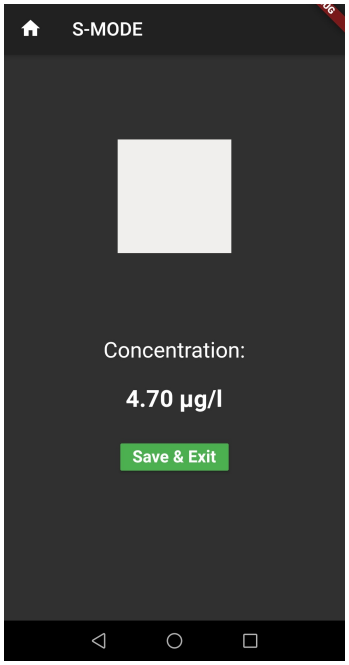


Figure 4.17: Results screen where the concentration of the sample is displayed along with its color after applying the color correction algorithm.

### 4.1.7 Past experiments screen

The past experiments screen is where a registry of usage of the application is stored. Two types of registries are stored here:

- **Already analysed samples:** The result of the analysis is displayed along with the date and a photo of the sample.
- **Samples to be analysed:** The sample that the user chooses not to analyse immediately are displayed and it is possible to start an analysis from here.

The entries are ordered and grouped by location and the user can delete any entry whenever we wants. Figure 4.18 shows the past experiments screen with an entry that has already been

analysed, it is displayed the date when the analysis occurred and the obtained result Figure 4.19 shows the past experiments screen with an entry that has not been analysed yet, the date when it was stored is displayed and it is possible for the user to start an analysis from here.

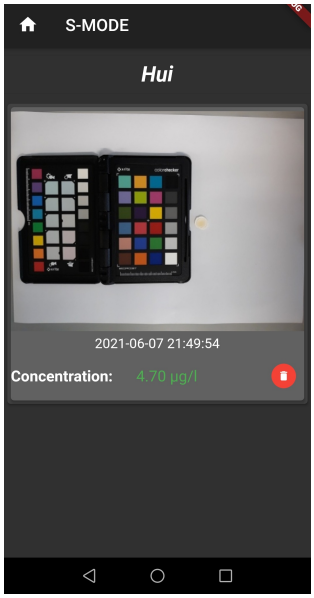


Figure 4.18: Past experiments screen with an entry that has already been analysed. The date, location, photo of the sample and the concentration are displayed.

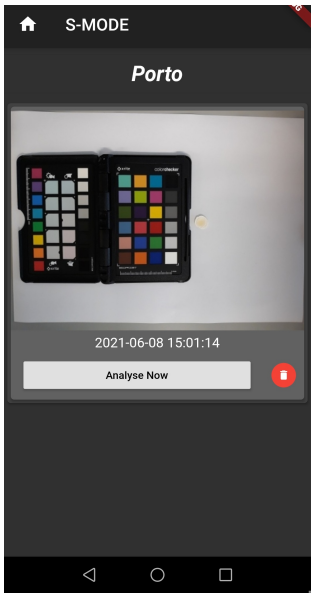


Figure 4.19: Past experiments screen with an entry that has not been analysed yet. The date when it was saved is displayed and it is possible to start an analysis from here.

## 4.2 Validation

As mentioned in section 3.5 an usability questionnaire was made with the propose to evaluate and measure the usability of the application. The target audience of this dissertation is a group of researchers specialized in chemistry who are associated with the S-MODE project, so the survey was delivered to them. It was possible to obtain a total of five different answers that are in the Annex B.1

In Table 4.1, it is possible to observe the average of the ratings assigned to each question as well as their standard deviation.

| **Question** | **1** | **2** | **3** | **4** | **5** | **6** |
|---|---|---|---|---|---|---|
| **Average** | 4.8 | 4.6 | 5.0 | 5.0 | 4.8 | 4.4 |
| **Standard deviation** | 0.4 | 0.5 | 0.0 | 0.0 | 0.4 | 0.5 |

Table 4.1: Usability questionnaire results

As can be seen in Table 4.1 of the answers to the questionnaire, all had a positive evaluation. However, the last question stands out a little from the previous ones, as some of the respondents found it difficult to manually crop the patches and the sample in the first attempt. When observing the standard deviation it is verified that questions 2 and 6 have a greater deviation which happens due to the issue of the difficulty felt by some users in the tool of manual cropping the patches. Regarding questions 3 and 4, it is possible to observe that they were given the highest rating by all respondents, so we can conclude that the application is quite explicit and consistent. The results obtained in questions 1 and 5 demonstrate that the application corresponds to the users' requirements and is able to give a very good answer to the problem in question.

# Chapter 5

# Conclusion

The use of antibiotics in the treatment of humans and animals is highly recurrent and with an increasing trend. It is its low cost and lack of toxicity that lead to its widespread use. Despite the benefits it has some severe side effects such as polluting aquatic environments. The presence of this compounds in environmental waters cause the non antibiotic-resistant bacteria to die causing the emergence of antibiotic-resistant bacteria, which will make intractable diseases more likely to appear.

The determination of the concentration of sulfonamides in water samples requires the use of laboratory equipment, which makes this solution impractical and expensive. This dissertation provides a cheaper and more practical solution to this problem.The application developed within the scope of this dissertation allows, through digital image colorimetry and the application of machine learning algorithms for detection and segmentation, to quickly and accurately determine the concentration of these compounds in a water sample without the need for any internet connection or resorting to a laboratory making local analyzes a reality. The application is cross-platform being able for both Android and iOS users.

To summarize, the developed application is equipped with the following functionalities:

- The user can upload a photo for analysis either from the gallery or taking a new one from the camera;
- The user is able to add a location to the photo that will server as identification;
- The user is able to store the photo for future analysis or to analyse it immediately;
- The user is able to choose if he accepts the automatic result of the segmentation algorithms (patches and sample) or if he pretend to manually crop them.
- The user is able to select which and how many patches he wants to be used in the color correction algorithm. The application provides by default two different templates: T24 (all the patches) and T13 (the thirteen patches that provide a more accurate result);
- The user is able to see the result and save it;

- The user is able to access a registry of past experiments where he will find all the photos that have already been analysed and the ones he stored for future analysis and start an analysis on this ones

## 5.1 Future Work

The main requirements that the S-MODE project assigned to this dissertation were successfully completed, however there are some aspects that can be improved or added in the future:

1. **Support for different types of samples:** Sulfonamides are not the only compound that needs to be controlled and monitored. Initially, the S-MODE team wanted this application to be able to analyse the concentration of fluoroquinolones which consists on black-box fluorescence photos, however there were not enough data to proceed with the implementation. If the data could be acquired this an interesting and important feature to add to the application.

2. **Machine learning for calculating the concentration value:** Initially, it was planned to use a machine learning model to calculate the concentration of the compound in the water sample. However, the algorithm was not provided in time for implementation, so the concentration is calculated by applying a mathematical equation that relates the concentration with the hue value of the sample corrected color.

3. **Support for different types of color charts and samples:** This solution is limited to the specific color chart described above. This is because of the top thirteen patches feature. This feature allows the user to use only the best thirteen patches to correct the color of the sample; however, these thirteen patches are chart-specific and also sample dependent, which limits the use of this application. In the future the user should be able to identify which chart is going to use as well the colorimetric reagent used in the sample so the best-thirteen patches can be determined accordingly.

# Appendix A

# Usability Questionnaire

**Questionnaire about the S-MODE application usability criterions**

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| **Is it appropriate for the main task?** [1] | | | | | |
| **Does the user have control over the interface?** [2] | | | | | |
| **Is it explicit?** [3] | | | | | |
| **Is it consistent?** [4] | | | | | |
| **Does it fulfill the user expectations?** [5] | | | | | |
| **Is it easy to use?** [6] | | | | | |

[1] The application should satisfy all the user demands in terms of sulfonamides detection.

[2] The application must allow the user to interact with the interface in an easy and fluid way.

[3] Information on how to use the main features of the application should be given as well as help and orientation when an input is needed.

[4] The graphic aspects and functionalities should remain consistent.

[5] The graphic aspect of the application should be compatible with the user demand.

[6] The application and its features should be easy to use.

Figure A.1: Usability Questionnaire [27]

# Appendix B

# Usability questionnaire answers

| Question nº / Questionnaire | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 5 | 5 | 5 | 5 | 4 | 4 |
| 2 | 5 | 4 | 5 | 5 | 5 | 4 |
| 3 | 4 | 5 | 5 | 5 | 5 | 4 |
| 4 | 5 | 5 | 5 | 5 | 5 | 5 |
| 5 | 5 | 4 | 5 | 5 | 5 | 5 |

Figure B.1

# References


[1] E. Agu, P. Pedersen, D. Strong, B. Tulu, Q. He, L. Wang, and Y. Li. The smartphone as a medical device: Assessing enablers, benefits and challenges. In *2013 IEEE International Conference on Sensing, Communications and Networking (SECON)*, pages 76–80, 2013.

[2] E. Agu, P. Pedersen, D. Strong, B. Tulu, and L. Wang. Wound image analysis system for diabetics. In *Proc. SPIE Medical Imaging*, 2013.

[3] Dhruv Bhutani. Throwback: Mobile camera tech was amazing even before iphone and android. Available at https://www.androidauthority.com/phone-camera-history-1128076/, June 2020.

[4] Pedro H. Carvalho, Sílvia Bessa, Ana Rosa M. Silva, Patrícia S. Peixoto, Marcela A. Segundo, and Hélder P. Oliveira. Estimation of sulfonamides concentration in water based on digital image colourimetry. In José Salvador Sánchez Aythami Morales, Julian Fierrez and Bernardete Ribeiro, editors, *Pattern Recognition and Image Analysis*, pages 355–366. Springer International, 2019.

[5] Barbara Chiavarino, Maria Elisa Crestoni, Annito Di Marzio, and Simonetta Fornarini. Determination of sulfonamide antibiotics by gas chromatography coupled with atomic emission detection. *Journal of Chromatography B: Biomedical Sciences and Applications*, 706(2):269 – 277, 1998.

[6] Eric E. Connor. Sulfonamide antibiotics. In *Primary Care Update for OB/GYNS 5(1)*, pages 32–35, 1998.

[7] A. K. Dey, K. Wac, D. Ferreira, K. Tassini, J-H. Hong, and J. Ramos. Getting closer: an empirical investigation of the proximity of user to their smart phones. In *Proc. UbiComp*, 2011.

[8] Stanislava Dmitrienko, Elena Kochuk, Vladimir Apyari, Veronika Tolmacheva, and Yu Zolotov. Recent advances in sample preparation techniques and methods of sulfonamides detection - a review. *Analytica Chimica Acta*, 850, August 2014.

[9] Pedro Daniel dos Santos Reis. Android application for determination of sulfonamides in water using digital image colorimetry. Available at https://hdl.handle.net/10216/129103, July 2020.

[10] A. P. Felt, M. Finifter, E. Chin, S. Hanna, and D. Wagner. A survey of mobile malware in the wild. In *Proc. SPSM*, 2011.

[11] Centers for Disease Control and Preventation. Antibiotic resistance threats in the united states 2019. Available at https://www.cdc.gov/drugresistance/pdf/threats-report/2019-ar-threats-report-508.pdf.

[12] Malgorzata Gbylik, Sikorska, Andrzej Posyniak, Tomasz Śniegocki, and Jan Zmudzki. Liquid chromatography–tandem mass spectrometry multiclass method for the determination of antibiotics residues in water samples from water supply systems in food-producing animal farms. *Chemosphere*, 119:8–15, 01 2015.

[13] Carvalho P. H., Rocha I., Azevedo F., Peixoto P. S., Segundo M. A., and Oliveira H. P. Color correction approach on uncontrolled lighting conditions. In *International Conference on Computer Analysis of Images and Patterns*, 2021.

[14] Peter Ha. Motorola dynatac 8000x. Available at http://content.time.com/time/specials/packages/article/0,28804,2023689_2023708_2023656,00.html, October 2010.

[15] Rodrigo Hoff and Tarso Kist. Analysis of sulfonamides by capillary electrophoresis. *Journal of separation science*, 32:854–66, February 2009.

[16] T. Korpimäki, E. Brockmann, Outi Kuronen, M. Saraste, U. Lamminmäki, and M. Tuomola. Engineering of a broad specificity antibody for simultaneous detection of 13 sulfonamides at the maximum residue level. *Journal of agricultural and food chemistry*, 52 1:40–7, 2004.

[17] E. C. Larson, T. Lee, S. Liu, M. Rosenfeld, and S. N. Patel. Accurate and privacy preserving cough sensing using a low-cost microphone. In *Proc. UbiComp*, 2011.

[18] Chenglong Li, Xiangshu Luo, Yonghan Li, Huijuan Yang, Xiao Liang, Kai Wen, Yanxin Cao, Chao Li, Weiyu Wang, Weimin Shi, Suxia Zhang, Xuezhi Yu, and Zhanhui Wang. A class-selective immunoassay for sulfonamides residue detection in milk using a superior polyclonal antibody with broad specificity and highly uniform affinity. *Molecules*, 24:443, January 2019.

[19] Ying Lin. 10 mobile usage statistics every marketer should know in 2021. Available at https://www.oberlo.com/blog/mobile-usage-statistics, April 2020.

[20] Zhongyu Lou, Theo Gevers, Ninghang Hu, and Marcel P. Lucassen. Color constancy by deep learning. In Xianghua Xie, Mark W. Jones, and Gary K. L. Tam, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 76.1–76.12. BMVA Press, September 2015.

[21] Mobius MD. 11 surprising mobile health statistics. Available at https://www.mobius.md/blog/2019/03/11-mobile-health-statistics/, March 2019.

[22] Mark Muldoon, Carol Holtzapple, Sudhir Deshpande, Ross Beier, and Larry Stanker. Development of a monoclonal antibody-based celisa for the analysis of sulfadimethoxine. 1. development and characterization of monoclonal antibodies and molecular modeling studies of antibody recognition. *Journal of Agricultural and Food Chemistry*, 48:537–44, March 2000.

[23] S. O'Dea. Market share of mobile operating systems worldwide 2012-2020. Available at https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/, November 2020.

[24] The Editors of Encyclopaedia Britannica. Colorimetry. Available at https://www.britannica.com/science/colorimetry, February 2014.

[25] World Health Organization. Antimicrobial resistance: Global report on surveilance 2014. Available at https://apps.who.int/iris/bitstream/handle/10665/112642/9789241564748_eng.pdf, Accessed last time in 21 December 2020.

[26] J. A. Paradiso and T. Starner. Energy scavenging for mobile and wireless electronics. In *IEEE Pervasive Computing*, pages 18–27, February 2005.

[27] Kyung S Park and Chee Hwan Lim. A structured methodology for comparative evaluation of user interface designs using usability criteria and measures. In *International journal of industrial ergonomics*, volume 25, pages 379–389, 1999.

[28] M. Z. Poh, D. J. McDuff, and R. W. Picard. Advancements in noncontact, multiparameter physiological measurements using a webcam. In *IEEE Trans Biomedical Engineering*, volume 58, pages 7–11, 2011.

[29] Inês Rocha. Computer vision algorithm for determination of sulfonamides in water. 2021.

[30] Conduct Science. Colorimetry- a comprehensive guide for your color science. Available at https://conductscience.com/colorimetry-a-comprehensive-guide-for-your-color-science/.

[31] Paulo Jorge Teixeira Silva. Digital image colorimetry for determination od sulfonamides in water. Available at https://hdl.handle.net/10216/106905, July 2017.

[32] D. Talbot. Computer viruses are "rampant" on medical devices in hospitals. In *MIT Technology Review*, October 2012.

[33] TensorFlow. Available at https://hub.tensorflow.google.cn/tensorflow/ssd_mobilenet_v2/fpnlite_640x640/1.

[34] TensorFlow. Bert question and answer. Available at https://www.tensorflow.org/lite/examples/bert_qa/overview.

[35] TensorFlow. Image classification. Available at https://www.tensorflow.org/lite/examples/image_classification/overview.

[36] TensorFlow. Object detection. Available at https://www.tensorflow.org/lite/examples/object_detection/overview.

[37] Uswitch. History of mobile phones and the first mobile phone. Available at https://www.uswitch.com/mobiles/guides/history-of-mobile-phones/#disqus_thread, December 2020.

[38] Lidian Vilazio. The cellphone evolution. Available at https://medium.com/@lidianvilazio/the-cellphone-evolution-8d2cd607f379, April 2018.

[39] X-Rite. What is a spectrophotometer / color spectro? Available at https://www.xrite.com/learning-color-education/other-resources/what-is-a-spectrophotometer.