THESIS

COMPUTER VISION ALGORITHM TO EXTRACT COLOR DATA OF PIXELS IN MICROFLUIDIC PAPER BASED ANALYTICAL DEVICES

Submitted by

Saurabh Deotale

Department of Computer Science

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Summer 2021

Master's Committee:

Advisor: James Ross Beveridge

Nathaniel Blanchard
Charles Henry

Copyright by Saurabh Deotale 2021

All Rights Reserved

ABSTRACT

COMPUTER VISION ALGORITHM TO EXTRACT COLOR DATA OF PIXELS IN MICROFLUIDIC PAPER BASED ANALYTICAL DEVICES

Microfludic paper-based devices are fast becoming a inexpensive and faster option than traditional methods for substance detection and chemical measurements. These devices are designed to be used in the field for quicker result. One hurdle towards that goal is a manual step of data extraction from the images of these devices for further analysis and results. This involves identifying and extracting color data from specific regions of interest. The color data is the color values in BGR and HSV color channels of the pixels lying in these regions of interest. The manual demands labor and time that can avoided by automating this process using computer vision techniques. The goal of this thesis is to aid chemists by automating the data extraction process. This thesis presents a layered algorithm which uses simple techniques like region growing and thresholding in conjunction with leveraging the knowledge of the device design to extract the required data. This data is then labeled and compiled in CSV file for further analysis.

ACKNOWLEDGEMENTS

I would like to thank Dr. Ross Beveridge, Dr. Charles Henry and Dr. Nathaniel Blanchard for their constant guidance, patience and encouragement. A special thanks to Ruth Menger and Kate McMahon for their contribution, guidance and constant help in this research work.

I would also like to thank my fellow graduate students, especially Viraj Shastri and Dhruva Patil for their constant help in critiquing and reviewing my work. I would like to extend my gratitude to my family for their constant support. I would also like to thank Elliot Forney for creating and maintaining the LaTeXtemplate. The template was immensely helpful in creating this document. Lastly, I am grateful to the Department of Computer Science at Colorado State University for giving me the opportunity to work with the Computer Vision group and the Henry group in the Department of Chemistry.

DEDICATION

I would like to dedicate this thesis to my family.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Microfluidic paper based devices are fast becoming a popular device for different substance(s) identification and chemical measurements. They provide an inexpensive and faster option than traditional methods [1] [2]. With the technological advances made in these paper-based devices there is still one manual step involved which demands time and labor. The way these devices are designed, there are specific regions of interest that need to be analyzed to provide any kind of result. Chemists are working on possible digital analysis techniques which works with the captured images of these devices. The analysis requires color data of pixels from specific regions in the image. These regions are where the chemical reaction takes place. The color data from these regions is then used for further analysis.

The data extraction process for the images is done manually. This simple task can be automated using computer vision techniques. Previous work involved using techniques like region growing algorithm [3] and mathematical morphology recognition [4]. This body of work presents an algorithm that can achieve the automation of the color data extraction process from the pixels in specific regions of interest.

Following previous work [3], simple region growing seemed like a promising solution. Such basic techniques were not sufficient by themselves in extracting the required data. But a layered approach to find the true centers of regions of interest turned out to be a better approach. A combination of thresholding in different color channels segmented the regions of interest and then a region growing algorithm is used to calculate the centers of the regions of interest. The knowledge of the design of the device is a crucial input for the algorithm's working. The calculated centers are then used to identify the pixels lying in the required regions of interest.

Roadmap

The rest of the document is structured as follows: Chapter 2 explains in detail the basic techniques involved in the algorithm as well as how these techniques work. It also discusses the similar and related research regarding automating the data extraction process from such paper-based devices. Chapter 3 explains what the microfluidic paper-based devices are, how they work, and what problem are this thesis focuses on. It also explains the pre-processing steps involved in the algorithm. Chapter 4 discusses how the approach progressed along with the pitfalls. Chapter 5 explains the algorithm which is successful in extracting the required data from the images of the paper-based devices. Chapter 6 discusses the result and possible avenues of future work.

Chapter 2

Related Work

This chapter reviews crucial techniques used in the proposed algorithm. Simple computer vision techniques like median filter, region growing algorithm have been used for a long time. Such simple techniques are combined together to solve one simple problem at a time. There are many variations associated with these techniques, but this chapter discusses them in a way to build up the knowledge required to understand the proposed algorithm.

Section 2.1 goes over how smoothing is done using a median filter, this filter is used to reduce the resolution of the images in the presented algorithm. Section 2.2 goes over how region growing can be used to segment a cluster of pixels from the rest of the images. These techniques were first used with gray-scale images, but the same principle works with color images as well. Sections 2.3 and 2.4 discusses techniques used to segment specific regions of interest form different paper-based devices. The pixel level data is then extracted for these regions, which is similar to the work presented in this thesis.

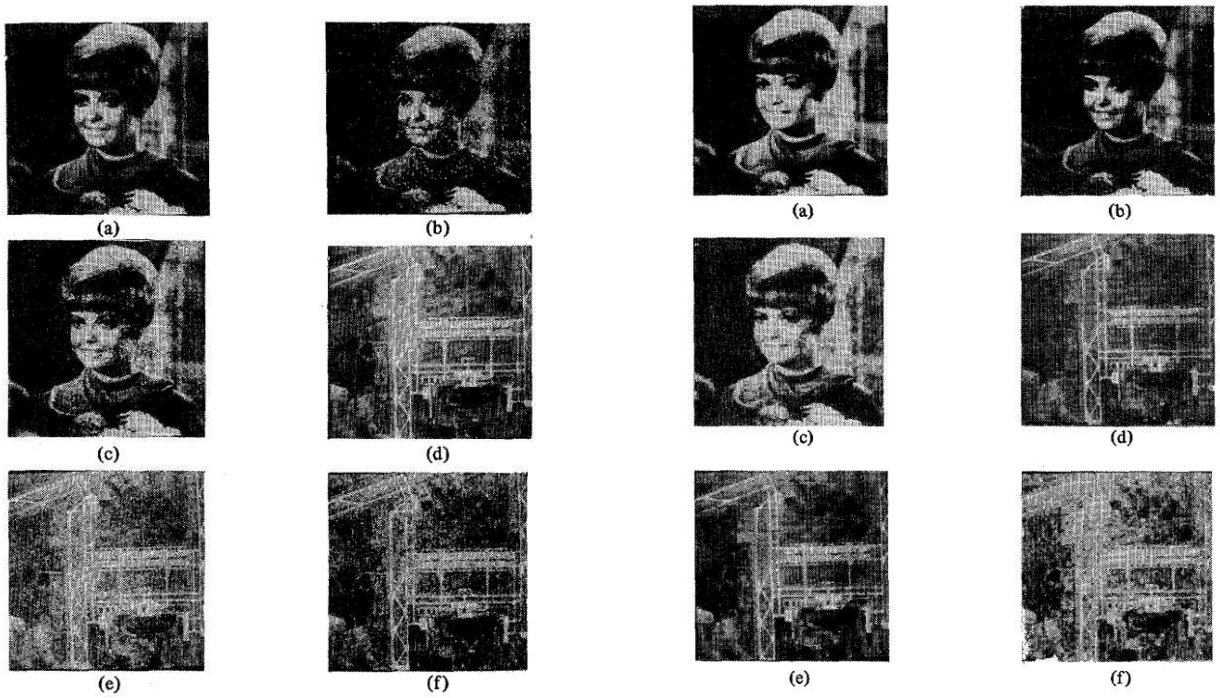
2.1 Median filter

Median filtering is a technique used to smooth noisy images [5]. The median filter does not change the resolution or the size of the image but it changes the color values of pixels.

The starting point is the two-dimensional matrix representation of any image. Where each element of the matrix represents a pixel position and the value of the element is the color value held by the pixel. The next step is to define the filter window size (mxn). This window is significantly smaller than the image. The idea is to slide the window through the image and modify the pixel values in each window. The process can be started at any corner of the image.

Once the window is set the color values of all the pixels are sorted and a median value is calculated. This median value is then assigned to all the pixels in this window. The next step is to remove one column from the start and add another column towards the end, moving the window

inwards. The paper [5] also discusses a fast median algorithm which reduces the burden of re-calculating the median value for every window by sorting the color values. Once this process is completed the image is rendered again. In Figure 2.1 we can see images without median filtering in Figure 2.1a and after median filtering in Figure 2.1b.



(a) Images before median filtering. **(b)** Images after median filtering.

Figure 2.1: (a) Girl, original. (b) Girl, with noise. (c) Girl, with Gaussian noise. (d) Airport, original. (e) Airport, with BSC noise. (f) Airport. , with Gaussian noise. This is Fig 1 and Fig 2 from Pham et al [5].

Median filter works very well with random noise, but as evident from the image it does not work well with Gaussian noise. But this is not a concern as median filter is used for scaling down an image in this work.

Median filter for scaling down an image

For scaling down an image the median filter window is used with a slight change than how it is used in a filter. The windows are independent and there is no intersection between any two placements of the window. This implies instead of shifting the window by one column, it is shifted

by n columns for a window of size ***nxn***, where n is the number of rows and columns. The median color value is calculated and that values replaces the whole window. This results in a new image which is scaled down by a factor of n. This topic is covered in more detail in section 4.1.2.

2.2 Region growing

A region growing algorithm is an image segmentation techniques which finds pixels of similar features clustered together. In this case the features are color values of the pixels. A simple region growing algorithm needs a seed pixel to start with. The next step is to check if the neighbours of the seed pixel can be included in the region based on a pre-defined criteria. This step is repeated a number of time till the terminating condition is reached. The criteria for neighbouring pixels to be included in the region and for terminating the algorithm can be based on the requirements of image segmentation. The next subsections discuss various ways region growing is used for segmenting regions from an image.

2.2.1 Seeded Region Growing

The seeded region growing algorithm is based on the idea that pixels in a region are similar to each other. The starting point is a seed pixel or a group of pixels as demonstrated by Adams et al. (1994) [6]. The process involves starting with these seed pixels and examining their immediate neighbors. This process was demonstrated on a grayscale image [6], so the criteria for similarity of pixels in a region is based on their grayscale value.

The algorithm starts by identifying a set of all the immediate neighbor pixels which are not included in the region. For a lone seed pixel, this set will consist of its eight neighbor pixels. Once this set is identified, each pixel in it is examined if it belongs in the region based on its similarity with the region. The similarity between a region and the pixel in question is decided by the difference between the grayscale values of the pixel and the mean grayscale value of all the pixels already identified to be in the region. First, for all the pixels identified as in the region a mean grayscale value is calculated. Initially this is the grayscale value of the seed pixel or the

mean value of all the seed pixels in a region. The pixel is then put in the list of all the boundary pixels, and the position in the list is determined by the difference of the grayscale values.

The next step is to pick the boundary pixel with the smallest difference and test its neighbours. All the neighbours will have one of three labels: "boundary", "in", "out". If all the neighbours have the same label, ignoring the ones labelled as "boundary" then the pixel is marked as visited and is labelled the same. If the pixels is now labelled as "in" then the mean grayscale value is updated. The paper [6] advocates the advantage of not updating the difference values for all the boundary pixels as the change is negligible. This accounts for the rapid processing of all the pixels.

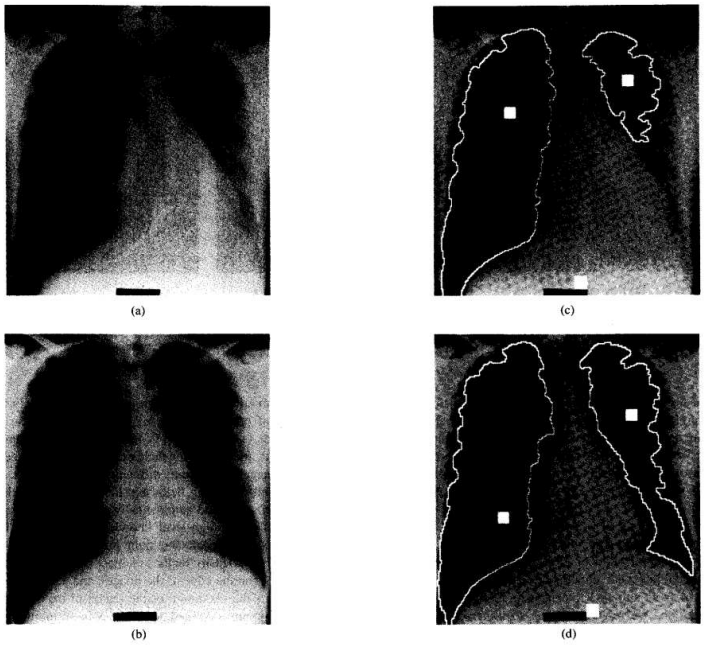


Figure 2.2: (a) X-ray image of human chest in which the left and right lung fields are to be segmented. (b) A second X-ray image of human chest with very different left lung field. (c) Result of segmentation of (a) by SRG using automatically derived seed areas (10 x 10 boxes). These seeds, for the left and right lung fields and the region in between them, are found using a converging squares algorithm. (d) Result of segmentation of (b). This is Fig 5 from the paper by Adam et al. [6]

The Figure 2.2 demonstrates region growing in grayscale images of human chest X-ray. Instead of choosing a seed pixel square region of pixels were selected as pixel. The reason behind this is that a single pixel chosen as seed might be an outlier in a region which is not similar to the pixels in its neighbourhood. This might result in region growing algorithm terminating prematurely. The

paper advocates starting with an area as seed to overcome any noise in the image and/or region. The paper also discusses ways to to automatically identify these seed regions, but that process is not a necessary step in this work.

2.2.2 Region growing along the boundary

The approach discussed in the previous subsection has no objective to control the direction of growth of the region. It simply focuses on adding the most similar pixels to the region. This section goes over one of the ways the direction of the pixels being selected can be controlled, which is presented by Hojjatoleslami et al. (1998) [7].

The paper presents their work on MR (grayscale) images of head. The images have high contrast regions required to be segmented from the image. The paper presents a framework which uses two properties: average contrast and peripheral contrast. It also involves defining layers of regions, namely internal boundary, current boundary and the current region. In Figure 2.3 the separation of these three layers can be seen.

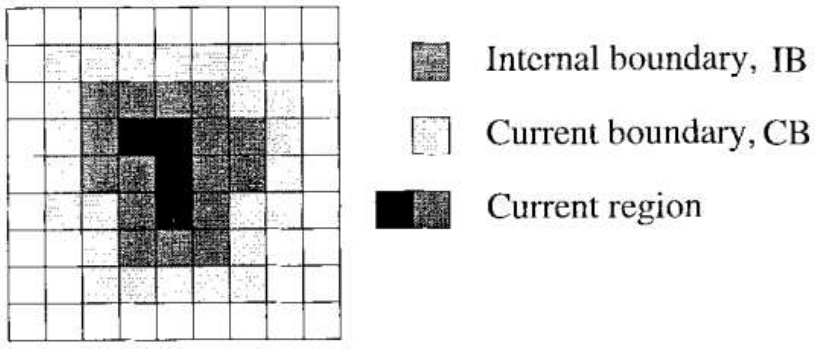


Figure 2.3: Schematic graph shows the CB, including candidate pixels to be joined to the region, and IB, including the outermost pixels of the region, during the growing process. The region contains 20 pixels. This is Fig 2 from the paper by Hojjatoleslami et al. [7]

The average contrast measure is the difference between the average gray level of the pixels in the region and the current boundary. The peripheral contrast measure is the difference between the average gray level of the pixels in the internal boundary and the current boundary. The idea is to

include one pixel at a time in the region having highest intensity. Once the region starts growing and subsuming pixels at the boundary the average contrast for the region starts decreasing. This indicates when the region starts growing in the background, so the average contrast is maximised.

The paper advocates the use of peripheral contrast measure for segmentation as it proves to be more useful in separating the pixels of low and high intensity. For a relatively homogeneous region the peak peripheral contrast value will be unique, but for a noisy image there are multiple local peaks. A single or multiple peaks can be used to maximise the peripheral contrast which in turn separates the high intensity region from the background.

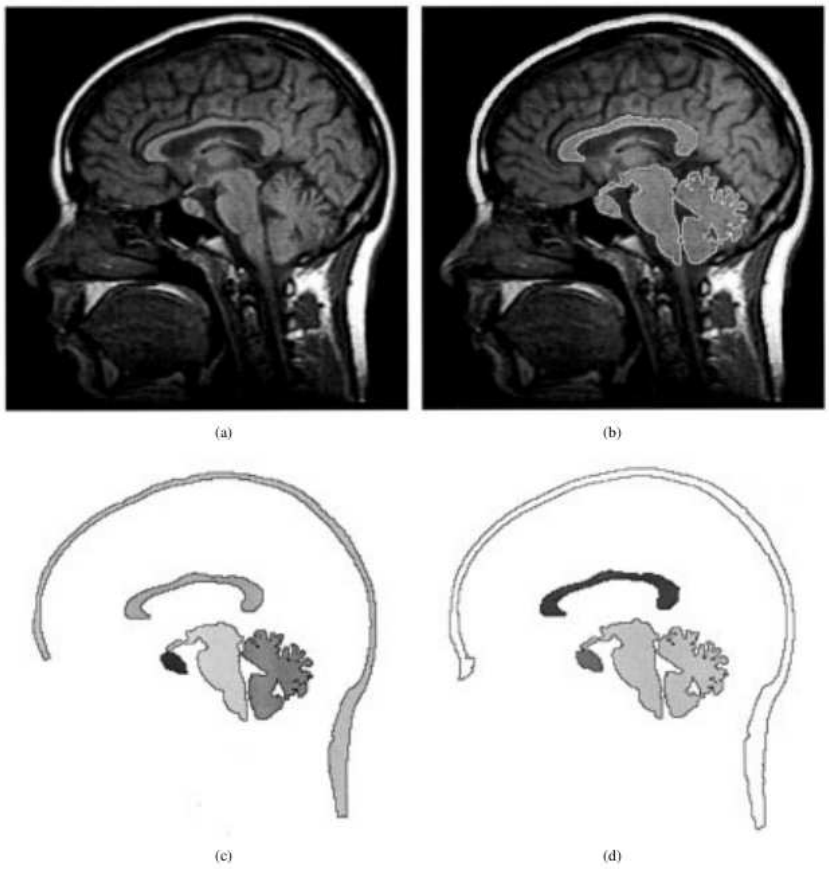


Figure 2.4: (a) Original MR image of head. (b) Segmentation result of five regions superimposed on the original image. (c), (d) peripheral contrast and average contrast regions and boundaries segmented out from image (a). This is Fig 4 from the paper by Hojjatoleslami et al. [7]

In Figure 2.4 the segmented regions are identified using the average contrast and peripheral contrast measures.This body of work simplifies the process by first segmenting the boundary region and then converting it to binary image instead of grayscale. This means the region inside the boundary is homogeneous and contrast between the region and its boundary is quite stark. This helps in segmenting regions of interest easily.

2.2.3 Thresholds and Multi-band thresholding

In region growing algorithms there have been a variety of criteria to decide whether any pixel being examined belongs to a region or not [8,9]. Zucker (1976) [8] has surveyed different methods adopted for this purpose. This subsection discusses a couple of approaches relevant to this body of work.

Threshold selection

One of the simplest technique is to select a constant threshold for a region in an image [8]. One example being the difference between intensities of the pixel and the region to be less than a certain tolerance threshold. This tolerance threshold can be defined according the set of images being examined for a particular problem statement, as the condition for a region being segmented will vary.

Another way of doing this is varying the tolerance threshold [8] all over the picture. This means that instead of a constant value, a function can be used which varies the tolerance threshold value as the region grows. One example of this is re-calculation of average intensity of pixels whenever a new pixel is added to a region. The tolerance threshold can then be defined as a function the standard deviation. A constant k can be multiplied to the standard deviation and the resultant value will act as the tolerance threshold. This implies that the difference between the average intensity of the region and the intensity value of the pixel can be compared against k times the standard deviation. If the difference falls below the tolerance threshold then the pixel is included in the region.

Multi-band thresholding

Pham (2007) et al. [9] demonstrates another method of image segmentation, but this is based on low-pass and high-pass filters on different color channels in CMYK model for an image. The goal of their work is to quantify immunohistochemistry (IHC) stains. This process is used for diagnosis of cancer and other diseases.

Their work found that analysing intensity of the yellow color channel in CMYK model helps in gauging the IHC intensity changes. This detection through the image analysis process shows that thresholds over different components of color values can provide more data about the images then just the perceived color changes through human eye.

2.3 Region growing in paper-based Salmonella immunoassay

The work presented by Carell (2019) et al. [3] discusses how a simple region growing algorithm can be used to segment specific region of interest from a microfluidic paper-based analytical devices (μPADs). A μPAD, referred as sample layer insert [3], is placed in a rotational manifold. A sample flows through the manifold and deliver the reagent on the sample layer / μPADs and then substrate is added to it which results in color change of the device. The Figure 2.5 shows the rotational manifold device, and the component labelled as *Sample Layer (ii)* is the μPAD.

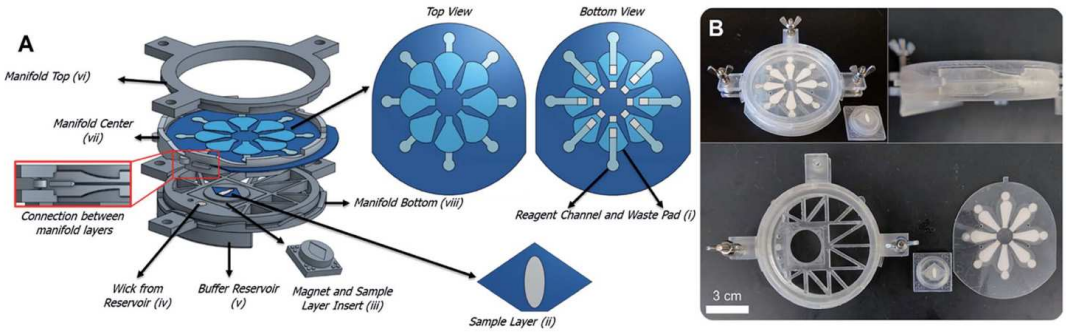
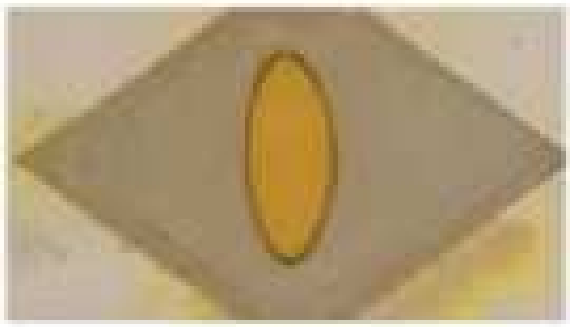


Figure 2.5: A CAD render of the rotational manifold and its 3D printed version. This is Fig 1 from the paper by Carell et al. [3]

Once the color change on the μPAD has occurred, Salmonella can be detected based on the observation of the color change. Researchers and chemists usually extract this color data manually using software like ImageJ [10] for devices such as these. But the design of the paper-based device is such that the data required is clustered in a single region surrounded by a boundary. The region is also distinguishable from its surrounding boundary. This makes using computer vision techniques for segmentation a viable option.



(a) Paper-based device showing no color change after adding substrate.

(b) Paper-based device) showing color change after adding substrate.

Figure 2.6: (a) Sample layer (paper-based device) showing no color change after adding substrate. (b) Sample layer (paper-based device) showing color change after adding substrate. This is from Fig 2 from the paper by Carell et al. [3]

The Figure 2.6 shows how the sample layer / paper-based device is used for colorimetric detection. The rotary manifold device lets the sample solution to be tested flow onto the paper-based device. Once this process is done, a colorimetric substrate (CPRG) is added on to the paper-based

device. This substrate is a liquid solution that is responsible for the color change in the oval region of the paper-based device. Based on the change in color of this region of interest, a colorimetric analysis will give the result of Salmonella detection.

The Figure 2.6a shows the paper-based device with no color change in the oval region. In Figure 2.6b there is a significant color change, showing examples of presence and absence of Salmonella. The stark color difference is not expected as the change in color varies for different samples. Hence, the need for colorimetric analysis of the oval region. The data for this region is gathered using a simple region growing algorithm implementation in the OpenCV [11] library.

The process starts with a seed pixel inside the region. The position of this pixel can be estimated using the geometry of the device, which is fixed. The region growing algorithm starts growing outwards towards the oval boundary. The change in color from the background near the oval boundary helps stop the region growth.

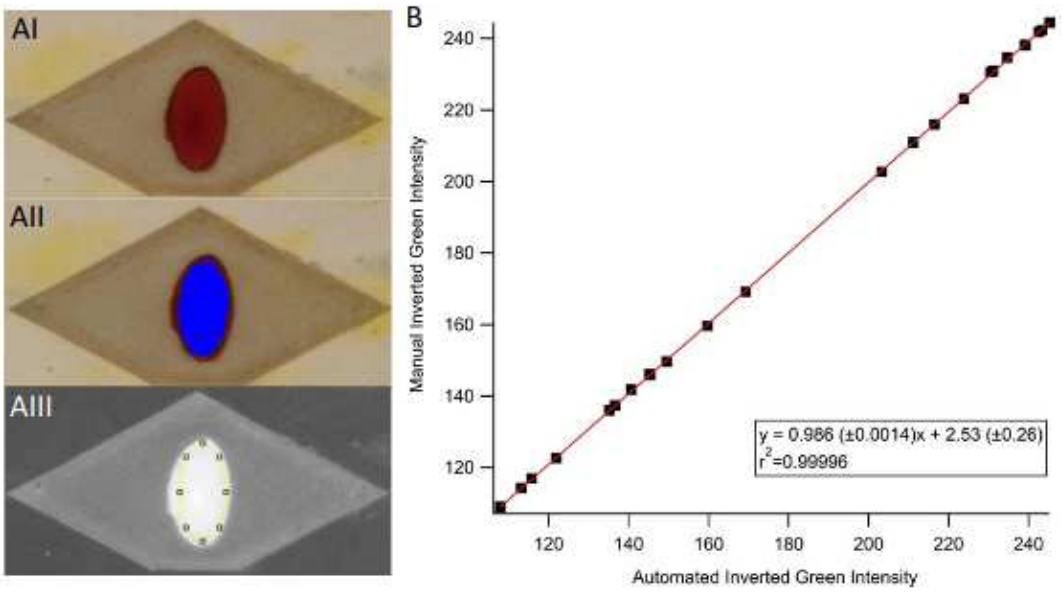


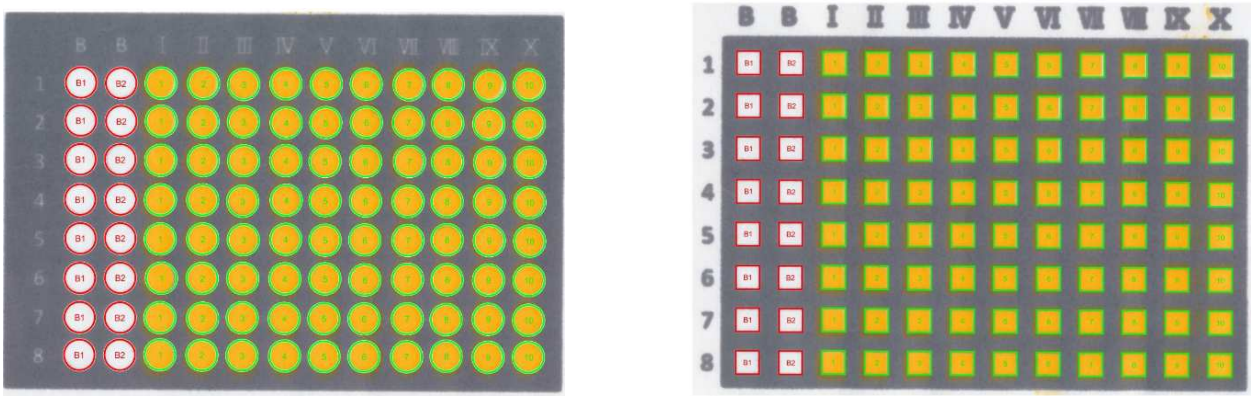
Figure 2.7: Original image (AI), area of original image analyzed by flood fill algorithm (AII), manual analysis of one -color channel. Inverted green intensity found using a manual analysis vs the inverted green intensity found with the automated algorithm. This is Fig S3 from the paper by Carell et al. [3]

The Figure 2.7 shows the region growth using the algorithm. The data collected using the algorithm is compared to manually collected data. The comparison is done for the inverted green channel which shows identical curves for manually collected data and the data collected using the algorithm. The reason for selecting the inverted green color values to compare the analysis is because it is one of the signals used for colorimetric detection.

The advantage of automating the process is it not only reduces the manual labor involved but it also helps in discarding the pixels inconsistent with their neighbors. Thus, increasing the accuracy and reducing the image processing time. This shows the advantages of using computer vision techniques to automate the process of pixel data collection aiding chemists in reducing labor and time spent meanwhile maintaining the accuracy of the analysis. This is one of the approaches that can be adopted for solving such problems. The next section goes over another approach to solve a similar kind of problem.

2.4 Morphological color image processing

In the previous section an application of a simple paper-based device was discussed. There are various designs possible for such devices. The paper by Hamedpour et al. [4] presents a solution automating the data collection process for one such design.



(a) **(b)**

Figure 2.8: (a)Paper-based device with circular regions. (b)Paper-based device with square regions. These images are from the paper by Hamedpour et al. [4]

The Figure 2.8 shows the images of the paper-based device being used for the experimental setup. These devices are different from the one discussed in the previous section. The devices have a layout of multiple regions in square or circular shapes. These regions have reagents in them that react with the solution being tested. Once the reaction is complete, the color change in these regions is used to calculate results. The goal of the paper is to present an approach to automate the color data extraction form these regions. This approach can also be adopted for other paper-based devices of similar design. The approach takes advantage of known layout of the device and uses it as the starting point. The number of regions of interests, their shape and the size of the regions is fixed for any given device. Using this data as starting point the regions of interests are isolated and then the required color data is extracted.

The devices shown in Figure 2.8 have fixed number of regions. The process involves compiling the data for the shape of the regions of interest, their size and the number of regions of interest. This data is used as the starting step for the algorithm. Each of these regions are surrounded by grey boundaries. The idea is to segment the regions of interest from the background using these boundaries. The Figure 2.9 shows one row of 12 square regions of a paper-based device. Each of these regions is surrounded by grey boundaries. This knowledge of the design of the device is essential to how the algorithm segments the required regions.

Figure 2.9: Example of an original RGB image. It shows one row of 12 square regions. This is Fig S2 from the paper by Hamedpour et al.. [4]

The very first step is scanning these paper-based devices using a scanner. A scanner is preferred over a camera as it operates under fixed conditions and also eliminates the possibility of ambient light effect. Once the images are ready the algorithm starts by processing all the pixels through a low-pass filter. The filter is set in red color channel of the RGB color model, with a threshold value

of 180. All the pixels having red color value lower than 180 are set to 0 and the rest of the pixels are set to 1. This segments the grey boundary regions from the rest of the images. The resultant image is a binary image consisting of either black or white regions.

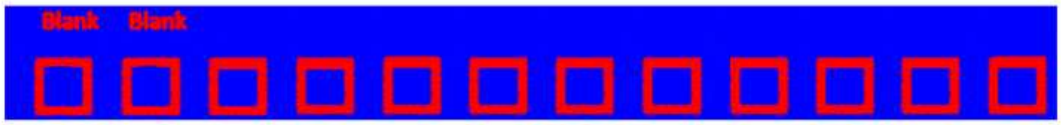


Figure 2.10: Segmentation of boundary region (one row sample) using a low-pass filter in red channel. The area highlighted passes through the filter. This is Fig S3 from the paper by Hamedpour et al.. [4]

The Figure 2.10 shows segmentation from the low-pass filter. The pixels highlighted in red are in the boundary region. These pixels have red color values less than 180. Now the region of interests lie inside these boundary regions. But the region outside the boundary can pose a threat for boundary detection as it contains text printed in the background. This can be seen in Figure 2.8. In the next step the image is converted to a binary image. The pixels identified using low-pass filter have their color value set to 0 , i.e blacked out, and the rest of the pixels have their value set to 1 (blank or white). After that the upper half of each row is also blacked out. This is done by setting the pixel value to 1 for all the pixels in the upper half. This is possible because of the pre-defined layout of the device.

Figure 2.11: Segmentation of boundary region (one row sample) blinding the upper half of each row of square regions. This is Fig S4 from the paper by Hamedpour et al.. [4]

The Figure 2.11 shows the binary image sample generated after the segmentation step where the upper half of the row is blacked out. This helps isolating the regions inside the boundary as well as negating the effect of any noise in the blacked out background. This aids the next

step of region detection inside the grey boundary regions. The image in Figure 2.11 is processed by mathematical morphology recognition (MMR). MMR is a technique of analysing geometrical structures in digital images. The MMR process detects regions of specific shapes. In this case the rectangular and circular regions are of importance. Once these regions are identified, the algorithm looks for the regions closest to the shape and size given as input. This leads to segmentation of all the pixels residing inside the regions of interest.

Once these pixels are identified, their color data can be extracted from the original image. This approach is similar to the one adopted in this body of work. The knowledge of the design being leveraged to segment the boundary regions is a step used in this work as well.

Chapter 3

Problem Statement and Data Description

The goal for this body of work is to automate the data collection process of color data of pixels in the regions of interest for μPAD device images. But to understand what this process is, it is important to know what are μPAD devices, how they work, and what data needs to be collected from the images of these devices. The chemists need color data associated with the pixels in specific regions of interest from the images of these μPAD devices. The problem in this process is that the data collection process is manual demanding a lot of time and manual labor. The focus of this work is to automate the manual process using simple computer vision techniques.

3.1 Microfluidic Paper-based Analytical Device (μPAD)

The origin of these kind of devices starts with litmus paper. Litmus paper is a very simple paper-based device made of specific substances. A litmus paper changes color based on the type of solution it comes in contact with, acidic or basic. The working of the device is pretty simple. A change in color of the paper can give some definitive information. This concept is carried forward as more advances are made in paper-based devices. The μPADs discussed in further sections have very few similarities with a litmus paper. But the concept that color change in paper-based devices can provide valuable information still acts as one of the basis of the device's purpose.

A μPAD is a paper-based device with complex mechanism involved [1]. To best understand how it works, some basic principles should be cleared. The starting step is to know which substances react with each other and a basic understanding of the reaction and its byproducts. Here a substance can be a single element, a chemical compound, specific protein, etc. These substances are mixed together and the chemical reaction observed leads to a color change. Based on this knowledge one of the substance is used to test the presence of the other, and it is called as reagent. The other substance is called as the reactant. This process is done in a controlled environment in a laboratory by researchers and/or experts.

Now these known chemical reactions are used to identify the presence of a specific substance. This can be done by controlling the reagent and the observable result of the chemical reaction can detect the presence of a specific substance as well as measure their concentration. Few examples are; detection of heavy metal ions in test solution [12], detecting presence of lead in river water, detection of pathogens in saliva [2], etc. These chemical reactions can also be performed using paper as a base device. This is an oversimplification of how the technology has progressed with these devices, but it helps in establishing a layman's understanding of such paper-based device. One of the requirements of performing these reactions on a porous surface (paper) is to control the amount of the reagent and the reactant being used, as well as controlling where they are placed and their movement on the device.

The first step is to place a pre-determined reagent on the paper. This is done by baking the reagents onto a paper through various printing processes. As a next step the other reagent in liquid form (test solution) is pipetted onto this reagent. The paper by Ozer et al. [1] reviews the advances in such devices and explains in more detail how they are made. The baked reagents are concentrated in specific regions surrounded by boundaries of different material that helps controlling the movement of test solutions. One such material is wax and the boundaries are printed using wax material. The concentrated regions can be of different shapes to control the flow of the test solution being pipetted.

The Figure 3.1 shows different 3D design of regions of various shapes. Inside these regions reagent(s) are concentrated which react with the test solution. These regions are surrounded by boundaries of different materials controlling the flow of the test solution. The same chemical reactions that can be tested in a laboratory now can be done anywhere using these devices. The goal of using specific reagents is to test for selective substance(s).

Once the reaction is completed on these paper devices, the regions concentrated with reagents may or may not change color depending on the presence of the substance being tested. The color change in these regions is part of the design. The amount of color change observed in these regions helps in determining if a specific substance is present and in what concentration in the test solution.

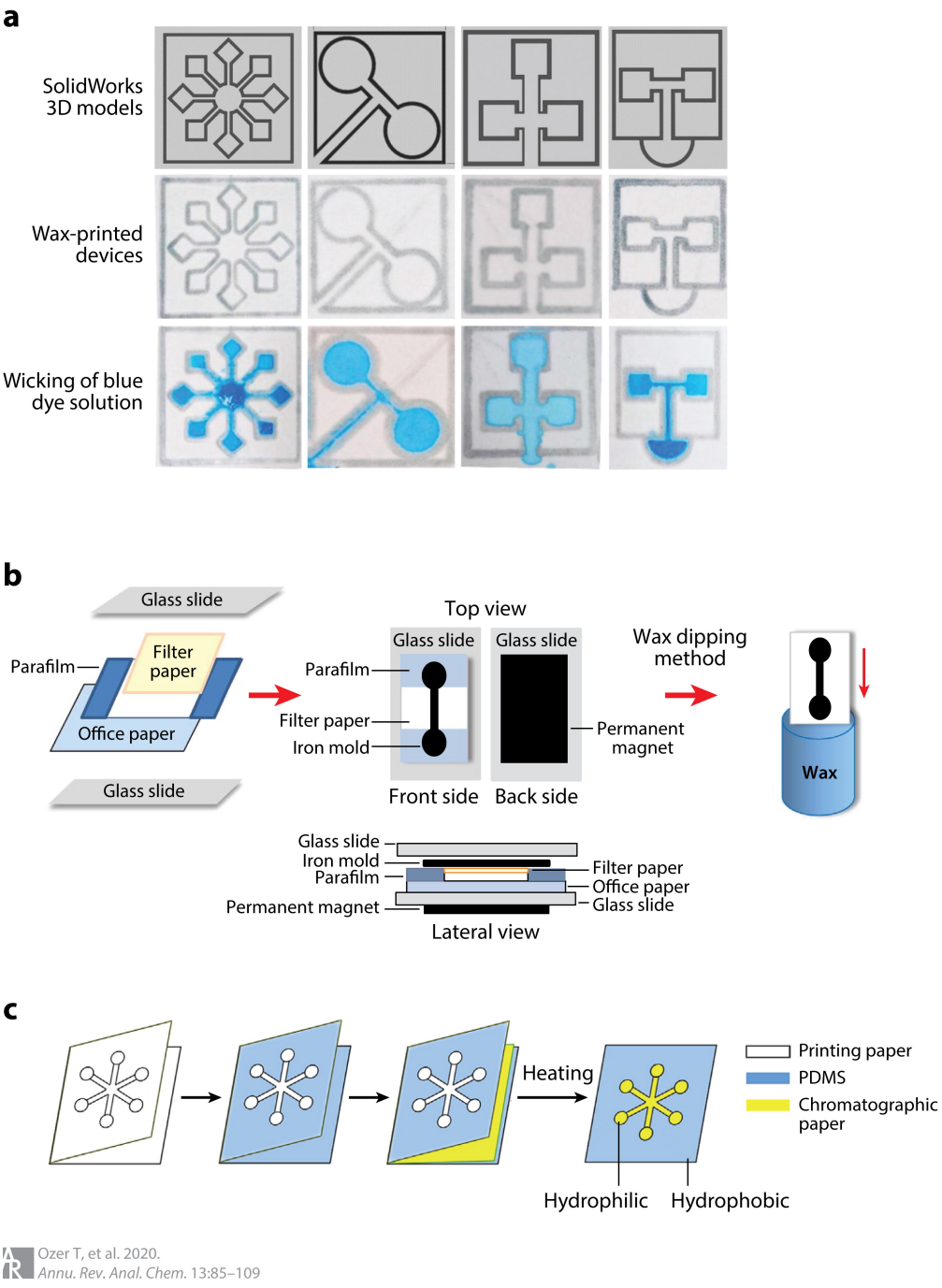


Figure 3.1: 3D design of regions of different shapes where reagents are concentrated surrounded by boundaries controlling flow of the test solution. This is Fig 1 from the paper by Ozer et al.. [1]

3.2 Why use μPADs?

The idea is to move such tests from a laboratory to field through affordable devices. The μPADs present a cheaper option as opposed to these tests being performed in laboratory. It also saves time and labor involved in getting the test results. The costs associated with the setup of a laboratory, which includes expensive devices such as a spectrophotometer. The results associated with these devices can be determined using a control / reference color guide. This guide helps map the intensity of resultant color with the presence and the concentration of the target substance.

This makes these devices usable in field without the need of an expert to interpret the results. The production cost of one device presented in the next subsection is around 1 USD. This is relatively cheaper and a faster option than the traditional methods involving laboratories and experts.

3.3 Devices used for this body of work

There are two types of devices referenced and used for analysis for this body of work. One of these devices is the actual device to be used in the field and the other device is used for experimental data collection.

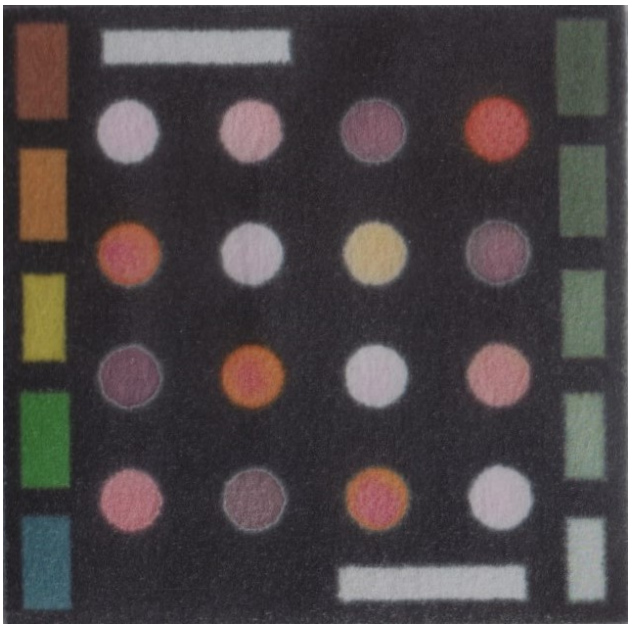


Figure 3.2: A sample of paper-based device which can be used in the field.

The Figure 3.2 shows a sample paper-based device that can be used in the field. The device is designed to have sixteen regions of interest. These regions are circular in shape. This device is blank device, which means it has not been used. There are different regions with different colors, this is so because they have different reagents which are used for testing different substances. The rectangular regions at the top, bottom and near the edges are used for calibration. The device is printed with particular color value in digital format. While capturing an image of the device in field can introduce noise from various factors such as lighting condition, device used for capturing the images, etc. The color values in these regions is compared with the ideal values and the difference accounted with the captured image is used for adjusting the data captured from the regions of interest.

Once the device is used the circular regions may show a change in color. The change in color is then used for detecting the presence and concentration of a target substance.

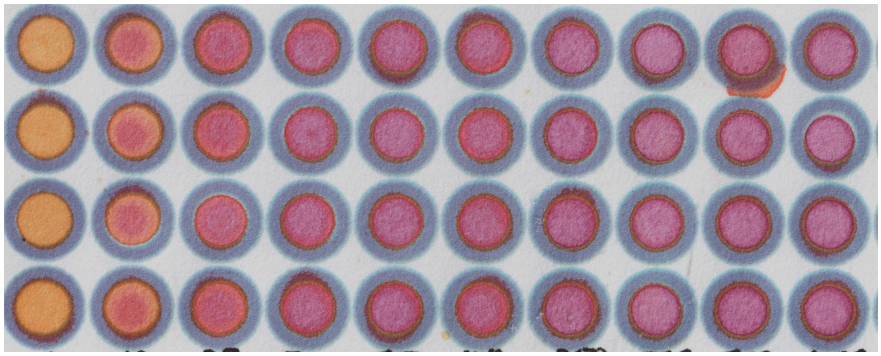


Figure 3.3: A sample of paper-based device which is used for carrying out experiments.

The Figure 3.3 shows a sample paper-based device which is used for experiments. When a device to be used in field is designed, the reagent and its amount (concentration) to be baked in the device must be known beforehand. One of the goals for using the experimental device is to find out that information. The design of this device involves circular regions arranged in a grid like pattern. The number of circular regions can differ in these devices, but the pattern remains the same. Each of these circular regions is surrounded by a greyish region which is made up of wax.

This is to restrict the fluid inside the regions otherwise they might flow into neighbouring regions. Each region in this device has the same reagent and is used for testing the same target substance.

The way this device is used is very interesting. The left most column of regions are tested with a blank solution so it does not show any color change. Moving to the left one column at a time the concentration of the target substance, in this case metal ions, is increased. The color change observed also has increased intensity moving towards left. This helps the researchers in finding the sensitivity of the device. Researchers / chemists can observe what concentration is the color change distinguishable, which is referred as sensitivity in this case.

With the field device there needs to be a reference color guide for mapping the color change with the presence and concentration of the target substance. The data from these experimental devices is compiled to generate the mapping for the reference color guide. The next subsection explains the data collection process in more detail.

3.4 Data collection for regions of interest (ROIs) - The actual problem

Once the device has been used and the ROIs show color change, the color data has to be analyzed. For this purpose the color data in the RGB channel is required for each pixel inside these ROIs. The current process involves processing images of these devices using ImageJ. Using ImageJ each of these regions inside the wax boundary are segmented using a circular shape of fixed radius. This process is completely manual. One such process is outlined in the supplementary material of the paper by Kamnoet et al. [12]. The researchers have to outline each circular region in one device. The number of these circular regions can range from 16 to 96. This takes a lot of precious time away from the researchers which can be better spent on analysis. This also increases the labor cost associated with these paper-based device which affects the affordability of these devices.

The goal is to automate this process of data collection for pixels in the ROIs. This will aid the chemists in reducing the manual labor involved and focus on the data analysis. It is important to understand that this solution should aid the chemists in data collection and not take over the

process of data analysis. This is the reason behind avoiding modern approaches like using machine learning models. This problem can be solved in a deterministic way by using simpler computer vision techniques like image segmentation and region growing. More complex approaches are simply not a suitable solution for this problem.

The goal of this work is to eliminate the manual work involved in the color data collection process for pixels in regions of interest. This problem can be overcome using simple techniques which are discussed in more detail in the next two chapters. The process starts by pre-processing the images for these devices and then identifying the regions of interest.

3.5 Pre-processing

Before the algorithm can start processing the image there are two pre-processing steps necessary for the algorithm to work. The first step is to compile data from the design of the device which is used as a starting point for region identification and segmentation. The is compiled data is referred to as template from here on-wards. The second step is scaling down of the device images. This step is required only for the experimental device and not the field device. The images for these devices are scanned using a scanner which generates high resolution images with least amount of noise. The experimental devices are larger in size generating images of very high resolution. The high resolution leads to high processing time and to avoid this the images are scaled down. The next sub-sections go over these steps in detail.

3.5.1 Template

The idea is to accomplish region identification and segmentation using a region growing algorithm. This is because the regions of interest are distinguishable from the background and boundary regions. But to start with a seed pixel or area is needed. This data can be gathered from the design of the device. The method outlined in this section corresponds to the final stages of the work. The process has evolved to include more data than just any pixel in the regions of interest.

The scanned images of devices from the scanner are loaded in GIMP [1]. A circle shape is defined which surrounds one of the circular regions. Precaution is taken to avoid the coffee rings near the boundaries. The Figure 3.4 shows examples of how this is done in GIMP. Once the region is outlined, the size is fixed. The position seen is the position of the upper-left corner of the bounding square. This helps in calculating the center of this region as well as its radius. This circular region is then moved and superimposed over each and every region in the image. Thus, at the end, the pixel co-ordinates of presumed centers for each region is compiled along with the radius (same for every region). This data is generated from one of the images and is treated as template for all the images of devices with same design / layout. The template is composed of this data.

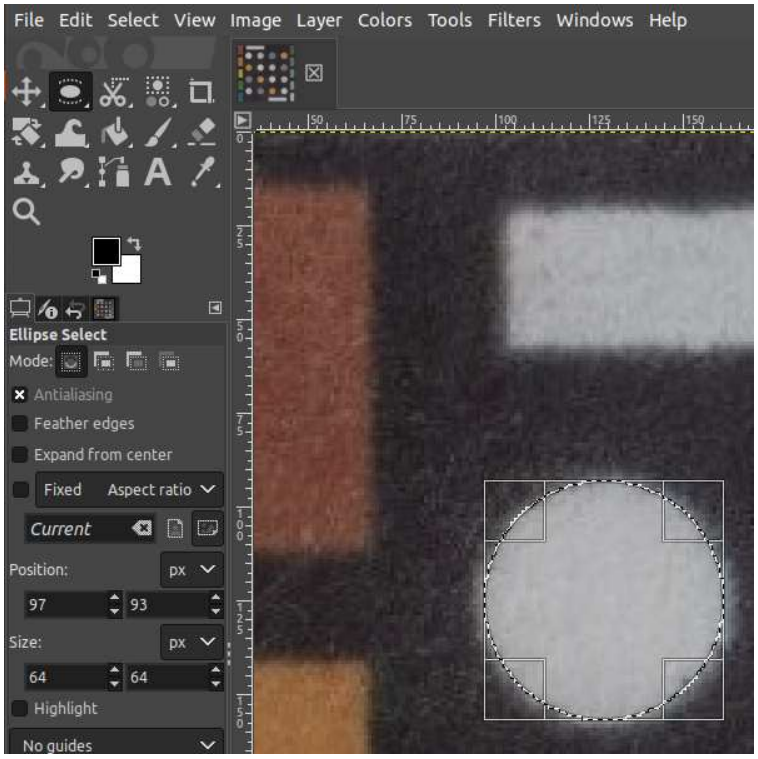
3.5.2 Scaling down the image resolution

From Figure 3.4, the Figure 3.4a shows an circular area having diameter of 64 pixels whereas the Figure 3.4b shown the circular area having a diameter of 300 pixels. The experimental device images have 5 times the resolution of field device images. This results in more time consumption for the algorithm to process any image. This led to scaling down the experimental device images by a factor of 5. For this purpose a median filter is used.

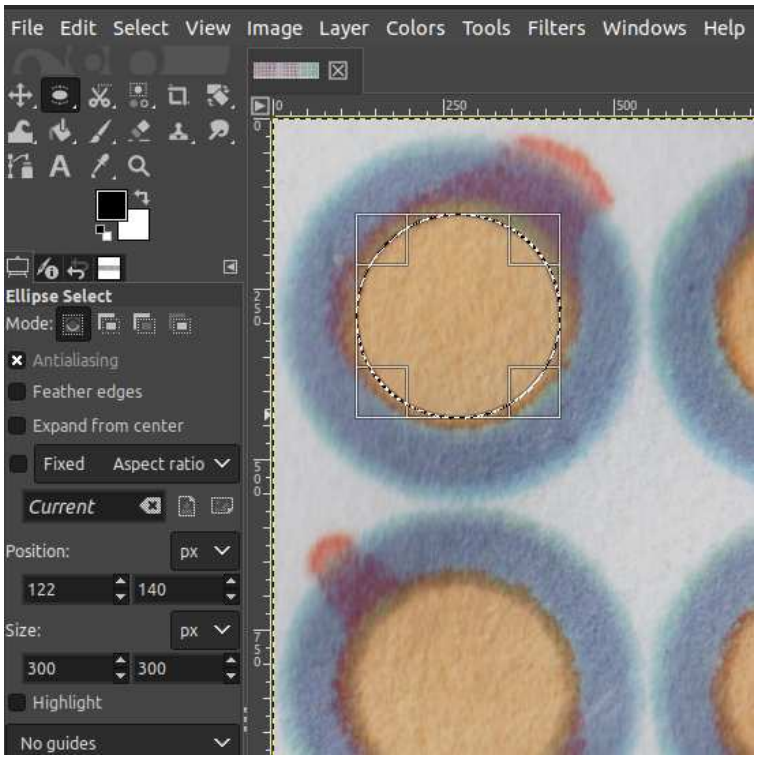
A median filter, as discussed in section 2.1, is usually used for smoothing out images / reducing noise. In this body of work the median filter is used as a method to scale down the resolution. While scaling down an image there are two dimensions to manipulate, width and height. This means that if the number of pixels in an image needs to be reduce a group of pixels will be reduced to just one pixel. The idea is behind median filter is that neighbouring pixels usually have similar color values. Based on this assumption for all the pixels in a square window the median color value is assigned to all of them. This helps in reducing the noise from outlier pixels having very different color values.

The same principle is used for scaling down an image. If a group of pixels are being replaced by just one pixel then the color value of that pixel should be closest to the maximum number of the

[1]GNU Image Manipulation Program - It is an open source software for image manipulation, similar to Adobe Photoshop. https://www.gimp.org/



(a)



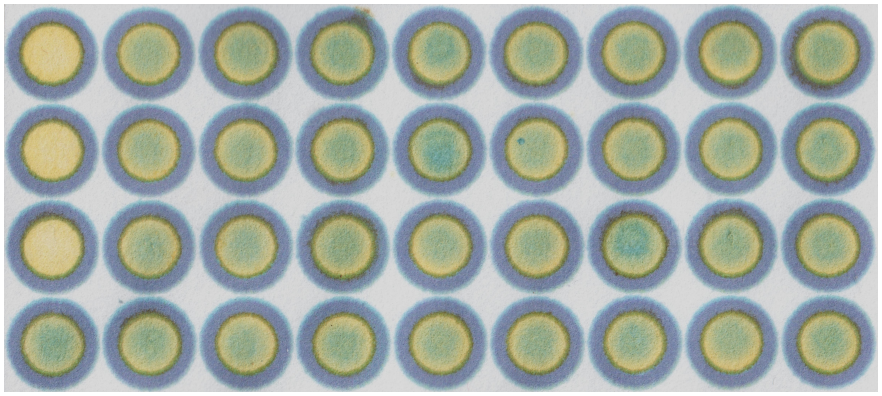
(b)

Figure 3.4: A circular shape superimposed over one region. The software also helps in estimating the position of the center of the region. (a) Field device (b) Experimental device.

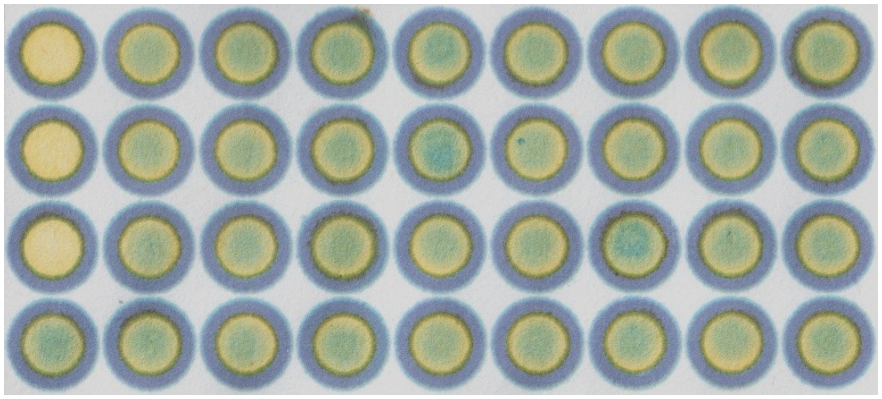
pixels in that region. In median filter process a sliding window is used of size *nxn*. All the pixels in the window are considered as one group and their color values are modified accordingly. Then this window is moved over by one column or one row at a time. The process for scaling down is somewhat similar.

For scaling down the image, a group of pixels are replaced by just one pixel. This means that no windows can have common pixels among them. Each window placement is adjacent to each other and does not overlap. In each window the RGB color values are sorted separately. Then the median value is picked for each color channel and assigned to the single pixel replacing the complete window. This process is repeated till all the pixels from the original image are covered and a new image is rendered. This new scaled down image is very similar to the original image. The Figure 3.5 shows an example of the original and the scaled down image of an experimental device.

This is process was optimized by dividing the images in parts and processing these part images in parallel. This is possible because no two windows have any pixels in common. The image is divided in 5 parts along the width of the image. The same median filter scaling process is then applied on each of these part images in parallel threads. Once processed these part images can be stitched back in the same order to render the complete scaled down image.



(a)



(b)

Figure 3.5: The image (a) is the original crop from the scanner (4966x2441 pixels) and (b) is the scaled down image (993x488 pixels).

Chapter 4

Approach

The overview of the solution seems simple. The first step is to identify the regions of interest. Once those are identified the pixels inside these regions will have the color data needed as the end product. The knowledge of the design of these devices is essential for identification and segmentation of the regions of interest. The layout gives us an estimate of the position of the at least one pixel inside these regions. This idea works on the assumption that the image of the paper-based device is cropped to the edges of the device. For this body of work this assumption is a requirement but it is not a focus point.

There are two devices being used and there are pre-processing steps involved before the algorithm starts processing the image. The next sections go over the different approaches that were experimented. The working algorithm is covered in detail in the next chapter. Although there are two devices associated with this work, the final algorithm focuses only on the experimental device. The initial work had only the field device to work with, but as the research by the chemistry group progressed the focus turned to the experimental devices. The failed approaches first are focused on the field device and later on the experimental device. But the final algorithm has been tested for the experimental devices only.

At this point if the readers are only interested in the final algorithm, then jump to Chapter 5 after Section 3.5.

4.1 Simple Region Growing

This section goes over the approach adopted in the initial stages of the research. The work presented by Carell et al. [3] serves as a base or starting point for this body of work as it was done in collaboration with the same research group. Initially the research focus were the field devices and later the experimental devices. The results on field devices were not according to the requirement and further sections discuss the failures and the requirements in more details. Although these

approaches failed and were not sufficient by themselves as a solution they are important stepping stones for the actual solution.

4.1.1 Region identification with simple region growing

Once the pre-processing steps are completed the image is processed using the template. The pixel co-ordinate is used as a seed pixel to generate a seed region. A square region of *10x10* size is selected with the seed pixel being its center. This region is treated as the seed region. Average color values for reg, green and blue color channels are calculated. Now all the pixels neighbouring the seed region are added to a list. The next step is to process this list and check if any pixels belong to the region. One pixel is removed from this list at a time and checked if it can be added to the region.

Any neighbouring pixel is included in the region if it is similar to the pixels inside the region. To establish similarity the color values of the pixel in question should be close to average color values of the region. The tolerance levels are set to $\pm$ *2.5 times the standard deviation*. The difference between the color values of any pixel being examined and the average color value for the region is calculated. If the differences for each color channel are within the tolerance levels then the pixel is included in the region. If the pixel is not included in the region then it is marked as visited so that it is not processed again.

When any new pixel is included in the region, the average color values are re-calculated. All the neighbours of this newly added pixel are examined based on the following criteria:

- Pixel is not included in the region
- Pixel is not in the neighbouring pixels list
- Pixel is not marked as visited

If the pixel satisfies this criteria the it is added in the neighbouring pixels list.

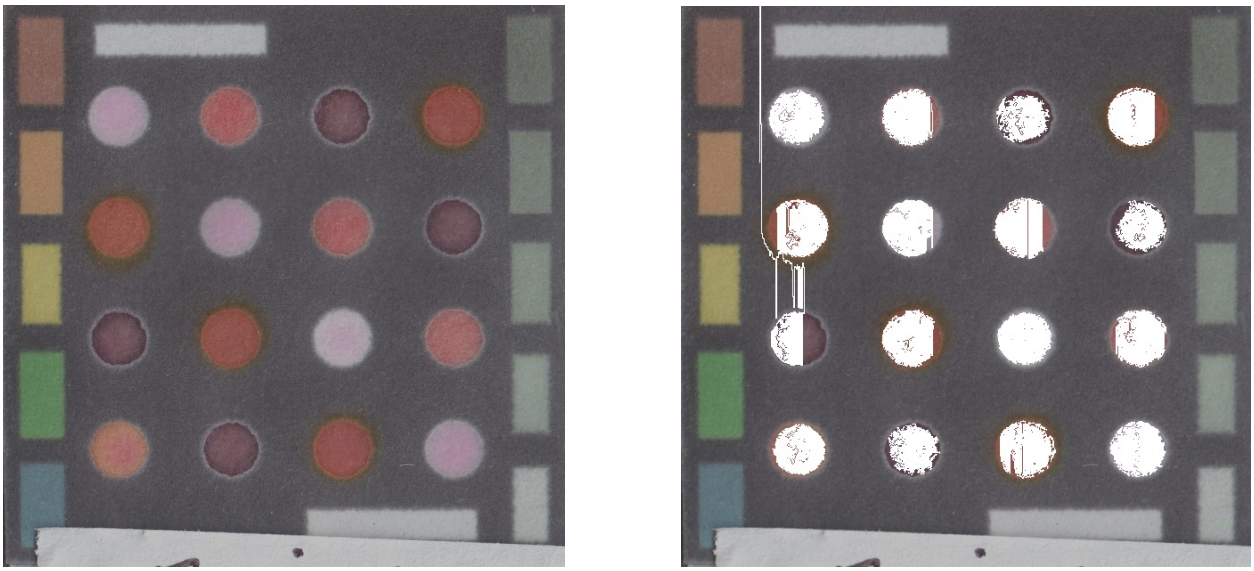
These steps are repeated for the next pixel in the neighbouring pixels list. The terminating condition for the process is that the number of pixels included in the region reaches a value calculated

by the area of circle formula. The radius is provided in the template. For e.g. the field device template has a radius of 30 pixels.

$$Area\ of\ circle = \pi * r^2 \tag{4.1}$$

$$Number\ of\ pixels\ in\ region = \lceil \pi * r^2 \rceil \tag{4.2}$$

The equation (4.2) gives the number of pixels in the region as 2828 for radius of 30 pixels. This approach fails because of lack of control over how the region grows. The region grows randomly and if there is noise in the background, the results are not as expected. The Figure 4.1 shows how this approach fails in Figure 4.1b. The identified regions are superimposed over the original image.



(a) **(b)**

Figure 4.1: A sample image demonstrating the failure of the simple region growing algorithm. (a) Original field device image. (b) Identified regions superimposed on the original image.

Because of noise in the background as well no control over the region growth direction leads to the pixels in the background being included in the region. This also happens because the color values for pixels inside the region and in the background are closer than the tolerance levels.

4.1.2 Region growing along the radius

The failure in the previous approach demonstrated the need to control the direction of region growth. One way to accomplish it is to let the region grow from the center to the boundary gradually. This implies that the pixels closest to the center should be considered first to be included in the region. This means that the regions grows along the radius stating from the center and moving outwards.

To accommodate this change, there needs to be a selection criterion on the pixels chosen from the neighbouring pixels list. Instead of creating a simple list for the neighbouring pixels, a dictionary is created with the Euclidean distance of the pixel from the center being the key. For this to work the pixel from the template is assumed to be the exact center for the region. This assumption is based on the idea that any device image is cropped to the edges and the template will stand true for every image.

$$d(P, C) = \sqrt{(P_x - C_x)^2 + (P_y - C_y)^2} \tag{4.3}$$

The equation (4.3) is used to calculate the euclidean distance between the pixel P and center C. Px, Py, Cx and Cy are x and y co-ordinates for the pixel and center respectively.
The algorithm does not have a lot of changes but an additional step is introduced. The selection of pixel to be checked for including in the region is not random. The steps to create the dictionary of the neighbouring pixels are as follows:

- Calculate the Euclidean distance between the pixel in question and the seed pixel
- The Euclidean distance is the key for the dictionary. Check if the key already exists:

- If the key does not exist create a key-value pair where the value is a list. Add the pixel in question to the list and add the pair to the dictionary.
- If the key exists then append the pixel to the list associated with the key.

When a pixel is being selected to be checked from the dictionary, the key with the least value is first selected. The list of pixels associated with this key is fetched and any pixel from this list can be selected. The selection is done on first in first out basis. This ensures that all the pixels with the same distance from the center are checked first.

This modification fixes the issue of controlling how the regions grows. The Figure 4.2 shows how this approach fails as well. One if the assumptions for the modification is that the seed pixel are assumed to be the exact centers of the regions of interest. Evidently that is not the case. InFigure 4.2 the superimposed regions in the first column are relatively better than in the next columns. In the rightmost column the superimposed regions are aligned towards right. This is how

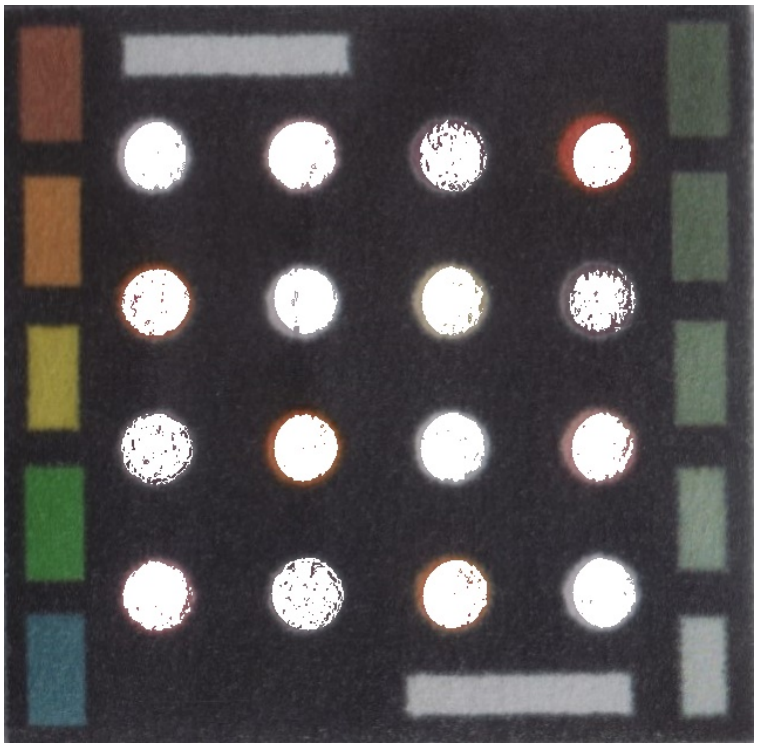


Figure 4.2: Sample image showing algorithm failing after introducing control measures for direction of region growth

this modification fails to reach the goal of identifying the complete region. This failure is because the assumed center for each region does not align with the assumed center from the template.

4.1.3 Recurring Region growing

Since the previous approach failed because of the wrong assumption, the natural solution is to remove the assumption. Removing the assumption leaves us with an incomplete algorithm as we do not have exact center for every region. This implies that the algorithm should find the exact centers.

The Figure 4.2 image shows that the seed pixel is still inside the region and close to the exact center. The following steps are added to rectify the assumption of the seed pixel being the actual center.

1. The region growing algorithm is completed and add all the pixels identified to be inside the region are added to a list
2. The center of the identified region is calculated by averaging the x and y co-ordinates of all the pixels inside the region
3. This new center is compared with the seed pixel
    - If they are same then the process stops
    - If they are not the same then
        - Set the new center as seed pixel
        - Run the region growing algorithm again
        - If the region with new center has more pixels then this pixel is marked as the center and the process from Step 2 is repeated
        - If the region has fewer pixels than the previous region then the process stops with the previous center as the actual center

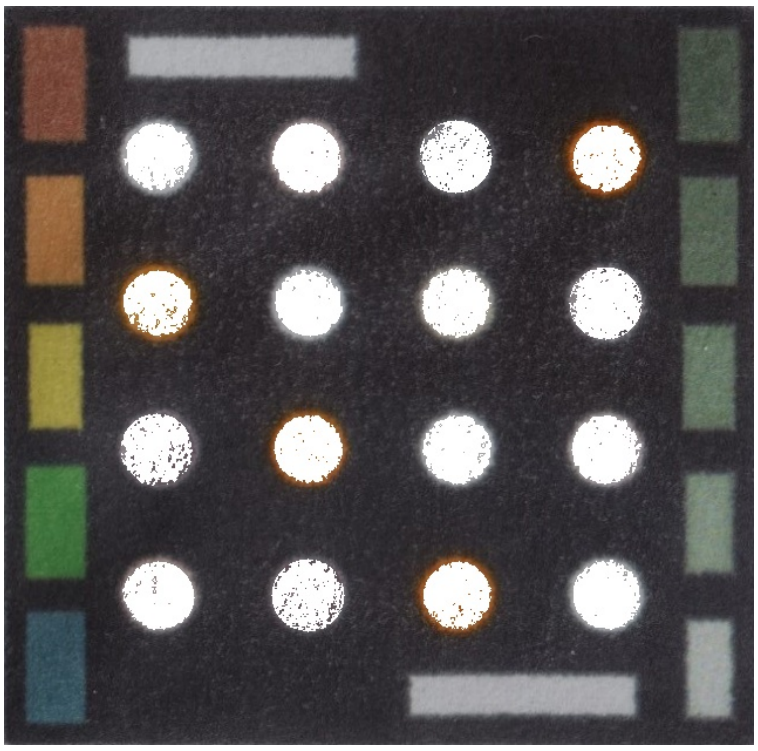


Figure 4.3: Sample image showing superimposed regions generated by the recurring region growing algorithm

The Figure 4.3 shows that with this modification the superimposed regions are much better. This modification has few drawbacks. The processing time for any image can increase exponentially. It requires more memory. Depending on how the image is captured for a device it might not be a feasible approach. The biggest drawback of this approach is that the superimposed regions are not uniform. There are pixels being excluded where gaps and holes are observed.

At this point the feedback from the chemistry research group was vital. For the data analysis to be accurate the identified regions should include all the pixels starting from the center of the region till the boundary where background region starts. This requirement rendered this approach unfeasible for the field device. Another issue at this point was that the focus of the chemistry research group shifted from the field devices to the experimental devices. The next chapter focuses on the algorithm that works on the experimental device and the approach that eventually led to it.

Chapter 5

A More Robust Algorithm

5.1 Change in approach

The issues plaguing the previous approach demanded a deeper analysis of the device images. As our focus shifted from field devices to experimental devices, the next steps was to analyze the images of those devices. A visual analysis of the experimental device images shows that there are three separate regions in the image. The white background, they wax boundary region, and the regions of interest inside the wax boundaries. The histograms of HSV color channels were analyzed.

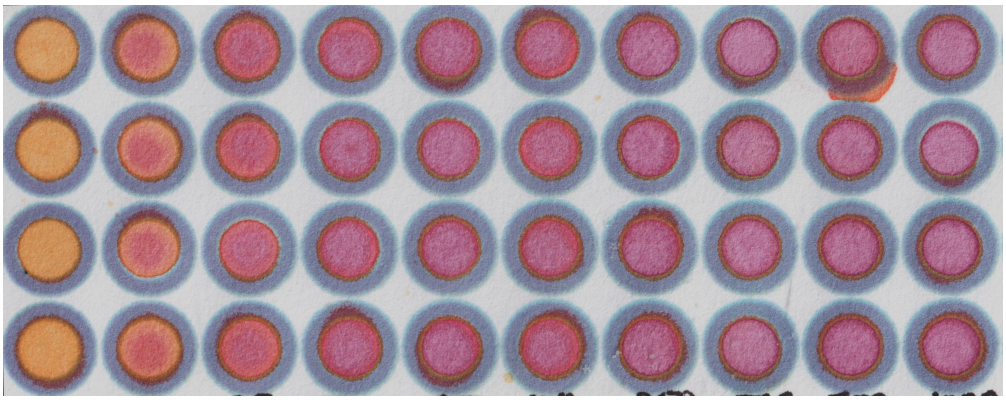


Figure 5.1: Image of the experimental device used for generating the histograms of HSV color channels.

The image shown in Figure 5.1 was used for generating the histograms. Only one image was used as the design pattern is similar for all the device. So the knowledge from this analysis could be applicable to all the images.

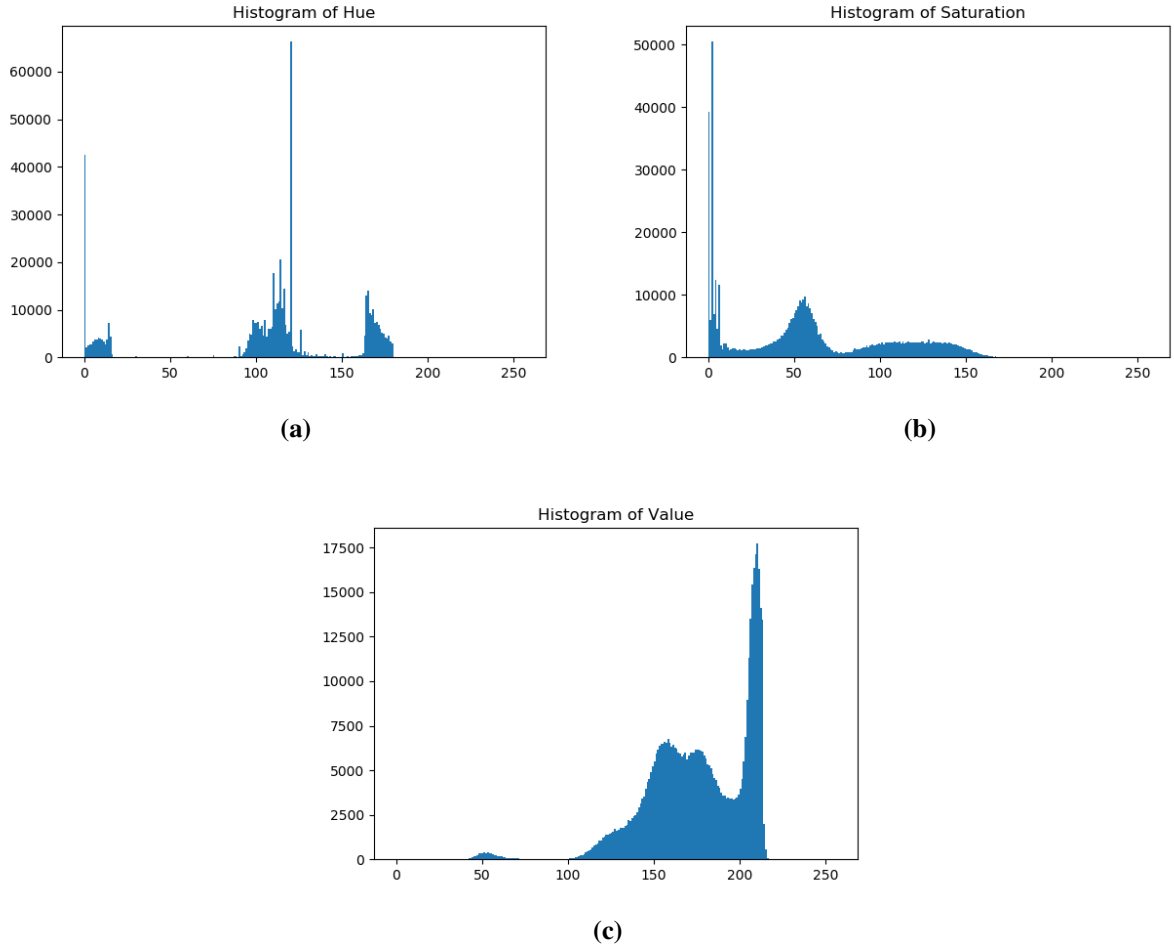
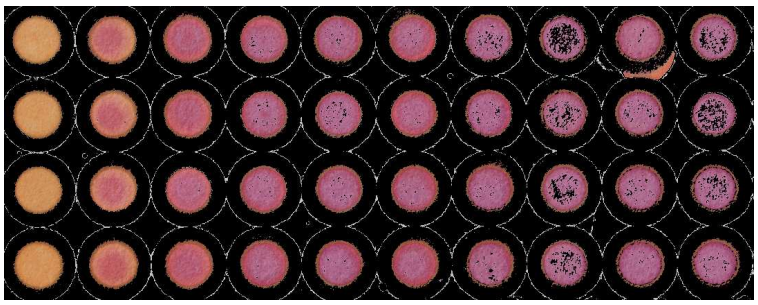


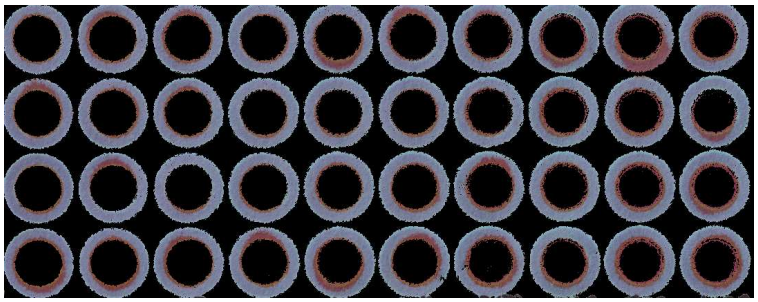
Figure 5.2: Histograms of (a)Hue (b)Saturation and (c)Value color channels respectively

The Figure 5.2 shows the histogram generated from the image in Figure 5.1. The histograms of Hue and Saturation channels show trimodal distribution. This is an important observation as this distribution helps in identification and segmentation of the regions of interest according to the requirements. On further analysis of the image, it is possible to segment different regions using simple filters.

There are two ways the segmentation of different regions can be done. One way is to segment the regions of interest themselves. The other option is to segment the wax boundary regions which surround the regions of interest. Both of these option scan then be further used to calculate exact center of the regions of interest. The Figure 5.3 shows how the segmented regions would look like.



(a)

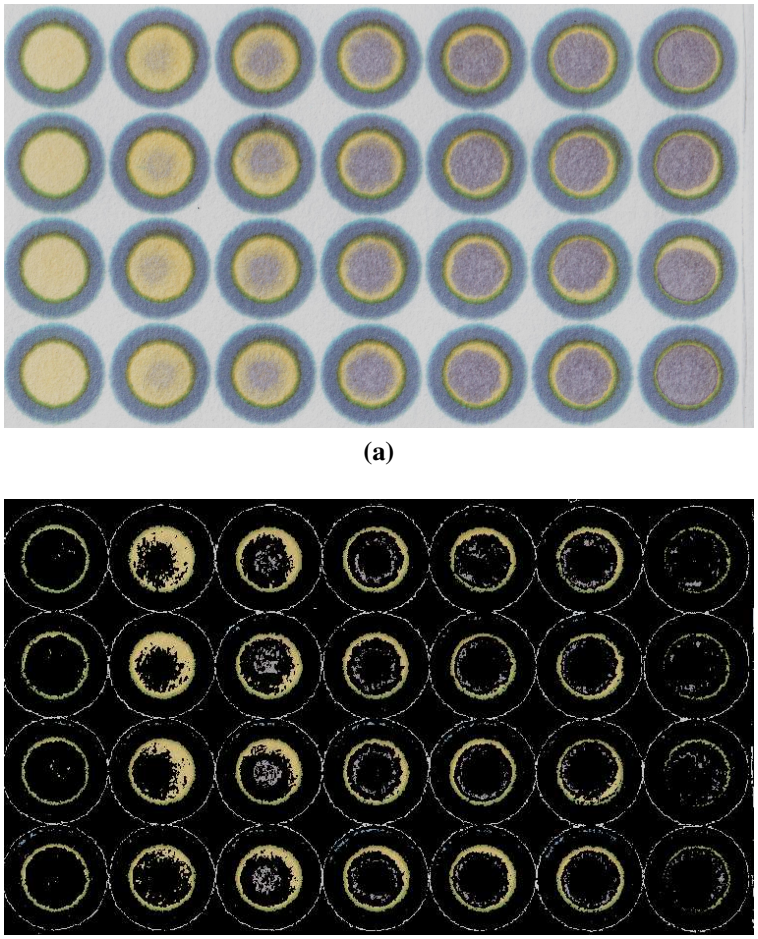


(b)

Figure 5.3: (a) The regions of interest segmented from the background and boundary regions (b) The boundary regions segmented from the rest of the image

Figure 5.3a shows how the regions of interest segmented from the rest of the images looks. This can be done by filtering the pixels based on the values of Saturation and Value channel. If pixels have a Hue value less than 130 and the Value channel value between 0 and 8 or between 16 and 85, then these pixels are rejected. If any pixel is rejected then the color values for it are set 0 for all three color channel. This makes the pixel look black. The Figure 5.3b shows the wax boundary region segmented from the rest of the image. This can be done by filtering the pixels based on the Red channel color values. If the Red color value of any pixel is greater than 145 then it is rejected.

Both the options were experimented with but the second option turned out to be more feasible. There are variety of reagents used in the region of interest. Interestingly for some of the devices, once the reaction was complete the color inside the region at higher concentration (read the regions in the columns near the right edge) was very close to the color of the wax boundary regions. This makes the segmentation process very difficult as demonstrated in Figure 5.4.



(a)

(b)

Figure 5.4: Sample image showing color of region of interest close to the color of wax boundary region and failure to segment regions of interest

The algorithm incorporates measures to take care of such cases with the second option of segmenting the wax boundary regions. The important point to note here that the multi-band thresholds required for filtering the boundary region have to be identified manually. This body of work has not identified to automatically detect the thresholds required for the segmentation process.

5.2 Final algorithm

The final algorithm is layered approach with the goal of finding the exact centers of the regions of interest. The process starts by identifying the thresholds required to segment wax boundary regions from the rest of the image. The next steps involve using a simple region growing algorithm using the template data to find the exact center. Once exact centers are identified a simple circular region can be set with a set radius from the template.

5.2.1 Identifying thresholds for segmenting boundary regions

This step needs manual analysis as it takes some trial and error to find the exact thresholds for segmentation of boundary regions as discussed in Section 5.1. The threshold(s) may or may not include range filters for multiple channels. For the image in Figure 5.3 the only threshold required is in red color channel. But the image in Figure 5.4 requires more complex thresholds for boundary region segmentation. The Figure 5.5 shows the segmentation of boundary regions for the same image. The segmentation requires a combination of filters using all three color channels.

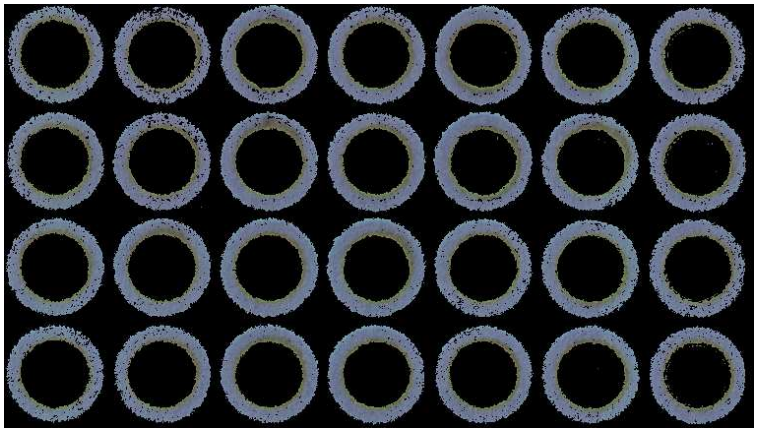


Figure 5.5: Region segmentation for the image in Figure 5.4

The following threshold combination is used for the above shown segmentation. The color values of any pixel are set to 0 if one of the following two conditions are true:

- If the red color value of any pixel is greater than 125

- If the red color value of any pixel is greater than the green color value and the blue color value for the pixel is greater than 110

An important observation is that same threshold combination does not work for each and every image. The images of devices having similar reagents as well as similar color change have a high chance of one threshold combination working for all of them.

5.2.2 Pre-processing

Once a threshold combination is identified, the next step is to generate the template for a group of images. Unlike the threshold combination, the template has to be generated only once for any image consisting the same number of regions of interest in the same layout. Another part in pre-processing of any image is to scale it down. Template generation and scaling down of an image is discussed in detail in Section 3.5.

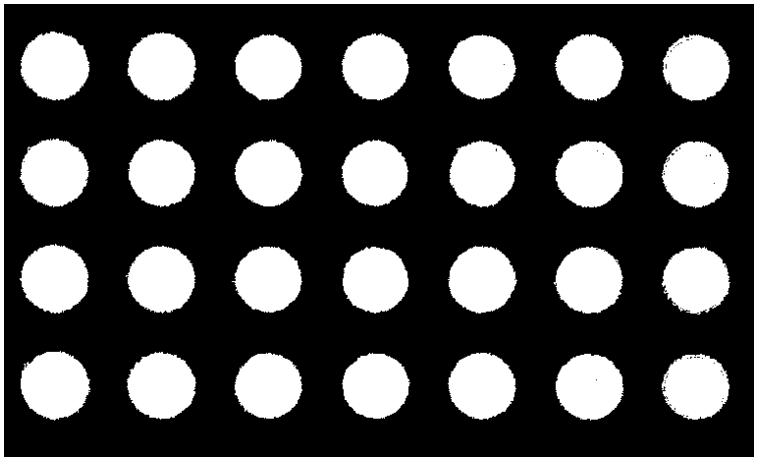
5.2.3 Region growing for calculating exact centers

Once the template is ready and the image is scaled down, the next step is to find the exact centers for each region of interest. Calculating the exact centers of the regions of interest is done by the following steps:

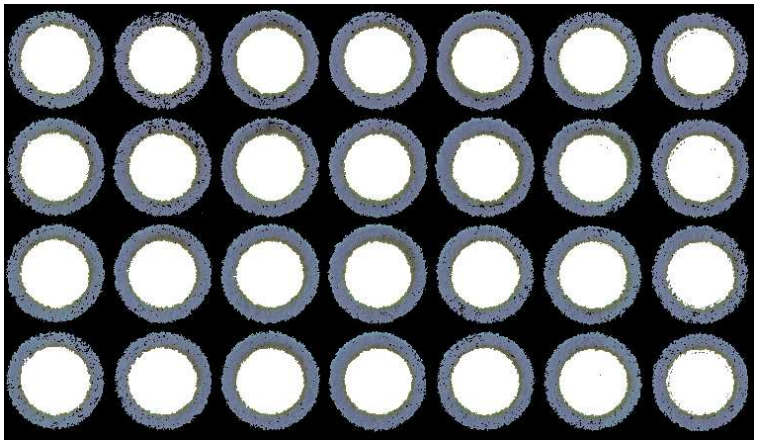
1. Since the image is scaled down by a factor of 5, the seed pixel co-ordinates and the radius also need to be adjusted. This is done by dividing co-ordinates value and the radius value by 5.

2. Segment the boundary regions by using the threshold combination identified in Section 5.2.1. The image seen in Figure 5.5 is the output generated.

3. Use the template and the image generated in the previous step as inputs. The seed pixel corresponding to each region is used to grow a region inside the segmented boundary regions. The criteria for region growth is that all the color values of the pixel should be 0.



(a)



(b)

Figure 5.6: Images showing grown regions inside the boundary region.

The Figure 5.6b shows the grown regions restricted inside segmented boundary regions. The Figure 5.6a shows the same grown regions without the boundary.

4. Once the regions are identified, new centers for each region are calculated. The x and y co-ordinate values of all the pixels belonging to each grown region are averaged. These averaged values are the co-ordinates for the center of these regions.

5. Now that the exact centers of each region are known, the next step is to collect data for the regions of interest. The radius provided with the template reaches close to the boundary. The radius is reduced in length by 3 units (amounts to 3 pixels). This step is taken to ensure that the data being collected does not include the data for the coffee ring near the boundaries.

6. The new radius and the centers for each region are used to compile a list of all the pixels whose color data is to be collected from the original image. This is not a region growing step. The only criteria for a pixel to be selected is its distance from the exact center of the region the pixel lies in. Figure 5.7 shows superimposed white regions. The color data is collected from the pixels in these regions.

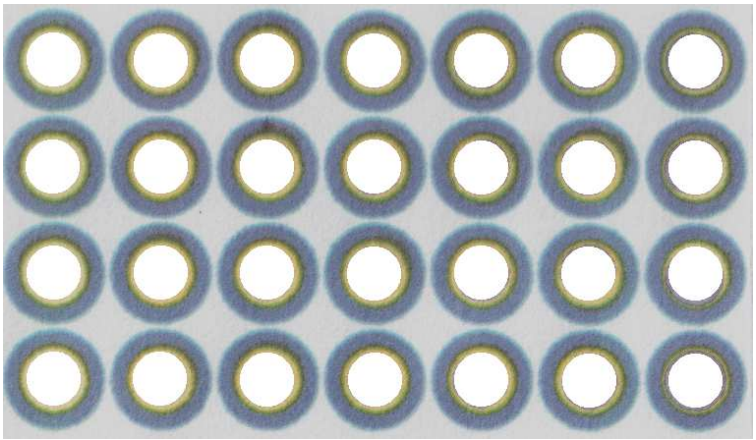


Figure 5.7: The superimposed white region highlights the pixels for which the color data will be collected.

7. The color data is nothing but the BGR and HSV color values associated with a pixel. This data is then compiled in a CSV file, where the the color data is labelled with the pixel co-ordinates and the region it belongs to.

CSV File

Table 5.1: Table showing a sample of the generated CSV file.

Region ID	Region Shape	Y	X	B	G	R	H	S	V
1	circle	57	51	205	207	208	20	4	208
2	circle	166	58	212	209	211	140	4	212
3	circle	280	52	209	204	206	132	6	209

Table 5.1 shows a sample of the generated CSV file. The details in each row includes the 'Region Id' which is identified by counting the regions in the order down and then across. This means that for the device image shown in Figure 5.7, a region with id 6 is the region in the second column and second row. The region shape is specified in the next column. The 'Y' and the 'X' column contains the y and x co-ordinates of the pixels in the region specified by the 'Region Id'. The next six columns specify the channel values of blue ('B'), green ('G'), red ('R'), hue ('H'), saturation ('S') and value ('V') channels for the pixel at co-ordinates specified by 'Y' and 'X' columns.

Chapter 6

Result and Future Work

6.1 Result

The algorithm worked well enough on a handful of test images. The next step was to quantify how well it worked over a large dataset. For this purpose the accuracy of the algorithm needs to be measured. The accuracy in percentage can be calculated by comparing the data generated by the algorithm with the ground truth data. The ground truth in this case is the data generated by finding the center for each image manually. This center is then used to identify all the pixels inside a circular region of the same radius (28 pixel length) set for the data generated algorithm data. The next sections go over how the ground truth data was collected and how it was compared to arrive at the accuracy percentage value.

6.1.1 Ground Truth

This process is the same discussed in Section 3.5.1. The difference being that this process is repeated for each and every image. A list of centers is compiled for every region. This list is then used to generate a CSV file similar to what the algorithm generates. This data is considered as the ground truth for calculating accuracy percentage.

6.1.2 Accuracy

The accuracy for this algorithm is measured by comparing the CSV files generated by the algorithm with the ground truth data. The comparison is done for each region in any image. The data files are parsed to generate a list of all the pixels in a region. One list is generated from the ground truth data and another from the algorithm generated data. These list contains the co-ordinates of all the pixels belonging to one region. The next step is to calculate the intersection of the two lists. The intersection is all the pixels (same co-ordinates) that are present in both the lists. The pixels in the intersection are counted and are then divided by number of pixels in the list of

pixels from the ground truth data. This value multiplied by 100 gives the accuracy percentage for one region (Eq.(6.1)).

$$Accuracy = \frac{N_I}{N_{GT}} \times 100 \tag{6.1}$$

Where:

N_I: Number of pixels in the intersection of ground truth and algorithm generated data for one region.

N_{GT}: Number of pixels in the list generated from ground truth data for one region.

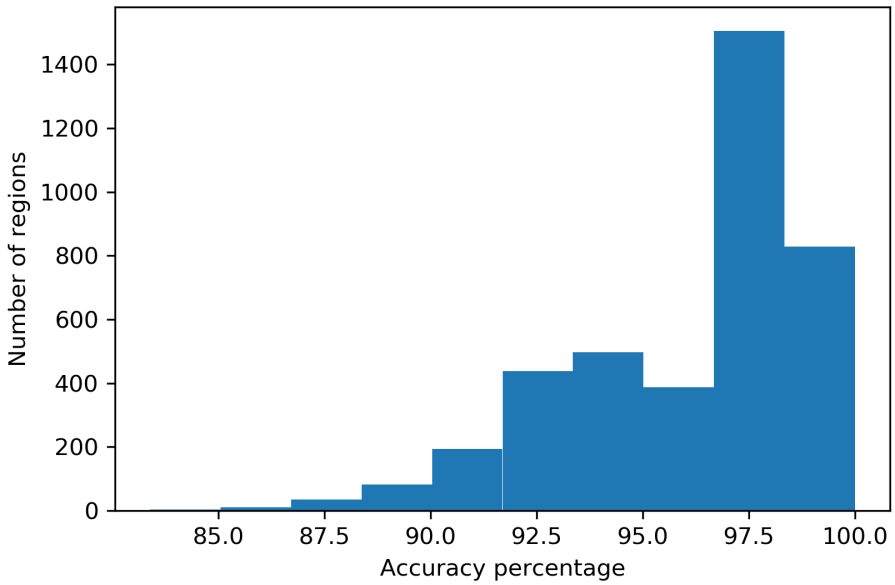


Figure 6.1: Histogram of accuracy percentage of regions from all the images

The Figure 6.1 shows the histogram of accuracy of regions from every image. The peak near 97% percentage accuracy shows that the algorithm works very well.

The algorithm takes 30 seconds to process an image of resolution 3865x2534 pixels. This tie can be further improved as there are avenues of optimization in the algorithm which have not been explored yet.

6.2 Future Work

The algorithm has been refined to use simple techniques like median filter scaling, thresholding and region growing in a layered manner and it works very well. The implementation for this algorithm does not include any optimization or parallelization for region growing or thresholding. Introducing parallel processing in these steps will improve the processing time of the algorithm.

Apart from reducing the processing time, there is another avenue of improvement in the algorithm. Before even any image can be processed, the thresholds in color channel(s) need to be identified. This step is still done manually. More research is needed to fins possible ways of automating this step.

The implementation of this algorithm is done in python. But this can also be implemented as a plugin for open source tool like ImageJ. This is a very good next goal as ImageJ is used widely by chemists and a plugin with this tool will be easily reachable to a large number of users.

Bibliography

[1] Tugba Ozer, Catherine McMahon, and Charles S. Henry. Advances in paper-based analytical devices. *Annual Review of Analytical Chemistry*, 13(1):85–109, 2020. PMID: 31986055.

[2] Cody Carrell, Alyssa Kava, Michael Nguyen, Ruth Menger, Zarina Munshi, Zachary Call, Mark Nussbaum, and Charles Henry. Beyond the lateral flow assay: A review of paper-based microfluidics. *Microelectronic Engineering*, 206:45–54, 2019.

[3] Cody S. Carrell, Rachel M. Wydallis, Mridula Bontha, Katherine E. Boehle, J. Ross Beveridge, Brian J. Geiss, and Charles S. Henry. Rotary manifold for automating a paper-based salmonella immunoassay. *RSC Adv.*, 9:29078–29086, 2019.

[4] Vahid Hamedpour, Paolo Oliveri, Cristina Malegori, and Tsuyoshi Minami. Development of a morphological color image processing algorithm for paper-based analytical devices. *Sensors and Actuators B: Chemical*, 322:128571, 2020.

[5] T. Huang, G. Yang, and G. Tang. A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(1):13–18, 1979.

[6] R. Adams and L. Bischof. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647, 1994.

[7] S. A. Hojjatoleslami and J. Kittler. Region growing: a new approach. *IEEE Transactions on Image Processing*, 7(7):1079–1084, 1998.

[8] Steven W. Zucker. Region growing: Childhood and adolescence. *Computer Graphics and Image Processing*, 5(3):382–399, 1976.

[9] N. Pham, A. Morrison, J. Schwock, S. Aviel-Ronen, V. Iakovlev, Ming-Sound Tsao, James C. Ho, and D. Hedley. Quantitative image analysis of immunohistochemical stains using a cmyk color model. *Diagnostic Pathology*, 2:8 – 8, 2007.

[10] Curtis T. Rueden, Johannes Schindelin, Mark C. Hiner, Barry E. DeZonia, Alison E. Walter, Ellen T. Arena, and Kevin W. Eliceiri. Imagej2: Imagej for the next generation of scientific image data. *BMC Bioinformatics*, 18(1):529, Nov 2017.

[11] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[12] Pornphimon Kamnoet, Wanlapa Aeungmaitrepirom, Ruth F. Menger, and Charles S. Henry. Highly selective simultaneous determination of cu(ii), co(ii), ni(ii), hg(ii), and mn(ii) in water samples using microfluidic paper-based analytical devices. *Analyst*, pages –, 2021.