Colonial Academic Alliance Undergraduate Research Journal

Volume 5 Article 2

2015

CSI-Pi: A novel automated secure solution to interpret on-site colorimetric tests

Subrata Acharya Dr.
Towson University, sacharya@towson.edu

Saurabh Khatiwada
Towson University, skhati264@gmail.com

Kelly M. Elkins
Towson University, kmelkins@towson.edu

CSI-Pi: A novel automated secure solution to interpret on-site colorimetric tests

Cover Page Note

This research has been supported by the School of Emerging Technologies, Towson University.

CSI-Pi: A novel automated secure solution to interpret on-site colorimetric tests

Abstract

Illicit drug evidence constitutes a vast majority of chemical evidence collected from crime scenes. However, determining which drug is seized is not a trivial task as most are white powders. Since their introduction, colorimetric chemical detection tests, also referred to as *presumptive drug tests* based on their tentative determination of unknown substances, aid in the *on-scene* differentiation of drug material with a rapid color change within 1-2 minutes. These colorimetric tests are an important tool used in crime scene investigation and for obtaining a search warrant to find illegal drug labs and drug distributors. However, both positive and negative color interpretation is often reported differently depending upon the user and two analysts may describe the same color differently, e.g., "*brilliant greenish blue*" vs. "*strong greenish blue*". Moreover, the high rate of false positives and working color memory limit the effectiveness of these manual tests. To this effect, this research reduces the subjective interpretation and reporting with regard to color in these tests by offering users with a new platform/technology in the form of a Raspberry Pi (standalone) application that *"reads"* the color of the presumptive drug test, searches and matches the color using a pre-built library database, and reports accuracy (%) matches for further laboratory evaluation.

KEYWORDS: colorimetric; secure; CSI; Raspberry Pi; interpretation

1. Background & Introduction

According to the Uniform Crime Reports released by the *Federal Bureau of Investigation (FBI),* the highest ranking crime arrests in the United States over the past 8 years consistently involve drug abuse offenses. In 2013, a total of 11,302,102 arrests were made, of which 1,501,043 arrests were due to drug abuse offenses [1]. This is roughly an astonishing 13% of all crimes committed nationally. This significant trend follows across state statistics as well. In the same year, Maryland saw 49,250 drug related arrests among 185,422 total crime incidents and Virginia, out of its 403,314 total crime incidents, saw 57,502 drug and narcotic offenses [2-3]. The task of identifying illicit drug content involved within crime scenes is a challenging task for forensic investigators. While some opt to conduct all tests within a lab setting, a more common approach is to conduct on-site colorimetric drug tests to determine potential drug content in suspected powdery substances to obtain a warrant for further search and/or arrest.

A colorimetric drug test, or a presumptive drug test, is a preliminary examination that determines whether a strong possibility of a given white substance chemical mixture containing traces of a particular drug, in a relatively pure form, exists. The chemical reagents within these tests are varying custom color reagent kits that react differently to various drugs often within 1 - 2 minutes [4].These tests are an important tool used in crime scene investigation for obtaining search warrants and finding illegal drug labs and drug distributors [5]. Once a substance is added to the reagent test kit mix, the presence of a drug is reflected by a change in the solution color that is accurately identified by a forensic investigator and interpreted either through memory, color test manuals, individual reagent label swatches, or color swatch books such as the Munsell Book of Color [6]. A confirmatory test is later conducted off-site in a lab setting to verify the results of the colorimetric test. These manual processes of identification as well as interpretation are two critical stages that determine the successful outcome of a colorimetric test.

The rapid execution of these tests compromise slight accuracy for speed that could potentially introduce false positive elements. The past decade alone has seen multiple court cases where everyday chemicals and substances react to the color solution in a manner similar to that of an illegal drug. One such incident involves one Janet Lee who was arrested in the Philadelphia International Airport because she was suspected of carrying cocaine when, in actuality, it was flour. The court cased that followed resulted in financial ramifications that cost the city $180,000 [5]. Incidents such as these have sparked debates among members of the scientific community who have begun questioning the validity and necessity of a test that could potentially yield wrongful arrests [7]. The increase in the variety of drugs found in crime scenes directly impact how interpretations are reported as well. What one might consider to be a "*brilliant greenish blue*", someone else might label "*strong greenish blue*" [8]. The potential impacts of these limitations, such as longer investigations and false arrests, necessitate that the colorimetric drug test process be devoid of ambiguity.

To that effect, this research aims to carefully redesign the process of colorimetric drug testing interpretation through automated technology to improve upon the existing barriers of possible false positives, subjective color interpretation, and time consumption. This is achieved by pairing a portable computing technology platform, such as the Raspberry Pi B+, with custom designed software that connects to a database attached server capable of securely automating the processing of the interpretation logic. The software essentially “reads” the color of the test using a spectrometer or a commercial off-the-shelf app with high accuracy, uses a prebuilt library database

[9] to search and match the color, and reports accuracy percent matches using a server to complement further laboratory evaluation.

The idea of using technology to enhance colorimetric drug testing is rather recent and has only surfaced in the past few years. One such attempt showed the use of the smart phones as a portable means to save data after conducting color tests [10]. The speed and reliability of mobile devices make ideal platforms for information storage and retrieval. The use of the cameras on modern smart phones as well as high-end digital cameras to capture the solution color change is also a tried method [11].

Images captured through cameras and smart phones are edited using powerful image editing software, such as Adobe Photoshop, that can identify the color through the RGB (Red, Green, and Blue) value of the solution [12]. A RGB value is a digital representation of a given color capable of denoting over 16 million unique definitions. These 16 million colors can represent every color through the 256 levels of each of the Red, Green, and Blue values [13]. Hence, a computer recognizes the range for each of these values with 0 being the lowest intensity and 255 the highest. Computer monitors and smart phones make use of this method of digital color representation to display graphical pictures and images on monitors and display screens.

However, the scope of both types of the aforementioned attempts lends itself at a surface level where the application itself does not lend a hand in the logic processing; both mechanisms simply function as a way to take input and print output. This project improves upon the technology integration component by incorporating the logical interpretation of color tests through technology aided automation.

The use of a spectrometer provides significant benefits over alternatives such as digital cameras and is a focal point of this project. These devices measure the intensity of light that passes through a chemical mixture to determine changes in color. They used to be very costly and were limited to being available in only a lab setting due to their sizes. However, a miniature spectrometer, such as the Qmini spectrometer, can be utilized for achieving a similar test with high portability [14]. It provides connectivity to hand-held peripherals through USB 2.0 ports and GPIO pins. Since it uses infrared signals to retrieve input, the spectrometer is an ideal choice for on-site drug testing as it is unaffected by disturbances in light or noise [15]. It is a highly stable device capable of detecting the accurate color of a chemical mixture and integrates seamlessly with hand-held analysis devices such as the Raspberry Pi B+.

2. Platforms and Technologies

Using a specialized, standalone system such as the Raspberry Pi B+ device designed to handle a specific task provides an inherently strong application security. A Raspberry Pi B+, as depicted in Figure 1, is an affordable, credit-card sized computer produced by the Raspberry Pi Foundation capable of running a Linux based operating system, such as the *Debian Linux* variant titled "*Raspbian"*. It is often used by students and enthusiasts alike to learn programming and get involved with creative projects such as weather stations, surveillance systems, gaming consoles and more [16-17]. The remarkable small system, when paired with external peripherals such as a battery pack and a display screen, is capable of providing excellent portable functionality for the process of field drug testing. The security and long term stability benefits that the Linux Kernel provides strong software security and reliability. Hence, this platform has practical application in

the context of this project and this research emphasizes the use of a Raspberry Pi B+ in the implementation of the revised colorimetric drug testing process.

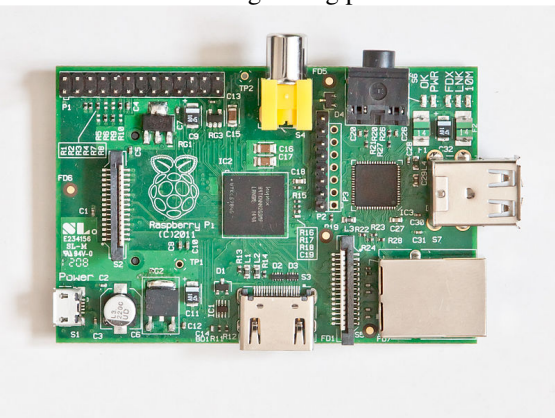

Figure 1 - Raspberry Pi B+ Model. This image illustrates the computing platform implemented in this research [17].

Alternative frameworks to the Raspberry Pi that have the potential to be used in this project are the remarkably popular Arduino boards such as the Arduino Board Uno (Figure 2). This particular device is a portable, microcontroller board capable of interacting effectively with a wide variety of external peripherals without the set up overhead [18]. It is a popular choice among enthusiasts who make use of external motors and sensors to implement various projects. Despite the simplicity and relative ease of use of such a device, the Raspberry Pi B+ is a comparatively more powerful platform with better integrated hardware support for peripherals such as keyboard and mouse, a network adapter, and significantly higher RAM size [16]. While the Arduino is performs well for single process tasks such as reading LED lights, the Raspberry Pi has the capability to extend this functionality by allowing interfaces such as HDMI display screens to be easily added and customized for bigger and more complicated projects.

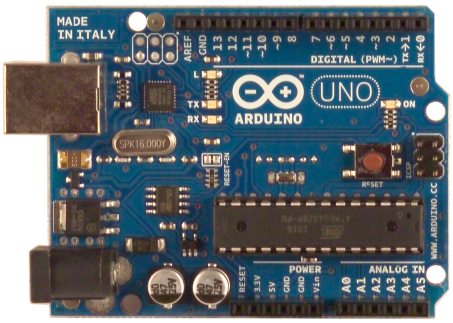

Figure 2 - Arduino Board Uno. An alternative to the Raspberry Pi framework [18].

The BeagleBone Black is a yet another potential substitute that stands alongside the Raspberry Pi B+ (Figure 3). For a slightly higher cost, it provides a higher computing power with a more powerful processor, increased RAM size, and compatibility with full Linux distributions such as Ubuntu Linux and Debian Linux [19]. Thus, being fully compatible with the spectrometer and containing native support for USB input and display screens, it is on par with what the Raspberry Pi B+ provides for the purposes of this project. However, it lacks support for an HDMI based

screen output which is an important element of the software application [16]. Since the application is designed with a complete Graphical User Interface (GUI), the capability of an integrated high quality display gives the Raspberry Pi a slight edge. While both platforms have the appropriate capability of performing quite well, the Raspberry Pi's cheaper cost makes it an excellent choice as well.

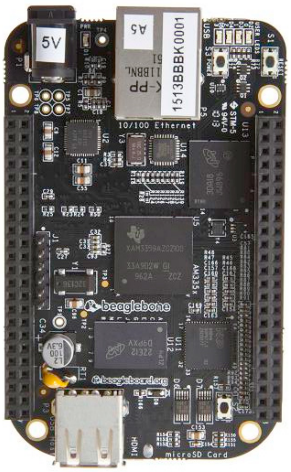

Figure 3 - BeagleBone Black. Potential alternative computing platform for this project [19].

3. Design and Implementation

This project solution will allow an investigator to effectively administer a colorimetric drug test at the scene of a crime and almost immediately be able to identify the drug at hand with high accuracy. As Figure 4 suggests, the setup consists of the Raspberry Pi connected to a 7" Mini HDMI display screen and runs from a 10,000 milliamp USB-compatible Battery Pack. These elements make the overall setup portable and thus, suitable for field use. A Micro SD card is used to store the custom designed software as well as the operating system information and the Wi-Fi Dongle provides simplified network connectivity for information sharing with the server. The setup also includes the spectrometer and the wireless keyboard / touchpad as USB peripherals to facilitate input. Due to the customizability of Linux, the setup is designed to boot straight into the application so that it is always running. By limiting the user from accessing parts of the system other than the software, we can eliminate potential user based errors from affecting the operating system. The visible screens of each of the stages of the application, depicted in Figure 7, 8, and 9, are custom designed to fit well with the specific 7 inch display screen dimensions for accommodation and visual appeal.

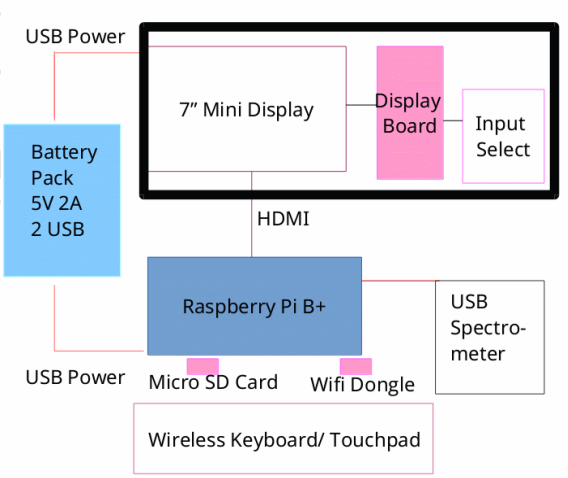

Figure 4 - The Raspberry Pi B+ Setup with Peripherals. Combining these elements within a single encasing allows for a fully portable setup.

3.1 Network and Communication

Moving from the software application, the web server is responsible for the processing of data and determining the identity of a suspected material. Hence, the overarching picture of how the application, server, and database are interconnected is depicted in Figure 5.

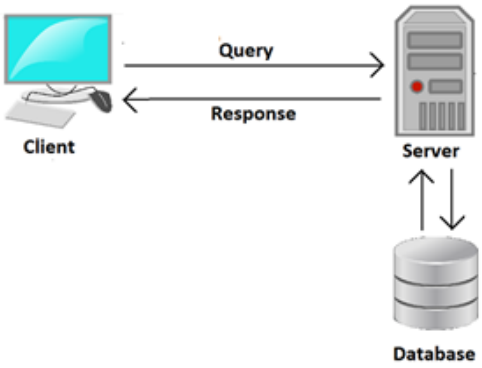

Figure 5 - An overview of the Server, Client, and Database structures. This image depicts the relationship between the three components

The application developed for the client uses the SOAP (Simple Object Access Protocol) protocol to facilitate this communication. SOAP employs the Web Services Description Language (WSDL) file which explicitly defines how the Web service functions. SOAP relies on XML (Extensible Markup Language) to return information to the client. Since XML is rather verbose, parsing of data is often met with a slight time overhead. While the alternative REST (Representative State Transfer) architectural style web services minimize the issue of bandwidth use and parsing overhead, SOAP provides a standardized platform capable of integrating with potential extensions, such as WS-Security, which greatly help enhance overall reliability and security [20]. All communication takes place over HyperText Transfer Protocol Secure (HTTPS) which provides data encryption through the Secure Socket Layer (SSL) cryptography protocol [21].

By designing the application to store all data directly on a secure server, the possibility of data compromise is vastly reduced in cases of data corruption, physical damages to the device itself, or unauthorized access to the application data or the operating system. In the case that internet connectivity is not immediately available, the application inserts the submitted information in a running thread process which continuously attempts to send data to the server regardless of whether the application itself has closed or a new drug test is in process until data transfer is complete.

3.2 Database Server

The server that uses the database to store pre-built drug information and individual drug test is designed using ASP.NET in Microsoft Visual Studio Professional 2013 platform. The function that the software application implements requires the type of test conducted as well as three integer inputs, corresponding to each of the RGB values, and calculates the closest drug matches using an algorithm. Another such function stores the client interpretation information in the database. The database these functions utilize are created using Microsoft SQL Server 2008 due to its table and data management features as well as its compatibility with Excel Spreadsheets. The Excel compatibility was a critical factor in the deciding of the platform as the sample test data being collected was being stored as a Microsoft Excel document. Hence, SQL Server 2008 greatly simplified the data import and management process for populating the database.

The database itself comprises of two distinct tables- "*DrugSubstances*" and "*Events*". The "*DrugSubstances*" table stores actual drug test type and color combination data. It also accommodates for extendibility by allowing for the storage of individual wavelength and absorption in the database, as depicted in Figure 6. On the other hand, the "*Events*" table stores the individual drug test event in conjunction with the appropriate timestamp and the query output. The "Events" table allows for the application to directly store the message as a record in this table. This message is retrieved from the "*Interpretation*" stage from the client through a PUSH based connection [22]. The application sends data to the server that is awaiting a service request for returning interpretation results or storing individual search events.

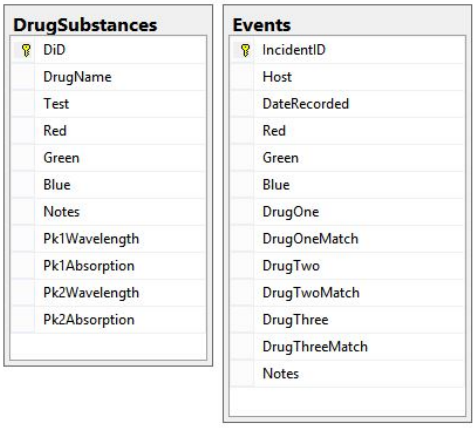

Figure 6 - Color Test Database Tables Layout. This diagram shows the two tables used by the application for information retrieval and storage.

3.2.1 Database Schema

Table 1 shows the raw sample data obtained on the corresponding color test reagent kits and drug substances [9]. The *"Test"* column denotes the type of reagent test kit, the *"Drug"* column identifies the drug standard, and the corresponding RGB values in the R, G, and B columns that would yield the specific drug listed in the *"Drug"* column. The "*Replicates*" column denotes an ID that corresponds to the individual number of test conducted to ensure accurate data recording. The values are observed using the "*Color Assist"* app method which essentially notes that the values were recorded without having a camera light on. Each of the test values are averaged to find the most accurate RGB color value for the appropriate drug substance which is stored on the database.

Table 1 Drug Test Data Collected. This table shows the information that was collected in a lab setting where only the positive RGB values are averaged and stored in the pre-build library database.

	A	B	C	D	E	F	G
2	Test	Drug	Replica				Col
3	Marquis	Morphine Sulfate	1	13	12	14	
4	Marquis	Morphine Sulfate	2	7	4	7	
5	Marquis	Morphine Sulfate	3	6	5	7	
7	Ferric Sulfate	Morphine Sulfate	1	57	56	51	
8	Ferric Sulfate	Morphine Sulfate	2	55	52	48	
9	Ferric Sulfate	Morphine Sulfate	3	68	59	49	
11	Nitric Acid	Morphine Sulfate	1	115	42	9	
12	Nitric Acid	Morphine Sulfate	2	102	24	4	
13	Nitric Acid	Morphine Sulfate	3	90	24	4	
15	Marquis	Codeine	1	4	3	5	
16	Marquis	Codeine	2	4	3	5	
17	Marquis	Codeine	3	7	6	8	
19	Ferric Sulfate	Codeine	1	81	59	33	
20	Ferric Sulfate	Codeine	2	70	58	39	
21	Ferric Sulfate	Codeine	3	81	58	22	
23	Nitric Acid	Codeine	1	117	107	47	
24	Nitric Acid	Codeine	2	111	102	47	
25	Nitric Acid	Codeine	3	93	88	45	
27	Marquis	Diacetylmorphine HCl m	1	5	4	6	
28	Marquis	Diacetylmorphine HCl m	2	4	3	5	
29	Marquis	Diacetylmorphine HCl m	3	5	3	6	
31	Ferric Sulfate	Diacetylmorphine HCl m	1	104	92	72	
32	Ferric Sulfate	Diacetylmorphine HCl m	2	72	61	45	
33	Ferric Sulfate	Diacetylmorphine HCl m	3	81	68	55	
35	Nitric Acid	Diacetylmorphine HCl m	1	104	91	75	
36	Nitric Acid	Diacetylmorphine HCl m	2	88	80	63	
37	Nitric Acid	Diacetylmorphine HCl m	3	108	98	85	
39	Wagner	Cocaine HCl	1	18	6	7	
40	Wagner	Cocaine HCl	2	16	5	6	
41	Wagner	Cocaine HCl	3	14	15	6	
43	Cobalt Thiocyani	Cocaine HCl	1	6	35	61	

3.3 Algorithmic Methods

Independent of the platforms and peripherals lays the core algorithm that features the logic portion of the colorimetric drug test evaluation. The pre-built library contains records of color tests and their color information, stored as an RGB value, as well as the corresponding drug matches that

allows for an accurate drug match comparison platform without data skewing [9]. The modular design of the server code allows for the algorithm to be easily modified depending on the type of evaluation that needs to be conducted.

The linear approach relied upon the implementation of a distance vector algorithm which reliably determined how a RGB combination matches against entry already present in the library database. It can be easily extended to be modified using a weight or a priority based algorithm if the data needs to be presented within the scope of those dimensions. The standard Euclidean distance vector algorithm can determine relevant drug matches in percentages relative to the actual drug information in the database. Given two RGB values, each a pair of three integers $R_1 = [a_1, b_1, c_1]$ and $R_2 = [a_2, b_2, c_2]$, the distance,D, between the two three-dimensional points is calculated using the distance vector formula [23]:

$$D = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2 + (c_1 - c_2)^2}$$

Alternative algorithms that could be implemented for a different form of drug match evaluation involve algorithms in the context of grouping data sets. One potential alternative to the Euclidean distance vector algorithm lies in a potential cluster based algorithm. In such algorithms, data points are essentially grouped into clusters for evaluation that consists of more than one sample [24]. This can prove to be very useful in situations where multiple potential drug packets are found in a crime scene and evaluation needs to be conducted for each instance.

Alternatively, the Naive Bayes algorithm is also applicable in this project. The Naive Bayes algorithm is a classification theorem, derived from the Bayes' statistical theorem, that generates probabilistic matches by the counting the frequency and combination of values [25]. Such an algorithm can prove useful accurate drug matches between the queried RGB and the corresponding data stored in the prebuild library. A self-similarity based algorithm could also be implemented as the core algorithm. Such an algorithm follows the fundamental idea of identifying objects that look relatively the same within a common context [26]. Hence, the process of color interpretation could be greatly facilitated by such an algorithm. In other words, by dividing the RGB information into similarly responding subsets, the algorithm can provide data set analysis on a number of elements throughout the drug matching process.

3.4 Lab Interface

The software application also accommodates a lab setting that allows a forensic investigator to view and analyze each individual drug test incident along with field investigator notes for further evaluation in real time. This allows for the confirmatory tests to take place in the lab to conduct follow-up procedures. As the investigator records more data that is uploaded to the server and the database, it is synchronized to all instances of the application. Thus, an investigator at any given lab can access this incident data in real time to conduct more tests as necessary and track input records. The implementation of the application accommodates for consistent read, write and append functionality among the on-site and off-site users without discrepancies in concurrent data manipulation. In other words, the synchronization of data in the *"Events"* table in the database is implemented such that only one user can update or delete a particular record at a given time.

4. Evaluation

The front end client software that runs on the Raspberry Pi is designed and developed using Java Development Kit (JDK) 8 (Figure 7). This interface is how a client, such as a forensic investigator, is able to interact with the spectrometer and the server to make use of the functionality provided. Since the Raspberry Pi runs a modified version of Linux capable of executing the Java Virtual Machine (JVM) and the Java Runtime Environment (JRE), this platform independence element makes Java the ideal choice for this project as it can even be utilized as a desktop application compatible with other platforms, such as Microsoft Windows or Apple OS X, that support JVM.

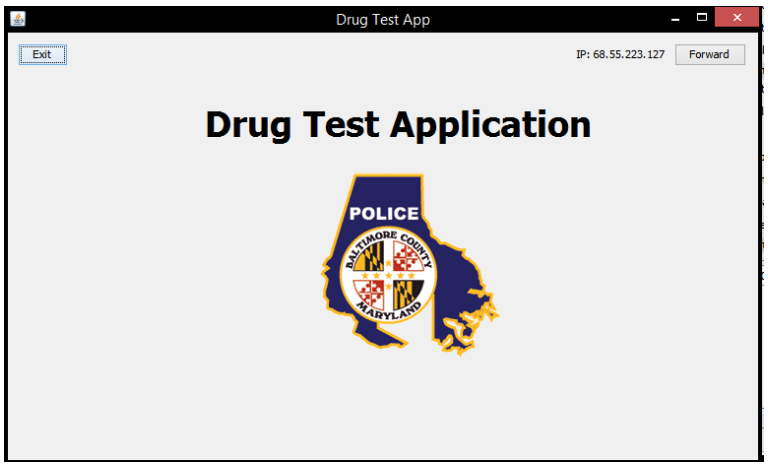

Figure 7 - Software Application Welcome Screen. This is the first screen seen when the program is launched and is programmed using the Java Programming Language.

The front end software developed for server interaction in the field setting can be divided into essentially two distinct phases-identification and interpretation. In the "*Identification*" (Figure 8) phase, the user enters necessary information about the type of test and the results of the test conducted. The results can be collected using two methods-manual or automatic. The user has the option of manually setting the RGB values using data collected using Color Assist in the corresponding textboxes and value sliders automating the data entry by using the "Scan" button option. This button utilizes the spectrometer to gather input. Once the spectrometer is ready, the investigator can point the spectrometer towards the color solution and retrieve the precise digital RGB value from a color solution or a third party application. This automated step is a critical element addressed by this research as it prevents ambiguity in color recording, identification, and reporting while vastly improving on overall investigation time.

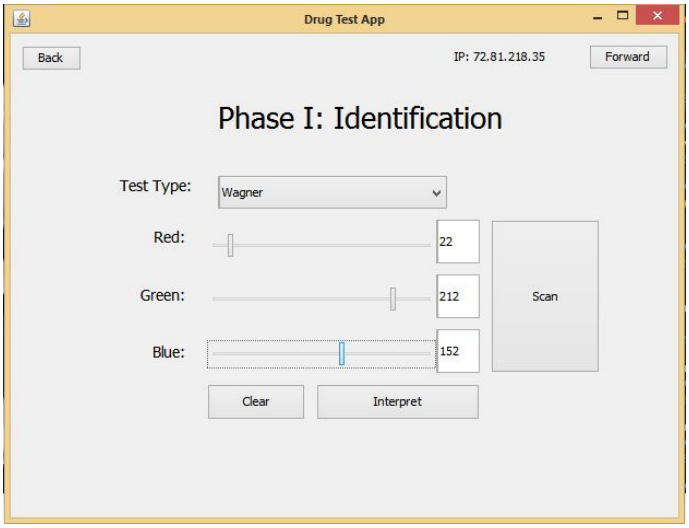

Figure 8 - Identification Phase Screen. In this phase, the user selects a test and either manually enters RGB information (using the textbox or slider objects) or scans directly using the spectrometer.

Once the user attempts to moves to the next phase, the program sends information to the server through a network connection established as soon as the application was launched. The information that is sent to the server includes the color test type and RGB color values for interpretation. The server considers the three dimensional RGB values, matches them against the actual value in the pre-built database using the algorithm and determines the top 10 closest matches as percentage values for each of the colors. The server then proceeds to return the drug match information back to the client device and the user is taken to the "*Interpretation*" (Figure 9) screen where they can see up to ten drug matches and their corresponding RGB value match. The screen also shows the actual color of the spectrometer output independent of the server out in a small textbox element. This is a very critical aspect of the phase as it allows the forensic analyst to have the last say in whether or not the color on the screen is in fact the color observed. This is where making use of the HDMI screen comes into play as ensuring this color is presented accurately in the display screen assists the investigator in making the judgment as to the validity of the results. Should any discrepancies exist, the analyst can optionally make a note of that issue or concern in the corresponding *"Notes"* section. This piece of information is stored with the search for future off-site evaluation. When the analyst submits the search result, the drug match information, along with actual color match, analyst notes, client device identifying hostname and the time as well as date is sent to the database to be stored in the "Incidents" table as a log entry.

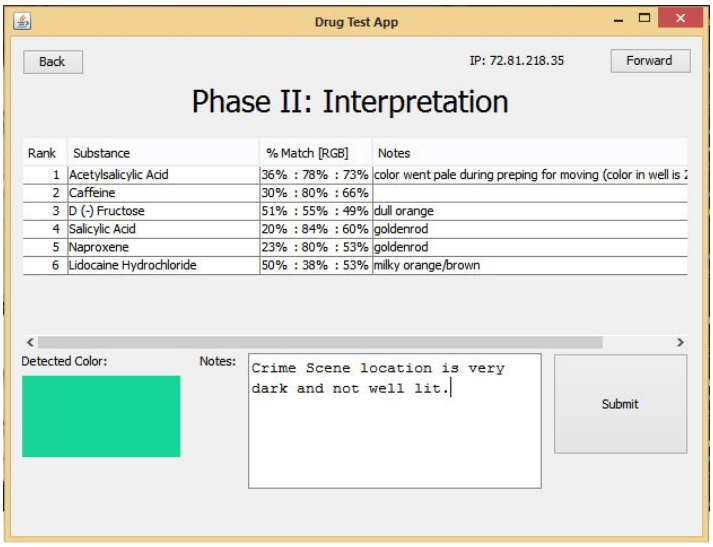

Figure 9 - Interpretation Phase Screen. This screen shows the output formatted into a table based on rankings as well as a sample "Detected Color" RGB interpretation for manual color match confirmation.

5. Conclusions

This research improves the existing methods of colorimetric tests by decreasing the time taken for a forensic analyst to manually map the color test output, improve accuracy in the drug identification process, determine the potential drug involved and record the corresponding result. Depending on the numbers of tests to be run or matches to be made, the application greatly improves upon the time factor by automating the data recording and interpretation process. The Linux based operating system strongly benefits from free software updates backed by the power of a globally active open-source community. With thousands of developers around the world constantly analyzing and improving the source code as well as addressing new bugs, it is considered to be a very secure operating system. The custom application designed for the Raspberry Pi will be released as open-source software as well.

The Raspberry Pi, thus, is a viable option with sound practical use in the context of portable colorimetric drug testing. It efficiently achieves the goal of infusing a technological backend for colorimetric drug test automation using a secure online web server as well as a physically connected, accurate spectrometer. While there are other alternatives could be considered, the Raspberry Pi consistently stands out in areas of cost, performance, and overall compatibility. Combined with the elements of cost and security, it stands to be a valid tool to aid in the automatic processing of a colorimetric presumptive drug test.

6. References

1. Uniform Crime Reports: Persons Arrested. (2014). Retrieved May 29, 2015, from http://www.fbi.gov/about-us/cjis/ucr/crime-in-the-u.s/2013/crime-in-the-u.s.-2013/persons-arrested/persons-arrested
2. Maryland State Police. (2014). *Crime in Maryland 2013 Uniform Crime Report*.
3. Department of State Police (2014). *Crime in Virginia 2013.*
4. Houghton, R. (2009). Appendix: Drug Confirmation Testing. In *Field confirmation testing for suspicious substances*. Boca Raton, FL: CRC Press.
5. Kelly, J. (2008). *False Positives Equal False Justice*. Retrieved May 29, 2015, from http://www.cacj.org/documents/SF_Crime_Lab/Studies__Misc_Materials/FalsePositives.pdf
6. U.S. Department of Justice, National Institute of Education Sciences. (n.d.). *Color test reagents/kits for preliminary identification of drugs of abuse* (Report No. NIJ Standard-0604.01). Retrieved from https://www.ncjrs.gov/pdffiles1/nij/183258.pdf
7. Harris, A. (2011). A test of a different color: the limited value of presumptive field drug tests and why that value demands their exclusion from trial. *Southwestern Law Review*, *40,* 531.
8. Wolchover, N. (2012, June 29). *Your Color Red Really Could Be My Blue*. Retrieved May 29, 2015, from http://www.livescience.com/21275-color-red-blue-scientists.html
9. Elkins, K. M; Weghorst, A.; Acharya, S. Color Quantitation for Chemical Spot Tests for a Controlled Substances Presumptive Test Database, Drug Testing and Analysis, 2015, (under review).
10. Choodum, A., Parabun, K., Klawach, N., Daeid, N. N., Kanatharana, P., & Wongniramaikul, W. (2014). Real time quantitative colourimetric test for methamphetamine detection using digital and mobile phone technology. *Forensic Science International*, *235*8-13. doi:10.1016/j.forsciint.2013.11.018
11. Choodum, A., & Daeid, N. (2011). Digital image-based colourimetric tests for amphetamine and methylamphetamine. *Drug Testing and Analysis*, *3*(5), 277-282.
12. Choodum, A., & Daeid, N. (2011). Rapid and semi-quantitative presumptive tests for opiate drugs. *Talanta*, *86*, 284-292.
13. Numbers (Level 5). (n.d.). Retrieved from http://www.rit-mcsl.org/fairchild/WhyIsColor/Questions/4-5.html
14. Qmini Spectrometers. (n.d.). Retrieved from http://www.rgb-laser.com/content_products/product_qmini.html
15. Infrared (IR) Definition. (n.d.). Retrieved from http://www.nchearingloss.org/ir.htm?fromncshhh
16. Raspberry Pi FAQs - Frequently Asked Questions. (n.d.). Retrieved from https://www.raspberrypi.org/help/faqs/
17. Robinson, A. (2014). *Raspberry Pi projects. [electronic resource]*. Chichester, England : Wiley, 2014.
18. Arduino Uno. (n.d.). Retrieved from http://www.arduino.cc/en/Main/ArduinoBoardUno
19. BeagleBone Black. (n.d.). Retrieved from http://beagleboard.org/black
20. SOAP vs REST web services - javatpoint. (n.d.). Retrieved June 9, 2015, from

http://www.javatpoint.com/soap-vs-rest-web-services
21. Castillo, P. A., Bernier, J. L., Arenas, M. G., Merelo, J. J., & Garcia-Sanchez, P. (2011). SOAP vs REST: Comparing a master-slave GA implementation.
22. Bozdag, E., Mesbah, A., & Deursen, A. (n.d.). A Comparison of Push and Pull Techniques for AJAX. *2007 9th IEEE International Workshop on Web Site Evolution*.
23. Vectors Tutorial. (n.d.). Retrieved from http://algebra.nipissingu.ca/tutorials/vectors.html
24. Hartigan, J. A. (1975). *Clustering algorithms*. New York, Wiley.
25. 15 Naive Bayes. (n.d.). Retrieved from http://docs.oracle.com/cd/B28359_01/datamine.111/b28129/algo_nb.htm#DMCON018.
26. Chandy, K., & Charpentier, M. (n.d.). Self-Similar Algorithms for Dynamic Distributed Systems. *27th International Conference on Distributed Computing Systems (ICDCS '07)*.