Automatic Munsell Color Detection

in Mobile Applications

Jennifer Benedict
Drew University, 2018

ABSTRACT

The process of color detection via mobile applications is not a new concept, as there exist a wide variety of mobile applications found in app stores that allow users to automatically detect the colors in an image with a click of a button. However, these applications typically only translate the image's pixels into color standards such as RGB and HSV. Although these are commonly used standards for interpreting and analyzing color, there exists no application that can take any image and process the image's colors in terms of Munsell, a widely used cylindrical color standard in such fields as archaeology, plant science, food science, and more. This thesis proposes an approach implemented as an android application that allows users to take a picture of an object and know what its corresponding Munsell Color is. This thesis also proposes and evaluates a calibration technique to be used in the application that would provide users with an accurate reading of the colors in the images they take, regardless of environmental factors like sunlight. The purpose of this research is to provide users of the Munsell Color system with a tool that can be used in their respective fields and increasing accessibility to this system, as well as an understanding of the current calibration technique used and how it can be adjusted for their particular needs.

Automatic Munsell Color Detection

in Mobile Applications

By Jennifer Benedict

An Honors Thesis
Submitted in Partial Fulfillment of the
Requirements for the Degree of Bachelor in Arts with
Specialized Honors in Computer Science
at Drew University

May 2018

Acknowledgments

I would like to thank Dr. Emily Hill for her guidance and assistance in conducting this research as well as for her role as the primary advisor for this Thesis. I would also like to thank Dr. Bjorg Larson for her assistance in implementing the calibration technique proposed in this thesis as well as Dr. John D. Muccigrosso for his guidance in designing the application and for providing the idea and opportunity to work on this project. I would also like to take a moment to thank Dr. Chris Casement for his assistance in providing detailed statistical analysis as well as writing several scripts to help analyze and visualize the data. I would also like to thank Camille Berger, Kent Harris, and Brittany Grabowski, Drew University students and the initial co-developers of this application, along with myself, for laying the foundations of this application to be further developed and improved. I would also like to thank Andrew Castelluccio, a Drew University student that served as a beta tester for this application. Finally, I would like to thank Drew University and the Department of Mathematics and Computer Science for allowing me to conduct the research presented in this thesis.

Table of Contents

Chapter 1 Introduction

Color exists all around us, from the photos we see to the foods we eat. A color system called Munsell is an extremely useful standard and notation to use in a variety of fields due to its ease of color identification, however, accessibility to Munsell resources remains an issue terms of cost. This chapter will describe the fundamental aspects of color perception and why the Munsell color system is significant, as well as the need for the application presented in this thesis.

1.1 Color Perception

The perception of color begins with light. Whether your light source is a fluorescent light bulb or natural sunlight, light travels in waves, each wave with a corresponding wavelength. Light ranging from 400-700 nm is referred to as the visible spectrum shown in Figure 1.1.1, and is visible to the human eye. These wavelengths are perceived as colors, with the longest wavelength being red and the shortest violet [1].

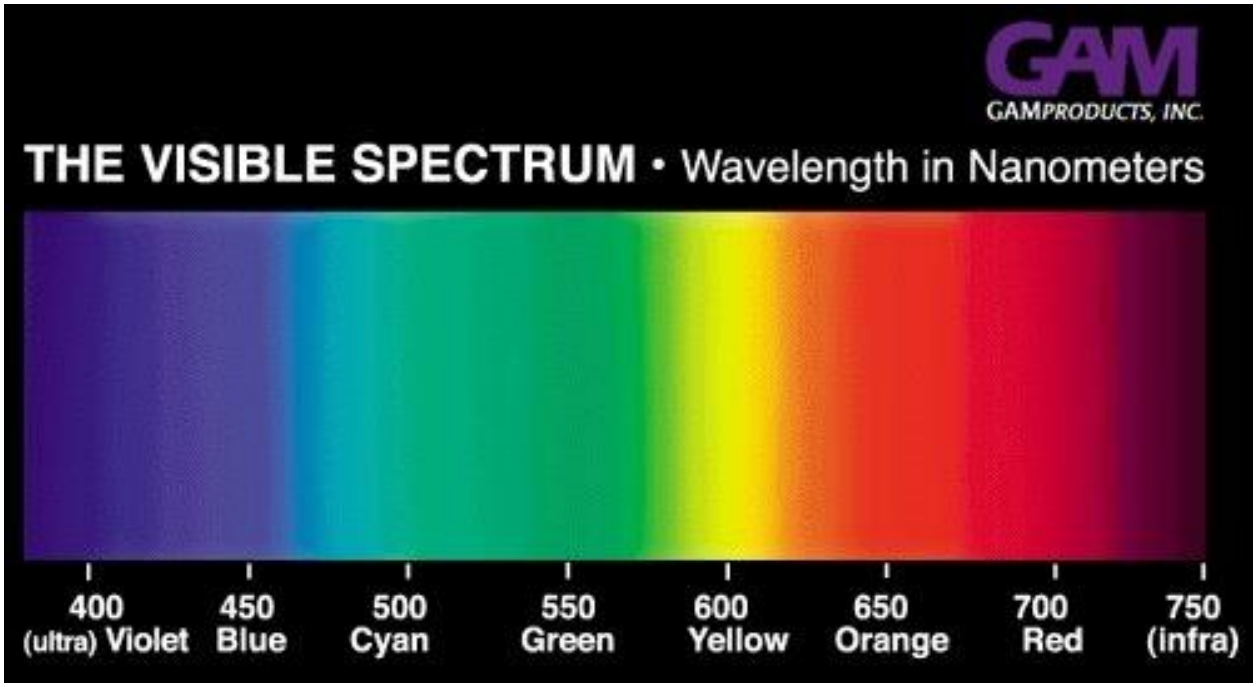


Figure 1.1.1 The Visible Spectrum "The Visible Spectrum - Light." *Www.gamonline.com*, GAMPRODUCTSINC., www.gamonline.com/catalog/colortheory/visible.php.

These wavelengths are significant in color perception because what the brain perceives to be color is the property of the object which causes different wavelengths of light to be absorbed and reflected off of itself [1]. For example, when white light, a combination of all visible light wavelengths, is shone on an apple, the brain perceives only the color red because red wavelengths are reflected from the apple, while the other wavelengths are absorbed. Whether or not a wavelength is absorbed is dependent upon the inherent chemical structure of an object. When a wavelength of light reaches the retina of the eye, the photoreceptor cells, rods and cones, decipher what color is being reflected. The red, blue, and green cones respond to the wavelength, and send the brain information, resulting in color perception. The way in which humans identify color is crucial to the Munsell color system described in the next section [2].

1.2 Munsell Color

On a daily basis, many use common terms to identify color, such as “light blue” or “blood orange”. However, for many occupations and areas of study, distinguishing between colors must be done precisely. For example, an archeologist may want to know the exact color of a limestone artifact for artifact comparison or to trace its origins. Similarly, an electrician working in a mass power plant may have the task of organizing thousands of wires, and simply knowing that one wire is light blue and another wire is a slightly lighter blue may not be enough to accurately distinguish between them. In an environment in which color identification is crucial to the maintenance of infrastructure, a way to precisely identify color is fundamental.

Many standards exist to interpret and identify color for multiple purposes, including HSI, HSV, and HSL, some of which are alternative representations of the Red Green Blue (RGB) model of color where color is represented by its components of red, green, and blue [1]. Most color standards define colors in terms of numeric coordinates, for example, the color black in RGB is identified by the coordinates (0,0,0), in which each coordinate can range from 0 (meaning no color) and 255 (the most amount of color).

A unique color standard called Munsell is distinguished from other standards for its consideration of how humans recognize color. In 1898, Albert Henry Munsell created the Munsell color system, a system in which hue, value, and chroma are the three attributes used to identify a color. He was motivated to develop this system by the fact that no system existed that would be accurate in expressing any particular color while still being intuitive to use [3]. The three attributes of the Munsell system follow the sequence of how humans recognize a color and are represented by small chips, which will be further discussed in chapter 2.

For the soil scientist wanting to know the color of a limestone artifact by identifying the Munsell chip notation, he could identify the Munsell chip notation of the limestone with much more ease, as shown in Figure 1.2.1, than the RGB coordinates of the limestone. For example, it would be much easier for the archaeologist to be able to identify Munsell chip notation using the three attributes, than knowing specific numeric RGB coordinates. The current uses for the Munsell are many in number, and the Munsell system may impact aspects of our daily lives. For example, Munsell government standards are used to analyze produce, distinguishing a healthy cherry or potato from an unhealthy one. Leading brands like Hoover and Black & Decker also utilize the Munsell color system to communicate color to suppliers for their products [4]. However, there are some setbacks to this system, such as cost and ease of use, that this thesis hopes to improve.

Figure 1.2.1 "Soil & Wine Quality." ***Www.munsell.com*****, munsell.com/color-blog/soil-wine-quality-a-geologists-analysis/.**

1.3 Munsell Application

As widely used as the Munsell system is, it is not easily accessible. Munsell color books, containing the color chips used for color identification, can range anywhere from $200 for a pack of certain individual chips to over $1,000 for a full book of color chips. Given how useful this system is, it should be made more accessible to companies, researchers, and individuals who would like to use it. For example, if college students would like to conduct independent research in plant science, but do not necessarily have funding to purchase a Munsell color book, it would be of great use to have an affordable digital alternative. The Munsell color books are also weight that users need to carry around, and can weigh up to ten pounds. For researchers and individuals working critical infrastructure occupations, the inconvenience of a ten-pound book can be burdensome in our digital age, especially in time-sensitive projects where color identification must be done in a reasonable amount of time before a deadline. Although the Munsell color system is intuitive to understand and use, these books often include over 1,600 small Munsell chips that can be lost or misplaced while being used. In order to promote use of the Munsell color system while making it accessible and affordable, a digital alternative for Munsell color identification may have advantages over using the book.

While not everyone may be able to afford a Munsell color book, most people already have cellular devices with applications. This thesis proposes a mobile application that allows individuals to take a picture of an object and have its corresponding Munsell chip notation returned. This application also includes functions for users to record their findings for future reference. the success of this application relies on the calibration technique proposed in this thesis, which adjusts color for various lighting conditions, which will be discussed in greater detail in Chapter 2.

The remainder of this thesis will describe the Munsell system as well as color recognition techniques in more detail. We will also highlight the challenges in developing and implementing such a prototype. Chapter 2 of this thesis will consist of a literature review around concepts such as color calibration, the Munsell color system, and the Android framework. Chapter 3 will describe the requirements for the application, the application in its current state, and possible future developments. Chapter 4 will consist of an evaluation and analysis of a crucial feature of this application that is the calibration feature. Finally, Chapter 5 will conclude this thesis with a summary of the results found in Chapter 4 as well as plans for future work on this application.

Chapter 2 Background and Related Work

The mobile application developed as part of this thesis draws upon a variety of fields of research, including physics, computing, mathematics, and even photography. The first part of this literature review explores the Munsell Color system in more detail, explaining how to interpret Munsell notation and what distinguishes it from other well-known color standards. Research in photography will be used to explore a calibration technique proposed further in this thesis. This literature review draws upon mathematics to understand how Munsell Color data is transformed for analysis, and how computing is drawn upon to understand the Android operating system, the underlying framework for this application.

2.1 The Munsell Color System

2.1.1 History

Albert Henry Munsell, known for his contributions in color theory and color science, invented the Munsell Color system seeking to create an organized method for identifying and interpreting color, and thus bring order to its study [5]. His believed that interpreting color using two dimensions or characteristics was not sufficient for identification and measurement, which led him to his three-attribute cylindrical color system in 1898 [3]. The three attributes of color that Munsell found to be crucial for identification were: Hue, Value, and Chroma. By means of his system, Munsell developed a way to completely express any particular color [3]. In 1918 Munsell formed the Munsell Color Company to produce color standards for different fields. The formation of the company was followed by his death, after which his son Alexander Munsell sponsored studies at the National Bureau of Standards and in the Munsell Color Laboratory, which led to the creation of the 1929 version of the Munsell Color Book with 20 hues, improved in 1950 to include the 40 hues that are used today [5].

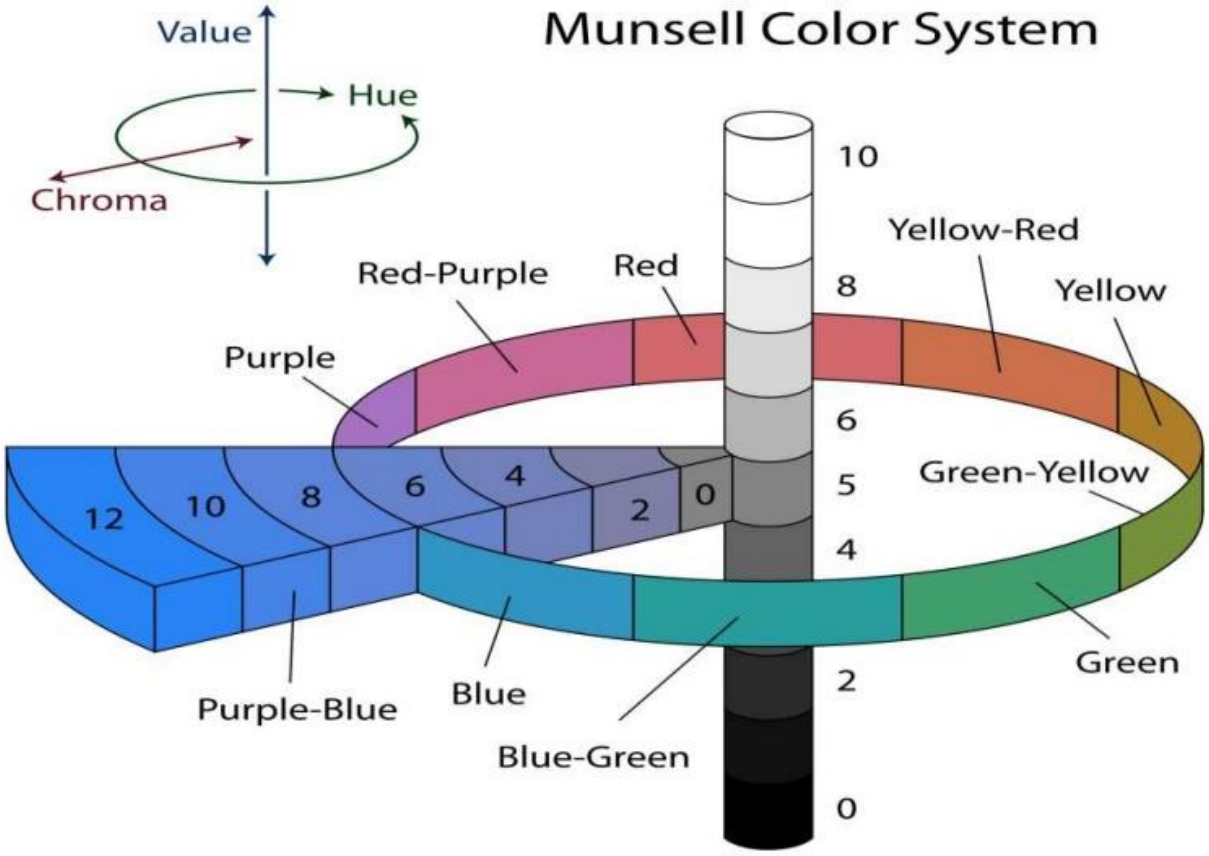


Figure 2.1.1 Munsell Color Chart "Munsell Color System." *www.synapse.koreamed.org*.

2.1.2 Munsell Color Notation

The *hue* of a color can be thought of as the name of a color, or the quality that allows us to be able to distinguish red from a green and green from a blue, based on the length of the waves impinging on the retina [3]. The human brain can easily attribute a hue to an object, as it is first recognizable characteristic of a color that the brain identifies. Given his interest in color education, Munsell recognized that the hue was the first characteristic of color that children learn to recognize [3]. For Munsell, hue is defined by a number between 0 and 10 prefixing one of ten hue names: red (R), yellow-red (YR), yellow (Y), green-yellow (GY), green (G), blue-green

(BG), blue (B), purple-blue (PB), purple (P), and red-purple (RP), resulting in 40 hues [6]. Despite being easily detectable on its own, a hue does not provide an accurate representation of color, without the other two attributes.

The *value* of a color is the quality that allows for a distinction between light and dark. This quality is attributed to the difference of height, or amplitude, of the wavelength that is reflected onto the retina [3]. Value is loosely applied using terms like "dark red" or "light red"; however, using such terms does not provide a precise interpretation, as there exist many shades within the range of what we know to be "dark red" or "light red". The value of a color is defined by a number between 0 (black) and 10 (white), signifying lightness or darkness [7]. As the value increases, the lightness of the color increases.

Chroma is the third quality that describes intensity, allowing for the distinction between a strong color and a weak color. Loss of chroma is loosely referred to as fading. For example, one could think of a pair of blue jeans that have lost the intensity of the blue after a few washes. Chroma is described in physics as the purity of a wavelength apart from the others [3]. For example, an intense blue is the result of white light shone on a blue object, which reflects just one wavelength at high intensity. Chroma is also defined by Munsell to be a number starting at 0 (grey) and increases as the color becomes more saturated or intensified [8]. However, unlike the previous attributes, the scale of chroma has no arbitrary end. For the purpose of the evaluation in this thesis, we will only be evaluating Munsell chips with chroma less than 16.

Using just two of these characteristics to define color would not be as precise and accurate as using all three of them, and Figure 2.1.1 depicts how the three qualities are combined to create a system of easily distinguishable colors. Although Munsell's system allows for ease of color identification, Munsell color notation can be difficult to comprehend at first glance. Each

color chip is placed in the color book strategically based on the three qualities discussed, along with its designated a chip notation.

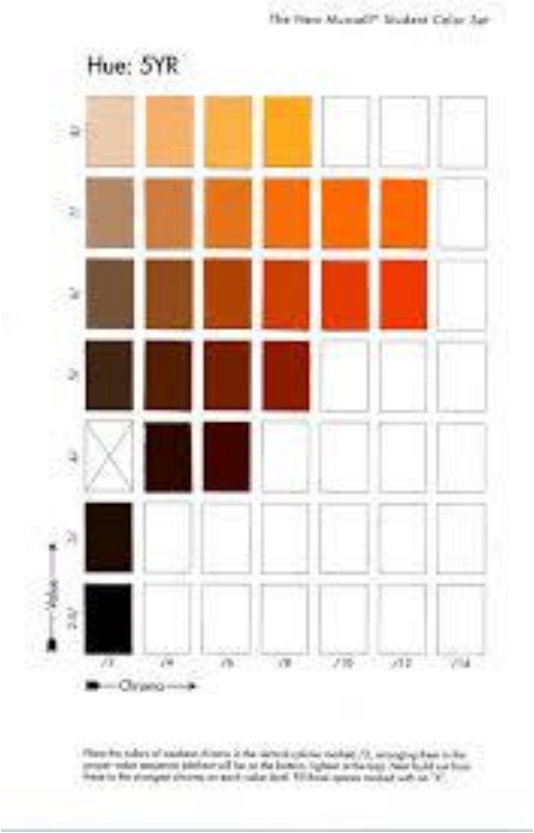


Figure 2.1.2: Page 5.0YR of the Student Munsell Color Book *The New Munsell Student Color Set*

Each page in a Munsell color book is separated by hue, as it is the first quality recognized in Munsell color matching. Figure 2.1.2 is a page of a Munsell color book, and on this particular page, the hue 5YR is represented. Also seen on this page is a vertical axis representing value and a horizontal axis representing chroma [9]. For example, the Munsell chip 5YR 7/2 has a hue (color) of 5 Yellow-Red, a value (lightness/darkness) of 7, and a chroma (strength/weakness) of 2 [9].

2.2 Image processing for color detection

The process of color detection using cameras and mobile devices is similar to the way humans identify color. The application proposed in this thesis utilizes the camera as well as standard Java libraries for processing images to extract color data. Similar to how human perception of color begins with a wavelength and ends with signals being transmitted to the brain, mobile devices and cameras are able to detect color as well., cameras have sensors which respond to wavelengths of light Comparable to the retina of the eye. Color sensors in mobile cameras include three color filters, red, green, and blue, that are comparable to the cones in a human eye. These sensors are responsible for filtering out the components of each color in the wavelength [10]. Knowing the colors present in an image then allows for algorithms to further analyze the colors.

Utilizing the `android.graphics` package, bitmap images can be analyzed by the pixel. Using a simple *for* loop, each pixel in an image is given an x, y coordinate which can be retrieved and referenced for color analysis [11]. The graphics package includes a class called color, which allows for extraction of color information at the pixel level. The functions `color.red()`, `color.blue()`, and `color.green()` return the components of red, green, and blue found at each pixel [12]. These components range from 0-255, the standard for RGB coordinates. Using this method of extracting color at the pixel level, an average of the red, green, and blue components of the image is taken. This results in one set of RGB coordinates that represents an entire bitmap image.

2.3 Android Overview

Android is an open source software stack utilizing a Linux-based operating system, used for writing applications [18]. Android applications are usually written in the Java programming language and use the Android Standard Development Kit (SDK). The SDK provides all the tools and APIs one would need to write, test, and debug applications. Android applications are typically developed in Integrated Development Environments, software applications that provide an environment that facilitates software development and debugging. The official IDE for Android is Android Studio, which is also the IDE used for the development of the Munsell application presented in this thesis [18].

Android is considered to be a software stack because it consists of various layers, each layer having its own functions and capabilities while also providing services to the layer above it. The layers are applications, application frameworks, libraries, and the Linux kernel as shown in

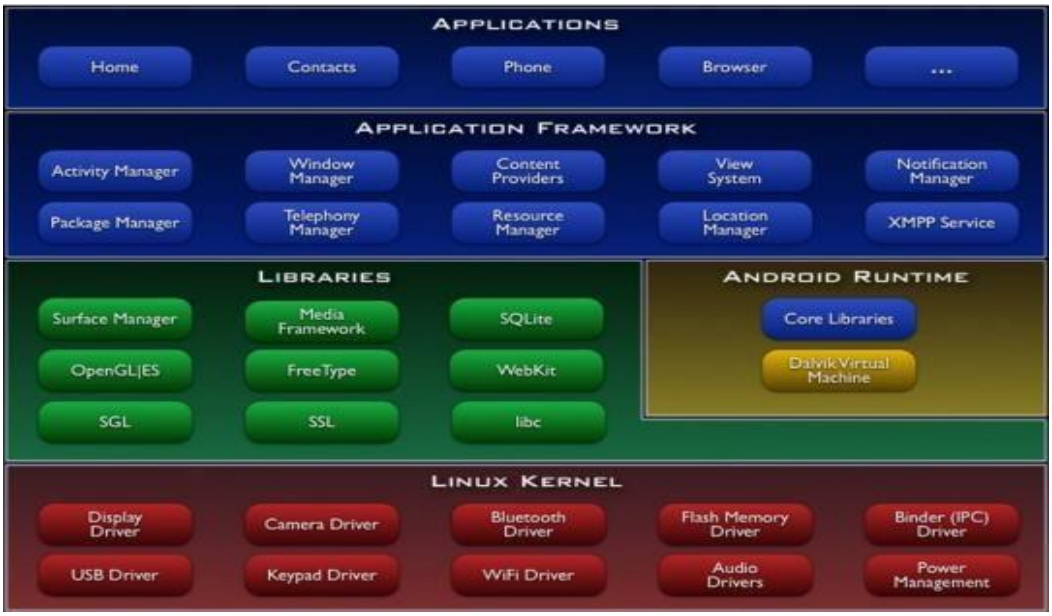


Figure 2.3.1 Android Stack Architecture "Android-Architecture." ***www.theandroid-Mania.com.***

2.3.1 Application Layer

All Android applications are built on the application layer, using the same API libraries, whether or not they are native or user-defined applications. The application layer uses the classes and services made available by the layer underneath it, the application framework layer and runs within the Android runtime. Even though each application only has access to its own components and cannot access other parts of the system without permission, it is possible for an application to access core or native applications such as the email client or maps, which the Munsell application does [18].

Application components form the essential foundation of an application and each component contributes to the application in a different way. There are four types of application components which are Activities, Services, Content Providers, and Broadcast Receivers.

1. **Activities**: An activity is a single screen with a user interface (UI) which can be thought of as the entry point for the user.
2. **Services**: A service does not have a UI as it performs long-running operations in the background. An example of a service would be music playing in the background when a user is in a different application.
3. **Content Providers**: Content providers allow for data sharing between applications upon request, and handles such requests.
4. **Broadcast Receivers**: This component responds to broadcast messages from other applications [18].

2.3.2 Application Framework Layer

The Application framework layer directly works with the Application layer above it by providing higher-level services in the form of Java classes used to create Android Applications. This layer also provides generic abstractions for access to hardware and manages the user interface as well as application resources [18]. The Android framework layer includes the following services:

1. **Activity Manager**: Controls and manages the lifecycle of the application as well as the activity stack.
2. **Content Providers**: Allows for accessing data from other applications as well as the ability for an application to share its own data.
3. **Resource Manager**: Provides access to and manages resources such as images, strings, and color settings.
4. **Notifications Manager**: Allows the application to display notification alerts to the user.
5. **View System**: A set of views that aid in creating the user interface of an application [18].

23.3 Android Libraries

This layer includes Java-based libraries that are specific to Android development. Android libraries basically stand as a set of instructions that allow the device to handle different kinds of data. These libraries include application framework libraries as well as libraries that allow for UI building, graphics drawing, and database access; these libraries are written in C and C++ [18]. A summary of some core Android libraries available to developers includes:

1. **`android.app`**: Contains high-level classes which encapsulate the overall Android application model.
2. **`android.content`**: Allows for accessing and publishing data.

3. **android.database**: Used to access data published by content providers, including SQLite database management classes.
4. **android.os:** Provides basic operating system services.
5. **android.hardware**: Allows access to hardware features such as the camera or sensor.
6. **android.media:** Allows for the use of interfaces such as audio or video.
7. **android.text:** Allows for creating or altering text on the screen.
8. **android.util:** Allows for use of common utilities like string conversion.
9. **android.view**: Contains basic user interfaces.
10. **android.widget:** Contains UI elements to be used in activity views, like buttons or checkboxes [19].

2.3.4 Android Runtime

The Android runtime section exists in the same layer as the android libraries. This section provides a key asset called the Dalvik Virtual Machine (VM), similar to a Java Virtual machine, but specifically optimized and used for Android. The Dalvik VM was developed by Google and relies on the Linux Kernel for low-level functionality. It should be noted that The Dalvik VM is specialized for small systems with low processing power and low memory, and therefore is not compatible with the standard Java VM [20].

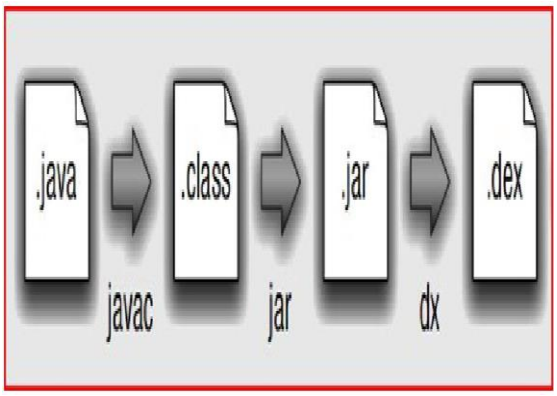


Figure 2.3.2 Compilation of java code into .dex code. "Conversion from .Java to .Dex." ***Http://Freenotesforeveryone.blogspot.com*****.**

Figure 2.3.2 shows the process of the Java code being compiled by the java compiler to generate .class files. These files are then converted to .dex files (Dalvik executable format) to be interpreted by the Dalvik VM [18].

Android runtime also provides core java libraries allowing developers to create Android applications using standard Java programming languages. A standard Java library used in the application presented in this thesis is Color.java.

2.3.5 Linux Kernel

Found at the root of the Android software stack is the Linux kernel. The Linux kernel is responsible for device drivers like the camera, display, keypad, etc. The kernel also provides preemptive multitasking as well as memory, power, and process management [18]. Chapter 3 of this thesis will describe how the Munsell application works in conjunction with the layers discussed to allow for this application to function as necessary. The next chapter will describe key algorithms and concepts used to optimize the mobile application in accounting for various lighting scenarios.

2.4 State of the Practice

There are a few mobile applications developed to try to make use of the Munsell color system more accessible. One application in particular is the *Munsell Viewer*. This is a mobile application, available for both iOS and Android operating systems, and essentially is a digitized version of the full Munsell color book. Figure 2.4.1 depicts one view of the application, in which colors are arranged on the screen as they would be in a Munsell color book. Users are able to hold the screen near the object for which they would like to identify Munsell color for, or if the object is small enough, the user can place the object on the screen. For example, soil scientists can place their soil sample directly on the screen for comparison.

Because this application is available at no cost, it minimizes the issue of accessibility as mentioned previously. However, the limitation of this application and similar applications is that users are still relying on their own perception of color. There is no automation process in which the application is able to detect color automatically.

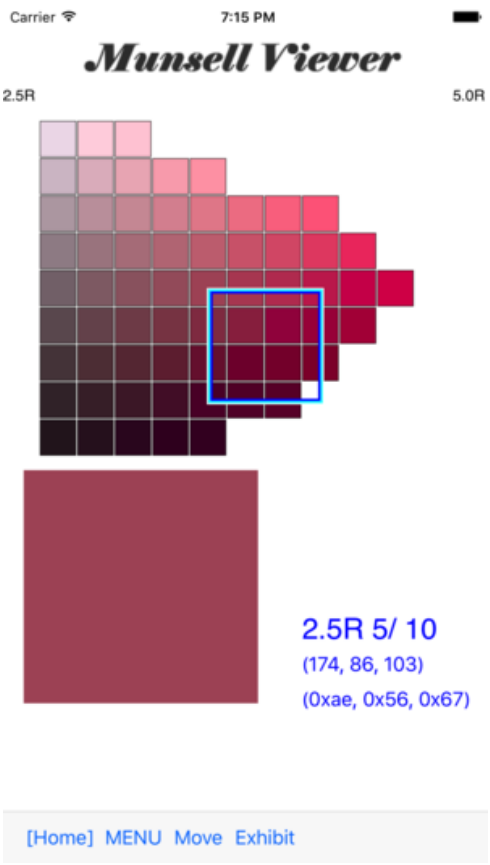


Figure 2.4.1 ***https://itunes.apple.com/us/app/munsell-viewer/id1240497502?mt=8***

Another application has been developed, specifically for soil research, is the *Munsell Soil System App*. This application was developed by researchers at the University of Grenada, in Spain, to allow users to take a picture of a soil sample, and detect its Munsell chip notation. This application stores information for 238 Munsell chips, particularly Munsell chips from the Munsell soil chart. This is a limitation of the application because it cannot be used to detect color on just any given object or artifact, but rather on objects that are within the range of soil color, like those shown in Figure 2.4.2. This application does however include a calibration feature, in which a white colored chip is used as a constant variable to account for color distortions due to lighting. I will describe in further detail why a calibration feature like such is necessary, how this calibration technique works, and how it is used in the application presented in this thesis.

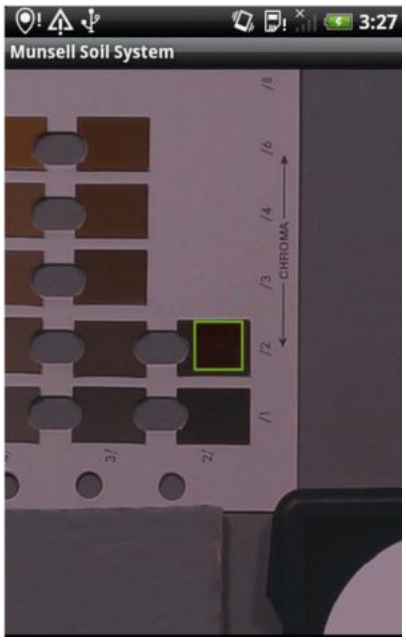


Figure 2.4.2 [15]

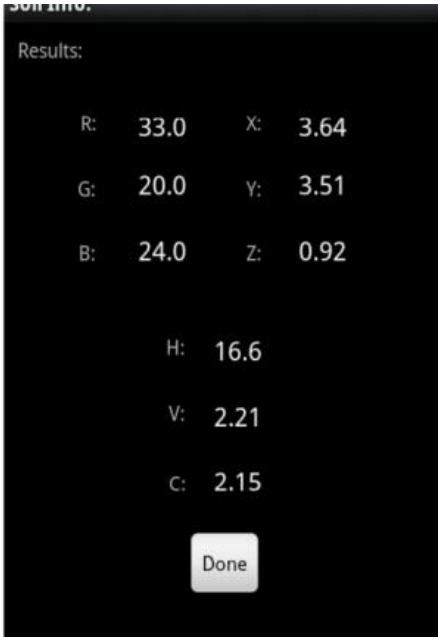


Figure 2.4.3 [15]

Chapter 3 Calibration

This chapter will describe the necessity for a calibration in the application developed as part of this thesis, as well as how them calibration technique was devised and used.

3.1 Color Calibration

The human brain allows for visual adaptability, which is why humans may not be able to distinguish differences in color caused by varying sources of light. For example, if we look at an apple in broad daylight and look at the same apple as the sun is setting, even though the apple may have an orange tint during sunset because the light reflected by the apple is not the same in each setting, but our brain accounts for this change and are still able to identify that the apple is red. This is caused by color constancy, a term describing the relatively constant perception of color under varying illuminants or light source. Factors like previous knowledge of the color of an object, or knowledge of the effects of light contribute to our ability to correctly identify color, regardless of the illumination [13]. However, when digital cameras capture images, the resulting image is based solely on the pixel values recorded by the sensors in the camera, which are

affected by the illuminants in the environment in which the image is captured. Many raw images may have a distinct color cast, which appears in the image due to the color temperature present in the light source [13]. In other words, unlike the brain, digital cameras do not have color constancy to account for illuminants.

In order for the color detection mentioned in the previous section to function optimally, the image taken by the camera must be an accurate representation of what the object looks like, despite environmental factors like light. The calibration mechanism that is used and tested in this application of color detection is similar in nature to a concept called white object purification, used widely in photography along with white balance detection. White object purification is used for the purpose of removing any color cast created by varying color temperatures at the scene of capturing, "purifying the white object", in order to capture images that remain constant in color [13]. Many digital cameras have automatic white object purification capabilities, however, photographers still use manual white balance cards to achieve color normalization [13]. White balance cards are white cards used to normalize colors in an image. Photographers first take an image of the white card, then proceed to take their desired images, and when editing these images, the white card's RGBs are normalized to what they should be (255,255,255), and applying that normalization to all the other images should essentially account for light, as depicted in Figure 3.1.1. Mathematically the problem of color constancy in digital cameras can be resolved if the red, green, and blue (RGB) sensor responses under an unknown illuminant or light source, can be mapped to RGB under a known illuminant, assuming the additive property of RGB coordinates [14].

White object purification involves storing RGB coordinates for the photographed white card or object. Known RGB coordinates for the white balance card (sometimes substituted by

grey cards), are also stored. Subtracting the object's RGB coordinates captured by the sensors in the device camera from the known RGB coordinates of the white card, returns a factor of difference, that when applied to the image, accounts for distorted color casts on the image from the light source's color temperature [13].

This technique is used in this application, using Munsell chips as substitutes for white balance cards, under the assumption that the additive property of RGB coordinates can be applied using any colored chip. This assumption will be tested and evaluated in later chapters, with the goal of proving that a user of the Munsell application can use any Munsell chip, so long as it has known RGB coordinates, to calibrate their captured images and account for distorted color casts.

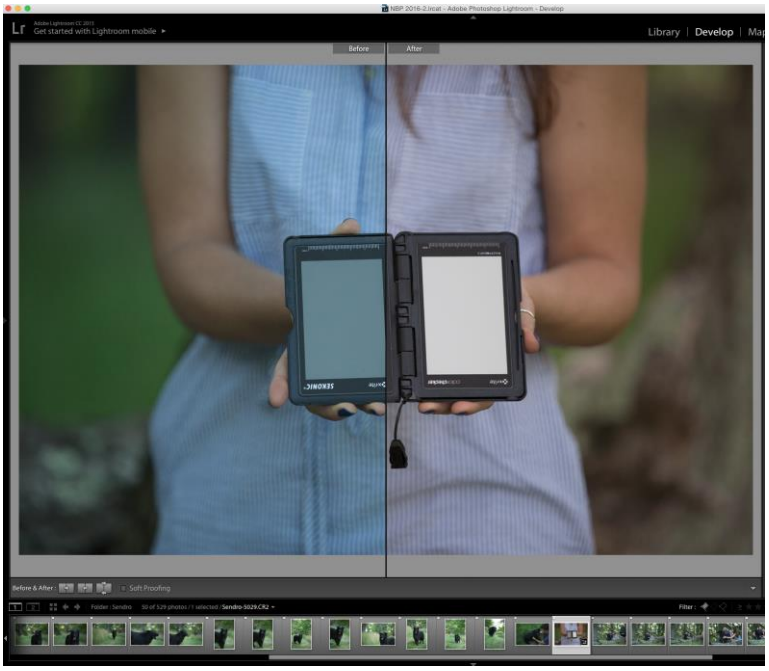


Figure 3.1.1 White balanced card used to normalize colors in an image

www.hairoftheblogdog.com

Similar mobile applications have been developed to allow for readings of Munsell color under different illuminants and for different types of objects. The Munsell Color system is used

widely for analyzing soil samples, and research has shown that using a specific Munsell database that includes chips from the Munsell soil chart under controlled illuminants provides accurate readings of Munsell color [14]. Studies have also been conducted to analyze the mobile phone's accuracy for determining Munsell color in soil, also under controlled illuminants, using standard white cards for calibration, as discussed in the previous section [15]. This thesis, however, proposes using different colored calibration chips to analyze differences in accuracy.

As discussed in the previous section, the analysis of color occurs at the pixel level for color detection, as well as in the calibration technique that this thesis proposes. The issue of color constancy is on-going in many applications. However, it has been proven that color constancy can be maintained at the pixel level, so long as the red, green, and blue device sensor responses, or RGB, under an unknown illuminant, can be mapped to corresponding RGB under a known reference light [16]. This is essentially how the proposed calibration technique, proposed in this thesis, functions. The sensor RGB under an unknown illuminant or light source, are mapped to what the RGB should be under a known light source, and the difference is calculated and applied to all the subsequent bitmap images until calibration is reset.

3.2 Analyzing Munsell Data

Below we will conduct a quantitative analysis of calibration methods, as mentioned in the previous section. In order to perform a quantitative analysis of colors, there must be a way to quantify the distance between any two particular colors.

Depending on what color standard is being used to interpret color, different distance formulas may be used. For example, in the standard RGB color space, the Euclidean distance formula can be used, as shown in Figure 3.2.1, where RGB correspond to XYZ [17]. Because

RGB points are numeric in nature, using them as Cartesian coordinates in a three-dimensional space in order to calculate distance is much more intuitive than doing so with Munsell data.

As previously stated, Munsell chips each have three attributes, hue, value, and chroma. Roy G D'Andrade and A. Kimball Romney, are anthropologists who have proposed a method for transforming Munsell data into numeric data, given hue, value, and chroma. Using the following formula, the same transformation to standard Cartesian coordinates can be made using basic trigonometry [17].

*X = sin (Hue) * (Chroma)*

*Y = cos (Hue) * (Chroma)*

Z = Value

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2}.$$

Figure 3.2.1 Euclidean Distance Formula Three Dimensions

https://en.wikipedia.org/wiki/Euclidean_distance

The first step in converting hue, value, and chroma into Cartesian coordinates XYZ is to convert the hue to an angle. Figure 3.2.2 shows a model of the Munsell color book with each of the 40 pages of hues fanned out in a circle. Looking down at the book from above, each page/hue being positioned to be equidistant 9° from its neighboring page/hue [17]. The hue 5R has been chosen by D'Andrade and Romney as the origin at 0°, with the remaining 39 hues distributed clockwise 9° apart. The chroma of a color is the distance from the central point of the book or the binding, outward along the page, and value is simply the height of the color on the page [17]. Having the hue now represented as an angle, transforming the three attributes to polar coordinates XYZ, can

be done using the Munsell transformation formulas discussed [17]. For example, the coordinates for chip 7.5YR 5/8 are (1.25, 7.9, 8).

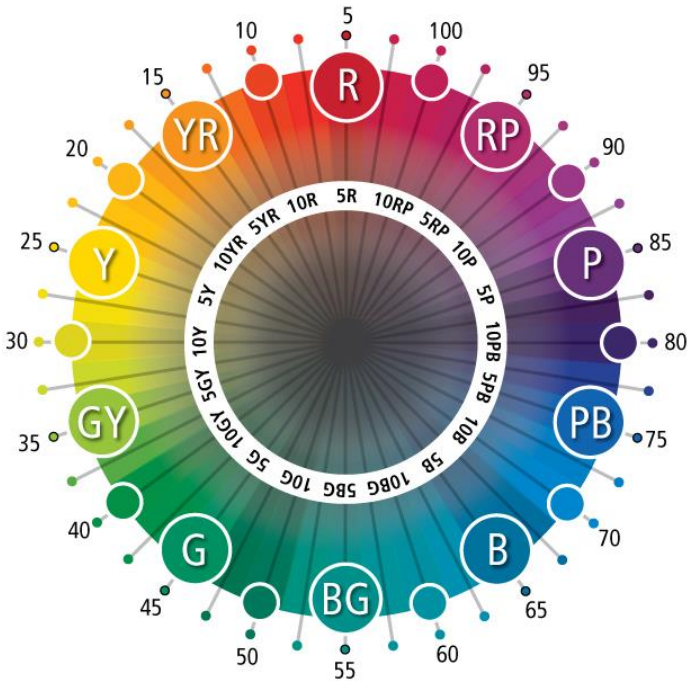


Figure 3.2.2 Hue model of the Munsell Color System Munsell Hue Circle

www.munsell.com

Measuring and understanding the distances between two colors in RGB is intuitive, as it is the combined distance between the red, green, and blue components of a color. For example, The Munsell chips 5.0G 6/8 and 5.0G 6/6 in Figure 3.2.3, perceptually seem to be very similar colors and are one unit in chroma apart, have an RGB distance of 29.7. When calculating the distance between these two chips using the transformation previously discussed, the Munsell distance between these chips is 2. Similarly, the distance in RGB between chips 5.0G 6/8 and

5.0G 5/8, which are only 1 unit in value apart have an RGB distance of 56.6 and a Munsell distance of 1. While RGB distances purely quantify the distances of red, green, and blue components in the colors being compared, Munsell distances take into account the spatial positioning of each chip, relative to another. Given that Munsell chips are positioned strategically to make identification intuitive, it is possible that colors that are not within the same hue, have smaller distances in Munsell than colors on the same hue page. For example, when comparing the chip 5.0G 6/8 in Figure 3.2.3 to chip 7.5G 6/8, the RGB distance between them is 14.21 and the Munsell distance between them is 0.31. These chips are very similar in color and are part of the same hue family of green, however, are part of different hue.

Though distances in RGB tend to vary for colors among the same hue family, Munsell distances tend to be relatively small. When comparing chips between different hue families, we tend to see higher distances in RGB as well as in Munsell. For example, when comparing chip 5.0G 6/8 with chip 5.0RP 6/8, we find the distance between them in RGB to be 150.6 and the distance in Munsell to be 15.2. Distances in Munsell and RGB increase as colors move further apart in the hue circle depicted in 3.2.2.

Now that it is understood how color is perceived, how mobile phones detect color, and how color constancy can be mimicked in mobile phones for accurate color measurements, the next chapter will describe the application in its current state as well as its features.

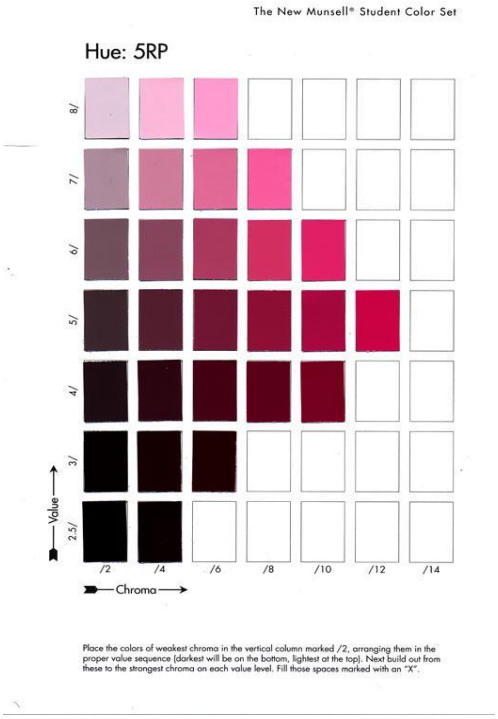


Figure 3.2.3 Page 5.0RP of the Student Munsell Color Book *The New Munsell Student Color Set*

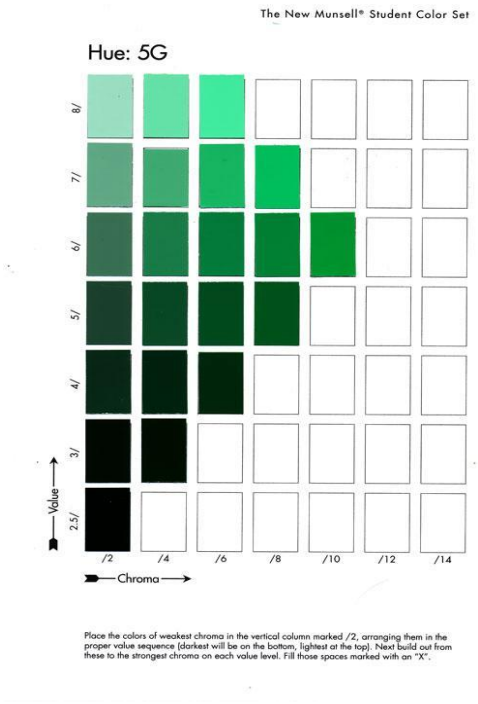


Figure 3.2.2 Page 5.0G of the Student Color Book *The New Munsell Student Color Set*

3.3 Munsell Color Matching Algorithm

A distinguishing feature that this application provides is the conversion between the Munsell and RGB color standards. As the sensors on a camera capture the colors in an image, the data is naturally processed in RGB units. Using a CSV file that maps Munsell chips to their corresponding RGB values. The first mapping between units that occurs is from Munsell to RGB when the user selects their desired Munsell calibration chip. The Munsell chip is then mapped to

its RGB coordinates, which is then compared to the RGB coordinates of the image of the Munsell calibration chip, to calculate an RGB calibration factor. The second conversion between Munsell and RGB occurs when the calibration factor is applied to the image that the user takes of the desired object. In this conversion, the calibration factor is applied to the RGB coordinates of the image, and the resulting RGB coordinates are mapped to the Munsell chip with RGB coordinates closest to the calibrated RGB coordinates. It should be noted that the conversion between Munsell and RGB would allow the user to use Munsell chips or RGB coordinates for calibration. However, for the purpose of this evaluation, the application has been designed to allow users to choose a Munsell chip for calibration. Now that the basic algorithms used in the application have been described, the next chapter will provide an overview of the application in its current state.

Chapter 4 Munsell Application

This chapter will explore the structure and design of the mobile application in its present state, the user interface, as well as define the requirements both functional and nonfunctional as specified by the primary user of this application at the moment, Professor John D. Muccigrosso, Department Chair and Professor in Classics at Drew University. This application was initially developed with three other students as part of the requirements for a Software Engineering course at Drew University. My contributions to this application include the calibration feature, cropping feature, and distance measurements for RGB and Munsell. Source code can be found in Appendices A through L or on GitHub at the following link: https://github.com/JenniferBenedict/Munsell-Android-Color-Recognition-App.

4.1 Requirements: Functional

This section defines the functional requirements which describe the expected behavior of the application as well as its individual components.

FR1 Android Application for Munsell Chip Detection

The android application should allow the user to take a picture of an object and have its corresponding Munsell chip returned.

FR2 Calibration Function

The application should have some mechanism for calibration of color in order to account for differences in lighting environments.

FR3 Image Selection

The application should allow the user to select a specific part of an image and have its associated Munsell chip value returned.

FR4 Email Results

The application should allow the user to email the results to the desired recipient email address for further analysis or sharing of results.

FR5 Save Results

The application should allow the user to save a copy of the image taken as well as the associated Munsell chip value to the device.

FR6 GPS Location

The application should allow the user to be able to include their GPS location as part of the results being emailed through the email feature.

FR7 Display Results

The application should display the results to the user in the order of hue, value, and chroma, as well as a background image that matches the Munsell chip.

4.2 Requirements: Non-functional

This section defines the non-functional requirements which the overall performance of the application as well as the functional requirements listed in the previous section.

NFR1 Android Compatibility

The application should perform as expected on all Android devices, including both mobile phones as well as tablets.

NFR2 Screen Compatibility

The performance of the application should not change based on screen size and the display should adapt to devices with larger screens such as tablets.

NFR3 User-Friendly GUI

The graphical user interface (GUI) of the application should be user-friendly and simple in design.

4.3 Activity Flow Overview

This section describes in detail the flow of the activities within the Munsell application.

Main Activity

Upon opening the application on a mobile device, the user is met with the home screen with an alert dialog prompting the user to turn on location accessibility as shown in Figure 4.3.1. Turning on location is necessary for the location field in the Submit Form, which will be discussed later. The user is then taken to the home screen which has three buttons, each with a different function as shown in Figure 4.3.2.

The *Take Picture* button prompts the user to take a picture and the *Select from Gallery* button prompts the user to access photos from the device gallery. Both these actions take the user to the Image Activity after choosing or taking an image, to process the image and return it's Munsell chip value.

The *Calibrate Camera* button brings the user to the Calibrate Activity, where the image will be used to calibrate the session for different lighting environments.

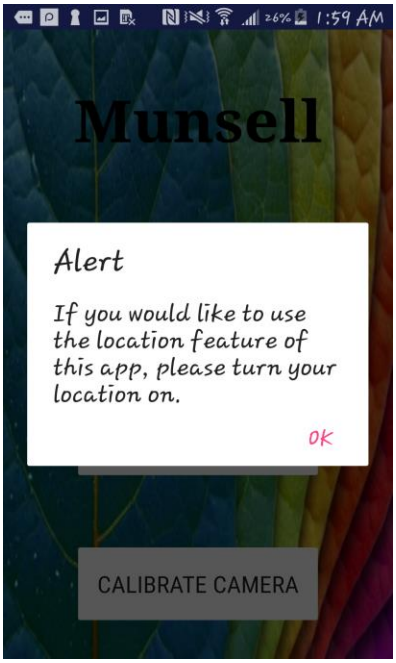


Figure 4.3.1 Main Activity

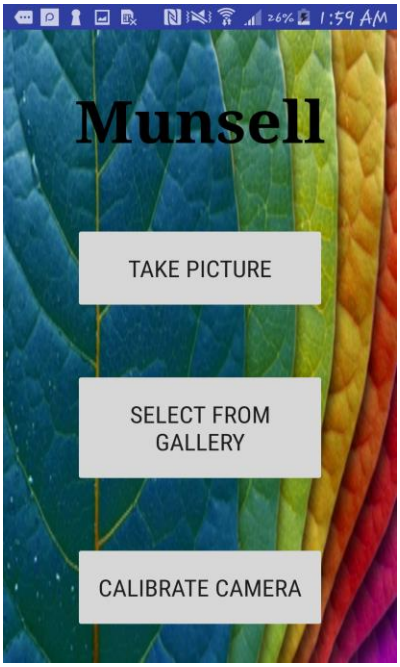


Figure 4.3.2 Main Activity

Calibrate Activity

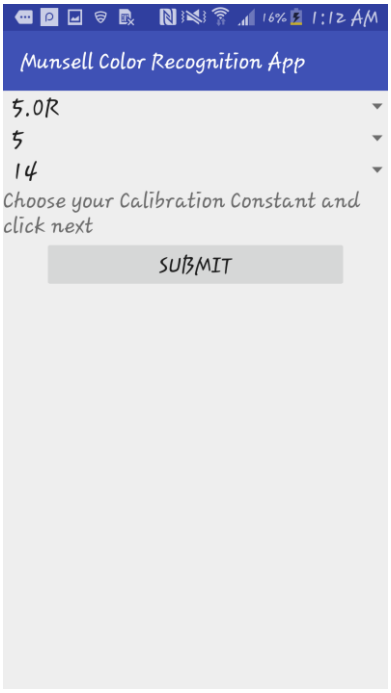


Figure 4.3.3 Calibrate Activity

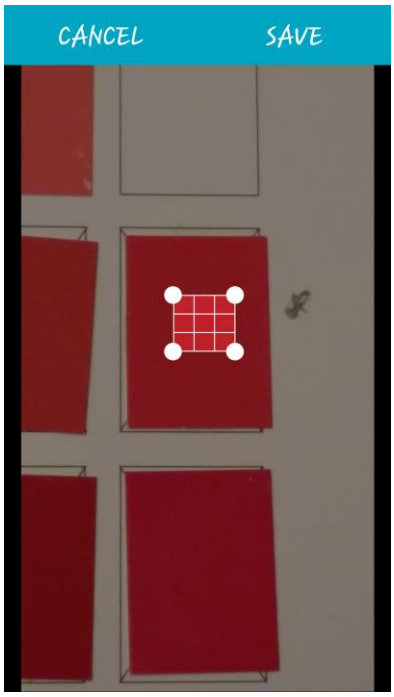


Figure 4.3.4 Taking Image of Calibration Chip 5.0R 5/14

The Calibrate Activity serves the purpose of accounting for different lighting environments when using the application. The user first selects the chip that they would like to use for calibration. The user is prompted to select their calibration Munsell chip of choice, as shown in Figure 4.3.3. and is then prompted to take a picture of the calibration chip. The user also has the ability to choose a specific part of a captured image to either calibrate or have the Munsell chip returned for. Figure 4.3.4 shows the image selection screen which can be prompted by either the calibrate button or the take picture button in the Main Activity of the application. The known RGB values of each chip are stored in the program and are compared to the RGB values of the image taken. The difference between the known RGB and the RGB of the captured image is calculated when the user clicks the calibrate button and is passed in a bundle to the Image Activity.

Figure 4.3.5 Calibrate Activity before calibration

Figure 4.3.6 Calibrate Activity after calibration

Figure 4.3.5 shows the Calibrate Activity before the user clicks the calibrate button and Figure 4.3.6 shows the calibrate activity after the user has clicked the *calibrate* button, making the *Take Picture* and *Gallery* buttons visible. When the calibrate button is clicked, the RGB factor of difference is passed through a bundle to Image Activity, to be used for detecting the Munsell Chip value of a captured image.

Image Activity

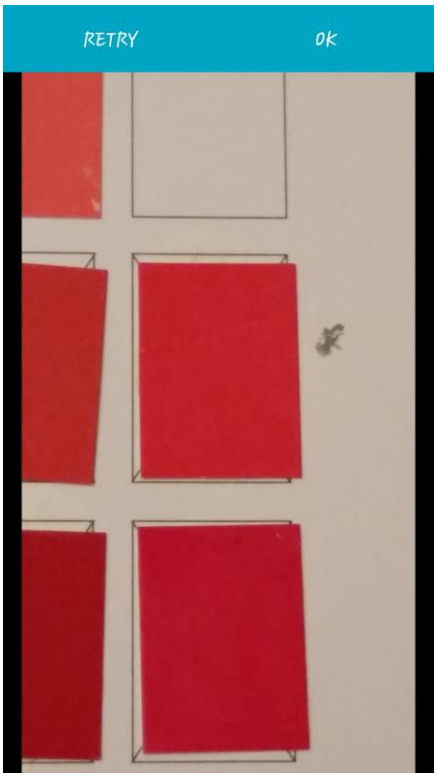


Figure 4.3.7 Image Activity

Figure 4.3.8 Image Activity Result

After the user has calibrated the application for their image capturing session, they have the option to take a picture or select an image from the gallery, from which they'll be taken to the Image Activity screen as shown in Figure 4.3.7., in which we take a picture of the calibration chip itself to verify that it's Munsell chip notation is 5.0R 5/14. The activity displays the image chosen or taken to the user, as seen in Figure 4.3.8.

Below the image are four buttons with various functionalities. The first button on the left is a *camera* button which opens up the camera intent, allowing the user to take another picture within this session, in which the calibration factors would remain the same. To the right of the camera button is the *home* button, taking the user to the home screen. Following the *home* button is the *submit* button, which takes the user to the Submit Form activity which will be discussed in further detail in the next section. The last button is the *save* button, which allows the user to screenshot the Image Activity screen and save the result to their device gallery.

Submit Form Activity

In the Image Activity, if the user clicks the *submit* button, they will be taken to the Submit Form Activity as shown in Figure 4.3.9. The purpose of this activity is to allow the user to save the data about their capturing session. The user has fields that they are able to enter information into such as ID number as well as notes. The Munsell Chip notation is passed from the previous activity and the location is also entered into the field automatically by the application. At the bottom of the activity, the user can press the save button which will add this entry to a CSV file in the Data Form activity, as shown in Figure 4.3.10.

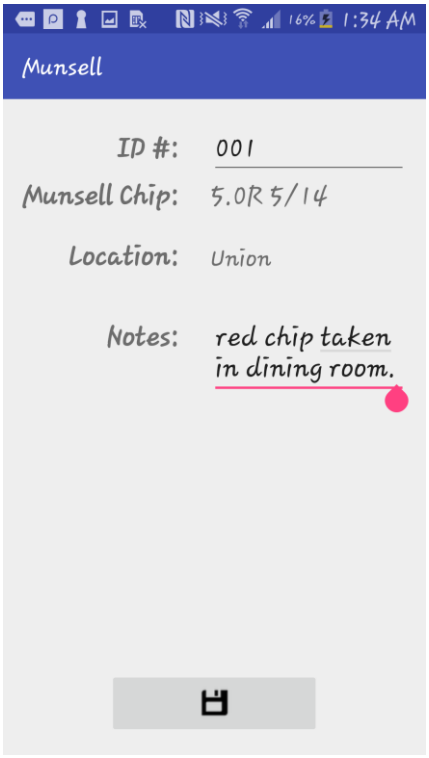


Figure 4.3.9 Submit Form Activity

Data Form Activity

After the user has clicked on the save button in the Submit Form activity, the user is taken to the Data Form activity and has the option to either to be taken back to the home screen or email the results to a specified email address. Assuming that the user will want to email their results after completing an image capturing session, the user will be warned that the application will not save the CSV file data for that session after the CSV file has been emailed, as shown in Figure 4.3.11. After clicking yes, the user will have the option to choose another application from their device to use to email their results, in which data form will be added as an attachment and will be viewable by the recipient as a csv form as shown in Figure 4.3.12.

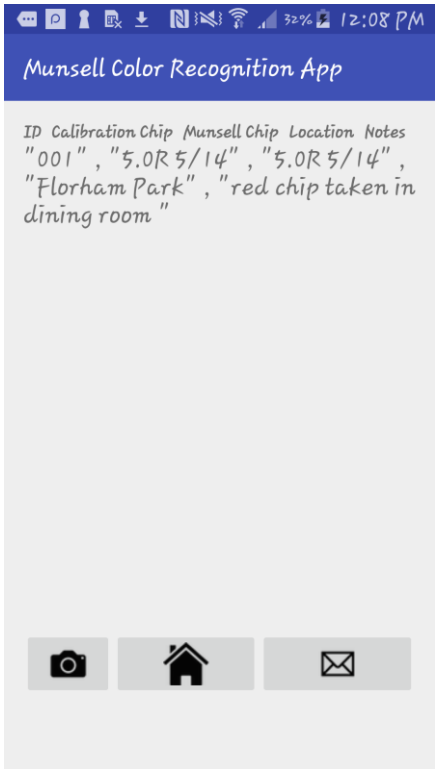


Figure 4.3.10 Data Form Activity Data List

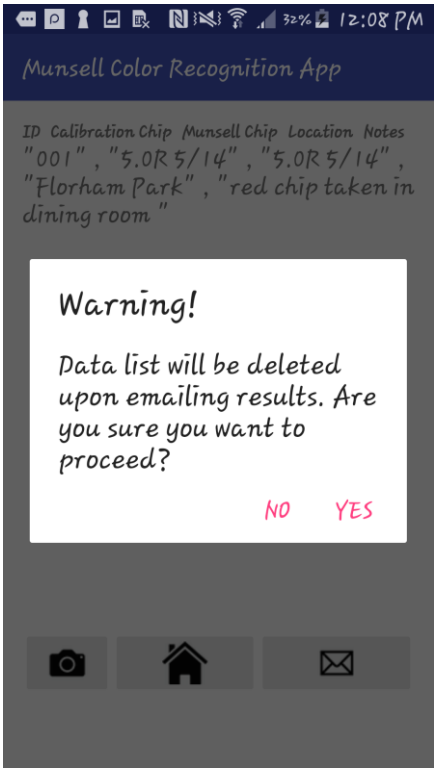


Figure 4.3.11 Data Form Activity Email Warning

```
"ID","Calibration Chip","Munsell Chip","Location","Notes"
"001" , "5.0R 5/14" , "5.0R 5/14" , "Florham Park" , "red chip taken in dining room."
```

Figure 4.3.12 Emailed CSV file

Chapter 5 Experimental Evaluation

Whether the environment is a desert or a cave, this application should be able to capture an accurate representation of color to be translated into the Munsell color system. This chapter will explore and evaluate the calibration technique depicted in Figure 5.1.1., also described in further detail in the previous chapter of this thesis. The proposed calibration technique assumes that calibrating an image using a known set of RGB coordinates, can be used to calculate an RGB factor that can be applied to any image taken within the same setting and to account for environmental factors like lighting, thus providing color constancy and allowing the application to map the correct Munsell Chip. The purpose of this evaluation is to observe and evaluate the differences in the accuracy of the application, based on the calibration chip used, providing users

of the Munsell system with guidance on how to choose appropriate calibration chips for their particular needs.

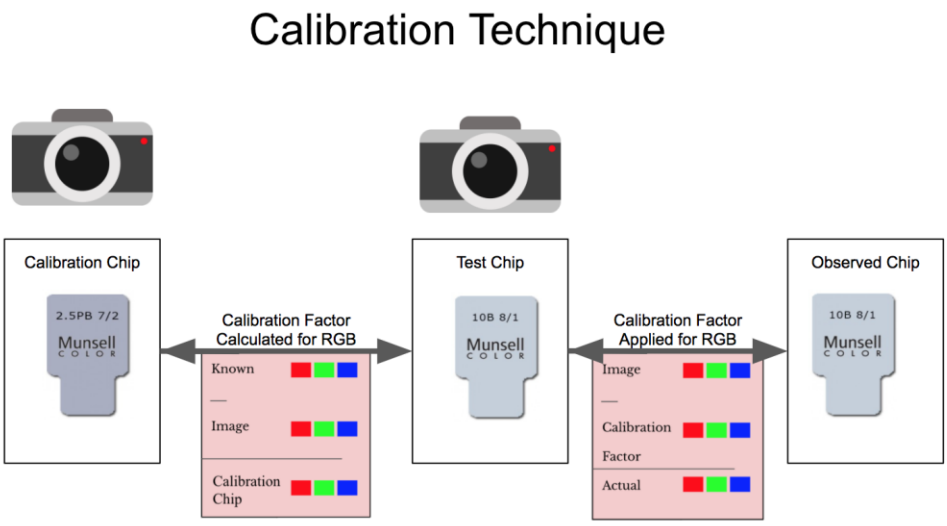


Figure 5.1.1. Proposed Calibration Technique

5.1 Research Question

The purpose of our evaluation of the Munsell Mobile Application and its color calibration technique to create color constancy focuses on evaluating the research question:

Is there a difference in the accuracy of the application based on which colored chip is used for calibration?

We hypothesize there are calibration chips for which the application yields a more accurate color prediction as well as calibration chips for which the application yields a less accurate color prediction.

5.2 Variables and Measures

The independent variable used in our study is the Munsell chip used for calibration, as we use multiple chips for calibration. The dependent variable is the accuracy of the returned Munsell chip. The measure used to evaluate accuracy is the distance between the Munsell chip that we expect (the test chip) and the Munsell chip that is returned by the app (the observed chip). The distances are measured in both Munsell and RGB units.

5.3 Experimental Procedure

Broadly speaking, the measure of how effective the application is in detecting the test chip correctly is the magnitude of the distance between the test chip and the observed chip. The tools used to run the experiment include a Samsung Galaxy S5 mobile phone running the operating system Android version 5, (better known as Lollipop), The New Munsell Student Color Set, and the mobile application proposed in this thesis. It should be noted that the student color book includes a page for each of the 10 hue groups (Red, Yellow-Red, Yellow, Green-Yellow, Green, Blue-Green, Blue, Purple, Purple-Blue, and Red-Purple). However, only one of the four variations of each hue was included in the book, and in this experiment. The hues tested were 5.0R, 5.0YR, 5.0Y, 5.0GY, 5.0G, 5.0BG, 5.0B, 5.0P, 5.0PB, and 5.0RP. In this evaluation, we use Munsell chips as the test subjects to represent objects or artifacts that users would conventionally use the application for, because the Munsell values for Munsell chips are already known, making testing for accuracy more intuitive.

We chose six Munsell chips to evaluate: red, green, blue, black, white, and grey, with their respective chips being 5.0R 5/14, 5.0G 7/10, 5.0B 7/8, N 0/0, N 5/0, and N 9/0.wechose

these specific colored chips particularly because they were the most intense and brightest chips for the red, green, and blue hues provided in the student book. The black, grey, and white chips were provided as part of a range, with black the lowest on the range, white the highest, and grey in the middle.

For each calibration chip, we chose a set of 42 Munsell chips as test points across the 10 pages of hues. Four test chips were chosen for each of the 10 pages of hues. We selected the test chips based on their distance from the brightest and most intense colors on the page, hence: one color high in chroma and low in value, one low in chroma and low in value, one low in chroma and high in value, and one high in chroma and high in value. In doing so, we increase the variability of test chips. Strategically choosing four chips from each page using this pattern, we aimed to create a representative sample. Using each calibration chip, we captured images of each test chip, utilizing the application to predict Munsell chips using the calibration technique discussed here, resulting in a dataset of 252 test and calibration chip pairs. For each pair, the following data were calculated: distance in Munsell between calibration chip and test chip, distance in RGB between calibration chip and test chip, distance in Munsell between test chip and observed chip, and distance in RGB between test chip and observed chip.

We then conducted a repeated measures ANOVA, using a significance level of .05, to determine whether mean distance (in both Munsell and RGB units) between the observed chips and test chips varied across the six calibration chips. The repeated measures ANOVA was followed by post hoc tests using the Bonferroni correction with an adjusted significance level of 0.05/15= 0.0033, given the 15 pairwise comparisons of the six calibration chips.

5.4 Results and Analysis

After conducting the tests for each calibration chip, we analyzed the resulting distances between test and observed chips, in both Munsell and RGB units. The following sections will provide a descriptive and inferential analysis of the data.

5.4.1 Munsell Analysis

After conducting the tests for each calibration chip, we analyzed the resulting distances between test and observed chips. Summaries of the distances in Munsell units can be found in Table 5.4.1 followed by an error-bar plot of the distances in Figures 5.4.1

Calibration Chip	**Mean**	**SD**
Blue 5.0B 7/8	2.39	1.15
Green 5.0G 7/10	3.33	1.62
Red 5.0R 5/14	1.68	1.30
Black N 0/0	3.69	1.02
Grey N 5/0	1.85	1.28
White N 9/0	2.59	1.40

Table 5.4.1 Means and standard deviations for the distances (in Munsell units) between the expected chip (test) and observed chip for each calibration chip.

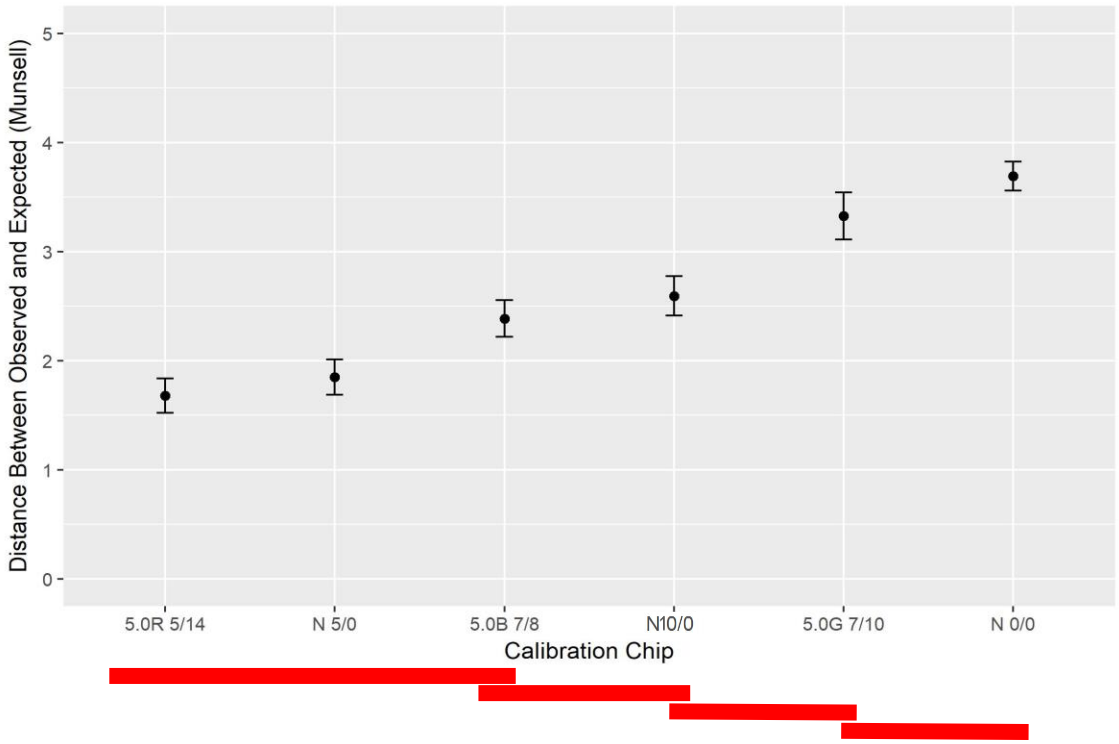


Figure 5.4.1 Plot of the mean color distance between test chip and observed chip, in Munsell units, after using each calibration chip (error bars show the within-subjects standard error of the mean) Chips labelled with the same line are not significantly different

A repeated measures ANOVA determined that mean Munsell distances between test and observed chips differed significantly based on the calibration chip used, $F(5, 205) = 21.49$, $p < 0.001$. Post hoc tests using the Bonferroni correction were then used to make pairwise comparisons. Table 5.4.2 includes the significant results.

Results of the pairwise comparisons indicated that the average distance between test and observed chips was significantly higher using the black calibration chip (M = 3.69, SD = 1.02) than the blue (M = 2.39, SD = 1.15, $p < 0.001$), grey (M = 1.85, SD = 1.28, $p < 0.001$), white (M = 2.59, SD = 1.40, $p < 0.001$), and red chips (M = 1.68, SD = 1.30, $p < 0.001$). The average

distance between test and observed chips was significantly higher using the green calibration chip (M = 3.33, SD = 1.62) than the blue (M = 2.39, SD = 1.15, $p = 0.005$), grey (M = 1.85, SD = 1.28, $p < 0.001$), and red chips (M = 1.68, SD = 1.30, $p < 0.001$). Based on the results, overall what we observed is that, for a more accurate Munsell chip identification it is better to use the red, grey, or blue chip, with white also being a reasonable choice, as opposed to the black or green chip.

Chip a vs. Chip b	**p-value**	**Confidence interval for average difference (chip a – chip b)**
Blue* vs. Green	0.005	(-1.69, -0.20)
Blue* vs. Black	< 0.001	(-1.99, -0.62)
Green vs. Red*	< 0.001	(0.79, 2.50)
Green vs. Grey*	< 0.001	(0.52, 2.43)
Red* vs. Black	< 0.001	(-2.61, -1.41)
Red* vs. White	0.008	(-1.68, -0.15)
Black vs. Grey*	< 0.001	(1.27, 2.41)
Black vs. White*	< 0.001	(0.43, 1.76)
Grey* vs. White	0.042	(-1.47, -0.01)

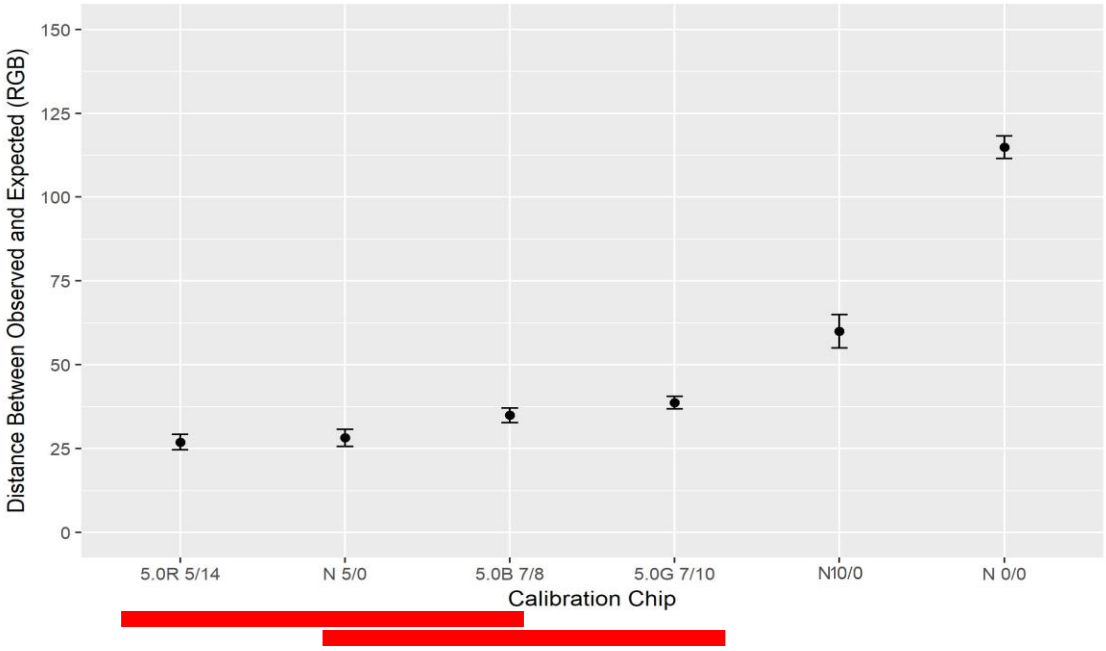
Table 5.4.2 Results of pairwise comparison tests, showing only pairs that resulted in significant differences in mean Munsell distances, with a Bonferroni correction [*= winner or significantly smaller average Munsell distance]

5.4.2 RGB Analysis

After conducting the tests for each calibration chip, we analyzed the resulting distances between test and observed chips. Summaries of the distances in RGB units can be found in Table 5.4.3 followed by an error-bar plot of the distances in Figure 5.4.2.

Calibration Chip	**Mean**	**SD**
Blue 5.0B 7/8	34.93	20.59
Green 5.0G 7/10	38.72	15.68
Red 5.0R 5/14	26.92	22.66
Black N 0/0	114.89	22.41
Grey N 5/0	28.21	21.78
White N 9/0	59.96	25.19

Table 5.4.3 Means and standard deviations for the distances (in RGB units) between the expected chip (test) and observed chip for each calibration chip



Plot of the mean color distance between test chip and observed chip, in RGB units, after using each calibration chip (error bars show the within-subjects standard error of the mean) Chips labelled with the same line are not significantly different

We then performed a repeated measures ANOVA with a Greenhouse-Geisser correction. This procedure determined that mean RGB distances between expected and observed chips differed significantly based on the calibration chip used, $F(2.54, 104.13) = 120.96$, $p < 0.001$. Post hoc tests using the Bonferroni correction were then used to make pairwise comparisons. Table 5.4.4 includes the significant results.

Chip a vs. Chip b	p-value	Confidence interval for average difference (chip a – chip b)
Blue* vs. Black	< 0.001	(-91.95, -67.97)
Blue* vs. White	0.001	(-42.93, -7.14)
Green vs. Red*	0.003	(2.91, 20.69)
Green* vs. Black	< 0.001	(-88.43, -63.91)
Green* vs. White	0.001	(-35.97, -6.52)
Red* vs. Black	< 0.001	(-100.34, -75.59)
Red* vs. White	< 0.001	(-51.98, -14.10)
Black vs. Grey*	< 0.001	(74.80, 98.55)
Black vs. White*	< 0.001	(34.61, 75.24)
Grey* vs. White	< 0.001	(-50.90, -12.61)

Table 5.4.4 Results of pairwise comparison tests, showing only pairs that resulted in significant differences in mean RGB distances, with a Bonferroni correction [*= winner or significantly smaller average RGB distance]

Results of the pairwise comparisons indicated that the average distance between test and observed chips was significantly higher using the black calibration chip (M = 114.89, SD = 22.41) than the blue (M = 34.93, SD = 20.59, $p < 0.001$), grey (M = 28.21, SD = 21.78, $p <$

0.001), white (M = 59.96, SD = 25.19, $p < 0.001$), red (M = 26.92, SD = 22.66, $p < 0.001$), and green chips (M = 38.72, SD = 115.68, $p < 0.001$). The average distance between test and observed chips was significantly higher using the white chip (M =59.96, SD = 25.19) than the blue (M = 34.93, SD = 20.59, $p = 0.001$), green (M = 38.72, SD = 115.68, $p = 0.001$), red (M = 26.92, SD = 22.66, $p < 0.001$), and grey chips (M = 28.21, SD = 21.78, $p < 0.001$). Based on the results, overall what we saw is that, for a more accurate Munsell chip identification it is better to use the red, grey, green, or blue chip, as opposed to the black, or white chip for calibration.

5.4.3 Overall Conclusion

Based on the results of mean distances in both Munsell and RGB units, overall what we can conclude is that it is better to use the red, grey, or blue chips, as opposed to the black, white, or green chips for calibration for a more accurate Munsell chip identification. The red, blue, and grey chips yielded significantly lower distances (more accurate Munsell color predictions) than the black, white, and green chips for the current prototype of this application. In the experiments conducted, we found the white calibration chip to perform significantly worse than the blue, red, and grey chips.

I noted in a previous chapter that the Munsell Soil System application currently uses a white chip for calibration. It would be ideal to share the results of this experiment with the researchers and developers, so that they could possibly optimize their calibration feature to use a red, grey, or blue chip, given that in this experiment, these calibration chips performed better than the white chip in identifying browns and beiges, common soil colors. Though there was no strong correlation found between the distances between calibration distance and observed chip, and observed chip and expected chip, we did find that there tended to be smaller distances between expected and observed, when the calibration chip used was a color close to the

dominant color of the object. As for why the green calibration chip, performed significantly worse than the other two colored chips, red and blue, I hypothesize that this has to do with the Bayer filter used in phone sensors. The Bayer filter is a color filter array which color sensors used to filter out the components of red, green, and blue in an image. Because the human eye is particularly sensitive to the green light, the Bayer filter mimics this physiology by having twice as many green filters than blue or red. I believe this heightened sensitivity to green somehow distorts the way in which green color is being processed, which would be something to test in future experiments.

5.5 Threats to Validity

A possible threat to the validity of this experiment is the effect of light on the image captured by the application, as different light sources can change the colors that the device sensors read. We attempted to minimize this threat by conducting testing in a room with no natural sunlight or windows, using only fluorescent lighting. This minimizes the threat by avoiding the need to re-calibrate, assuming that the fluorescent light source did not change.

Another potential threat to validity and a difficulty faced in this experiment was the use of only 10 out of the 40 possible hues. To increase generalizability, we chose four test chips from each of the 10 hues, varying in chroma and value. A threat that unfortunately we could not minimize, is the fact that although hue and value have set scales in Munsell, chroma has no arbitrary end, as some fluorescent materials have chroma as high as 30. Currently, the application maps from RGB to Munsell for Munsell chips with chroma values up to 18.

The camera used for testing also poses a threat to the validity of this experiment and its generalizability, as a different camera might yield different results. Though cameras may have the same megapixel count, elements like image sensors, lenses, and the ways in which smartphones interpret the data gathered by the sensor to adjust the final image can all differ.

Chapter 6 Conclusions

In this thesis, we have presented the Munsell Mobile application: a mobile application that allows users to use a mobile device to take a picture of an object or artifact of their choice and estimate the Munsell Chip notation of that object or artifact. We have also presented a method for calibration that involves RGB transformations. We have implemented both the application and the calibration technique and have tested the calibration technique on a sample of Munsell chips used to mimic other objects or artifacts. The Munsell Mobile application serves as a basis for a tool that could potentially replace the use of Munsell Color Books.

The success of this application is heavily reliant on the ability to calibrate the application when a change in the lighting environment occurs. Our goal in this thesis was to see if there was a difference in the accuracy of the application based on the calibration chip used, which we found to be the case.

6.1 Future Experimental Developments

The immediate next steps for this research are to expand our sample to include test chips from a larger selection of Munsell hues and to test the same test points and calibration chips in multiple lighting scenarios. In doing so, we would like to see if the results of this evaluation hold true in different environments. In expanding our sample of Munsell chips, we would also like to expand the number of chips used for calibration. With a larger set of calibration chips, we would eventually like to experiment with using multiple calibration chips at once. For example, rather than individually calibrating with a red chip for a particular test chip, we would like to see if there is a significant difference in distances between test and observed chips when using multiple calibration chips at once, which would inherently modify the calibration technique. In expanding the scope of our experiment, we would also like to conduct these tests using different cameras, to observe whether the camera used and quality of image, affects the application's ability to correctly identify Munsell color. With continued research, we hope to improve the application as well as the calibration technique discussed, so that anyone with a mobile device is able to utilize Munsell color notation for their specific needs.

6.2 Future Application Developments

As of now, the presented application is compatible with Android devices, and a proposed future development is to integrate the application to be compatible with iOS devices as well, increasing the scope of possible users. Another possible development that would diffuse the issue of compatibility is to create a web application version, in such a way that users can access the application through the browser on their mobile devices.

As discussed, Munsell chips are not easily accessible, and thus finding single Munsell chips to use for calibration is not ideal. Adding a feature to the application to allow users to use a known RGB paint chip or card to calibrate the application would be ideal, as RGB paint chips are much more accessible than individual Munsell chips. Other possible developments include adding accessibility features for visually impaired users, conversions from multiple color standards to Munsell, and improving the current features as well. All of which seek to make the use of Munsell color notation more accessible to the general public.

Bibliography

[1] Wyszecki, Gunter, and W. S. Stiles. *COLOR SCIENCE Concepts and Methods, Quantitative Data and Formulae, 2nd Edition*. 2nd ed., John Wiley and Sons, 2000.

[2] Nickerson, Dorothy. "History of the Munsell Color System and Its Scientific Application." *Journal of the Optical Society of America*, vol. 30, no. 12, Jan. 1940, p. 575., doi:10.1364/josa.30.000575.

[3] Munsell, A. H., and A. E. O. Munsell. *A Color Notation*. Munsell Color Co., 1946. pp. 7-20

[4] "Munsell Color Order System: What Is It and How Is It Used?" *Munsell Color System; Color Matching from Munsell Color Company*, 15 May 2013, munsell.com/color-blog/munsell-color-order-system-what-is-it-and-how-is-it-used/.

[5] "Development of the Munsell Color Order System." *Munsell Color System; Color Matching from Munsell Color Company*, 12 June 2012, munsell.com/about-munsell-color/development-of-the-munsell-color-order-system/.

[6] "Munsell Hue; 3 Dimensions of Color." *Munsell Color System; Color Matching from Munsell Color Company*, 16 Mar. 2018, munsell.com/about-munsell-color/how-color-notation-works/munsell-hue/.

[7] "Munsell Value Scale; 3 Dimensions of Color." *Munsell Color System; Color Matching from Munsell Color Company*, 16 Mar. 2018, munsell.com/about-munsell-color/how-color-notation-works/munsell-value/.

[8] “Munsell Chroma; 3 Dimensions of Color.” *Munsell Color System; Color Matching from Munsell Color Company*, 16 Mar. 2018, munsell.com/about-munsell-color/how-color-notation-works/munsell-chroma/.

[9] “How to Read a Munsell Color Chart.” *Munsell Color System; Color Matching from Munsell Color Company*, 22 Mar. 2017, munsell.com/about-munsell-color/how-color-notation-works/how-to-read-color-chart/.

[10] Lyon, Richard F, and Paul M Hubel. “Eyeing the Camera: Into the Next Century.”*Color and Imaging Conference*, 1 Jan. 2002.

[11] “Android.graphics.” *Android Developers*, 3 Apr. 2018, developer.android.com/reference/android/graphics/package-summary.html.

[12] “Android.graphics.Color.” *Android Developers*, 3 Apr. 2018, developer.android.com/reference/android/graphics/Color.html.

[13] Weng, Ching-Chih, et al. “A Novel Automatic White Balance Method For Digital Still Cameras.” *2005 IEEE International Symposium on Circuits and Systems*, doi:10.1109/iscas.2005.1465458.

[14] Stiglitz, Roxanne, et al. “Soil Color Sensor Data Collection Using a GPS-Enabled Smartphone Application.” *Geoderma*, vol. 296, 2017, pp. 108–114., doi:10.1016/j.geoderma.2017.02.018.

[15] Gómez-Robledo, Luis, et al. “Using the Mobile Phone as Munsell Soil-Colour Sensor: An Experiment under Controlled Illumination Conditions.” *Computers and Electronics in Agriculture*, vol. 99, 2013, pp. 200–208., doi:10.1016/j.compag.2013.10.002.

[16] Finlayson, Graham D., and Steven D. Hordley. “Color Constancy at a Pixel.” *Journal of the Optical Society of America A*, vol. 18, no. 2, Jan. 2001, p. 253., doi:10.1364/josaa.18.000253.

[17] Ruck, Lana, and Clifford T. Brown. “Quantitative Analysis of Munsell Color Data from Archeological Ceramics.” *Journal of Archaeological Science: Reports*, vol. 3, 2015, pp. 549–557., doi:10.1016/j.jasrep.2015.08.014.

[18] Meier, Reto. *Professional Android 4 Application Development*. John Wiley & Sons, 2012.

[19] “Package Index.” *Android Developers*, 7 Mar. 2018, developer.android.com/reference/packages.html.

[20] Matos, Victor, and Rebecca Grasser. “Building Applications for the Android OS Mobile Platform: A Primer and Course Materials.” *JCSC*, 2010.

Appendix A : Main activity source code

```
package com.munsellapp.munsellcolorrecognitionapp;

import android.app.Activity;
import android.content.ActivityNotFoundException;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.support.v7.app.AlertDialog;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import java.io.ByteArrayOutputStream;
import java.io.IOException;

/* THis activity displays the main screen of the app, with three buttons (Take
picture, Select from gallery and calibrate camera)
NOTE calibrate camera button does not work yet, will have to be integrated when
ImageSelction and Calibration activities work
as expected. Take Picture Button and Select from Gallery Button will prompt the camera
intent, which will pass the image to Image
Activity. NOTE select from gallery option currently only works on images that were not
previously taken by the camera, but does
work for images that are downloaded. */
public class MainActivity extends Activity implements View.OnClickListener {


    private static int TAKE_PIC = 0;
    private static int CALIBRATE_PIC=2;
    private static int SELECT_FILE = 1;
    final int CROP_PIC = 3;
    final int CROPCALI_PIC = 4;
    private Button chooseImage;
    private Uri photo;
    private Uri caliPhoto;
    protected final static String TAG = "ColorUtils";




    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        chooseImage = (Button) findViewById(R.id.ChooseImage);
        chooseImage.setOnClickListener(this);

        new AlertDialog.Builder(MainActivity.this)
                .setTitle("Alert")
                .setMessage("If you would like to use the location feature of this
app, please turn your" +
                        " location on.")
                .setPositiveButton("OK", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int which) {
```

```
                }
            })

            .show();



    }


    /*Starts camera Intent -JB*/
    public void CameraClick(View v) {
        Intent intent2 = new Intent(this, Calibrate.class);
        startActivity(intent2);

        try {
            Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
            startActivityForResult(intent, TAKE_PIC);
        } catch (ActivityNotFoundException anfe) {
            Toast toast = Toast.makeText(this, "This device doesn't support the crop
action!",
                    Toast.LENGTH_SHORT);
            toast.show();
        }
    }

    /*Starts camera Intent -JB*/
    public void CalibrateCameraClick(View v) {
        Intent my = new Intent(this, CalibrateHome.class);
        startActivity(my);

    }

    /*Opens gallery view, then sets Result Code signaling that
    image has been selected -JB
     */
    private void galleryIntent()
    {
        Intent intent = new Intent();
        intent.setType("image/*");
        intent.setAction(Intent.ACTION_GET_CONTENT);//
        startActivityForResult(Intent.createChooser(intent, "Select
File"),SELECT_FILE);
    }


    /*Gets result code from camera and gallery intents
    if result is from camera intent, sends image to ImageActivity;
    if result is from gallery intent, calls function to send image to
    ImageActivity -JB
      */
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {

        if (requestCode == TAKE_PIC && resultCode == RESULT_OK) {
             photo = data.getData();
             performCrop();
           // PassBitmapToNextActivity(photo,ImageActivity.class,"CameraImage");
        }

        if (requestCode == SELECT_FILE && resultCode == RESULT_OK) {
            Bitmap bm;
            if (data != null) {
```

```
                try {
                    bm =
MediaStore.Images.Media.getBitmap(getApplicationContext().getContentResolver(),
data.getData());
                    PassBitmapToNextActivity(bm,ImageActivity.class,"GalleryImage");
                } catch (IOException e) {
                    e.printStackTrace();
                }}}

        if (requestCode == CALIBRATE_PIC && resultCode == RESULT_OK){
            caliPhoto = data.getData();
            performCalibrationCrop();
        }
        if (requestCode == CROPCALI_PIC) {
            // get the returned data
            Bundle extras = data.getExtras();
            // get the cropped bitmap
            Bitmap theCaliPic = extras.getParcelable("data");
            PassBitmapToNextActivity(theCaliPic,Calibrate.class,"CalibrateImage");
        }
        else if (requestCode == CROP_PIC) {
            // get the returned data
            Bundle extras = data.getExtras();
            // get the cropped bitmap
            Bitmap thePic = extras.getParcelable("data");
            PassBitmapToNextActivity(thePic,ImageActivity.class,"CameraImage");

        }
    }

    /*Passes Bitmap from any intent (camera, gallery, or calibrate camera) and passes
it to specified activity)*/
    public void PassBitmapToNextActivity (Bitmap bm, Class myClass, String extraName
){
        Intent intent = new Intent(this, myClass);
        ByteArrayOutputStream stream = new ByteArrayOutputStream();
        bm.compress(Bitmap.CompressFormat.PNG, 100, stream);
        intent.putExtra(extraName, stream.toByteArray());
        startActivity(intent);

    }

    /*
    Either calls galleryIntent when ChooseImage button is clicked
    or opens cameraIntent -JB
     */
    @Override
    public void onClick(View view) {
        if (view.getId() == R.id.ChooseImage) {
            galleryIntent();
        } else {
            Intent intent=new Intent(this, ImageActivity.class);
            startActivity(intent);

        }


    }
    private void performCrop() {
        // take care of exceptions
        try {
            // call the standard crop action intent (the user device may not
            // support it)
```

```
            Intent cropIntent = new Intent("com.android.camera.action.CROP");
            // indicate image type and Uri
            cropIntent.setDataAndType(photo, "image/*");
            // set crop properties
            cropIntent.putExtra("crop", "true");
            // indicate aspect of desired crop
            cropIntent.putExtra("aspectX", 2);
            cropIntent.putExtra("aspectY", 1);
            // indicate output X and Y
            cropIntent.putExtra("outputX", 256);
            cropIntent.putExtra("outputY", 256);
            // retrieve data on return
            cropIntent.putExtra("return-data", true);
            // start the activity - we handle returning in onActivityResult
            startActivityForResult(cropIntent, CROP_PIC);
        }
        // respond to users whose devices do not support the crop action
        catch (ActivityNotFoundException anfe) {
            Toast toast = Toast
                    .makeText(this, "This device doesn't support the crop action!",
Toast.LENGTH_SHORT);
            toast.show();
        }
    }

    private void performCalibrationCrop() {
        // take care of exceptions
        try {
            // call the standard crop action intent (the user device may not
            // support it)
            Intent cropIntent = new Intent("com.android.camera.action.CROP");
            // indicate image type and Uri
            cropIntent.setDataAndType(caliPhoto, "image/*");
            // set crop properties
            cropIntent.putExtra("crop", "true");
            // indicate aspect of desired crop
            cropIntent.putExtra("aspectX", 2);
            cropIntent.putExtra("aspectY", 1);
            // indicate output X and Y
            cropIntent.putExtra("outputX", 256);
            cropIntent.putExtra("outputY", 256);
            // retrieve data on return
            cropIntent.putExtra("return-data", true);
            // start the activity - we handle returning in onActivityResult
            startActivityForResult(cropIntent, CROPCALI_PIC);
        }
        // respond to users whose devices do not support the crop action
        catch (ActivityNotFoundException anfe) {
            Toast toast = Toast
                    .makeText(this, "This device doesn't support the crop action!",
Toast.LENGTH_SHORT);
            toast.show();
        }
    }


}
```

Appendix B : Calibration home activity source code

```
package com.munsellapp.munsellcolorrecognitionapp;

import android.content.ActivityNotFoundException;
import android.content.Intent;
import android.graphics.Bitmap;
import android.net.Uri;
import android.provider.MediaStore;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.Toast;

import com.opencsv.CSVReader;

import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;

import static com.munsellapp.munsellcolorrecognitionapp.R.id.button5;
import static com.munsellapp.munsellcolorrecognitionapp.R.id.spinner5;

import static com.munsellapp.munsellcolorrecognitionapp.R.id.spinner4;
import static com.munsellapp.munsellcolorrecognitionapp.R.id.spinner6;
// This class allows the user to choose a chip to use for calibration, then
//prompts them to take a picture of the chosen chip

public class CalibrateHome  extends AppCompatActivity implements View.OnClickListener{
    Spinner expectedHue;
    Spinner expectedValue;
    Spinner expectedChroma;
    String expectedHueString, expectedValueString, expectedChromaString;
    String[] line4;
    int actualRed, actualGreen, actualBlue;

    private static int TAKE_PIC = 0;
    private static int CALIBRATE_PIC=2;
    private static int SELECT_FILE = 1;
    final int CROPCALI_PIC = 4;
    private ImageView imageView, img;
    private Button next;
    private Uri caliPhoto;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_calibrate_home);
        expectedHue = (Spinner) findViewById(spinner5);
        expectedValue = (Spinner) findViewById(spinner6);
        expectedChroma = (Spinner) findViewById(spinner4);
        next=(Button) findViewById(button5);
        next.setOnClickListener(this);

    }
    public void PassBitmapToNextActivity (Bitmap bm, Class myClass, String extraName
```

```
){
        Intent intent = new Intent(this, myClass);
        ByteArrayOutputStream stream = new ByteArrayOutputStream();
        bm.compress(Bitmap.CompressFormat.PNG, 100, stream);
        intent.putExtra(extraName, stream.toByteArray());
        Bundle bundle=new Bundle();
        bundle.putString("actualRed", Integer.toString(actualRed));
        bundle.putString("actualGreen", Integer.toString(actualGreen));
        bundle.putString("actualBlue", Integer.toString(actualBlue));
        bundle.putString("calibrateHueString", expectedHueString);
        bundle.putString("calibrateValueString", expectedValueString);
        bundle.putString("calibrateChromaString", expectedChromaString);
        System.out.println("actual red: "+ actualRed +" actual green: "+ actualGreen
+" actual blue:  "+ actualBlue);

//        if(updatedText.equals("")){
        intent.putExtras(bundle);
        startActivity(intent);

    }
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {


        if (requestCode == CALIBRATE_PIC && resultCode == RESULT_OK){
            caliPhoto = data.getData();
            performCaliCrop();
        }
        if (requestCode == CROPCALI_PIC) {
            // get the returned data
            Bundle extras = data.getExtras();
            // get the cropped bitmap
            Bitmap theCaliPic = extras.getParcelable("data");
            PassBitmapToNextActivity(theCaliPic,Calibrate.class,"CalibrateImage");
        }

    }
    private void performCaliCrop() {
        // take care of exceptions
        try {
            // call the standard crop action intent (the user device may not
            // support it)
            Intent cropIntent = new Intent("com.android.camera.action.CROP");
            // indicate image type and Uri
            cropIntent.setDataAndType(caliPhoto, "image/*");
            // set crop properties
            cropIntent.putExtra("crop", "true");
            // indicate aspect of desired crop
            cropIntent.putExtra("aspectX", 2);
            cropIntent.putExtra("aspectY", 1);
            // indicate output X and Y
            cropIntent.putExtra("outputX", 256);
            cropIntent.putExtra("outputY", 256);
            // retrieve data on return
            cropIntent.putExtra("return-data", true);
            // start the activity - we handle returning in onActivityResult
            startActivityForResult(cropIntent, CROPCALI_PIC);
        }
        // respond to users whose devices do not support the crop action
        catch (ActivityNotFoundException anfe) {
            Toast toast = Toast
                    .makeText(this, "This device doesn't support the crop action!",
Toast.LENGTH_SHORT);
```

```
            toast.show();
        }
    }
    public void getRGB() throws IOException {
        // System.out.println("expectedssssss"+expectedRed+" "+expectedGreen+" "+
expectedBlue);

        InputStream csv4;

        csv4 = getAssets().open("munsell.csv");

        InputStreamReader is4 = new InputStreamReader(csv4);

        CSVReader csvReader4 = new CSVReader(is4);
        expectedHueString = expectedHue.getSelectedItem().toString();
        expectedChromaString = expectedChroma.getSelectedItem().toString();
        expectedValueString = expectedValue.getSelectedItem().toString();
        csvReader4.readNext();
        while ((line4 = csvReader4.readNext()) != null) {
            if (expectedHueString.equals((line4[0]))) {
                if (expectedValueString.equals((line4[1]))) {
                    if (expectedChromaString.equals((line4[2]))) {
                        actualRed = Integer.parseInt(line4[3]);
                        actualGreen = Integer.parseInt(line4[4]);
                        actualBlue = Integer.parseInt(line4[5]);

                    }
                }
            }
        }
        csvReader4.close();
        System.out.println("expectedssssss+"+actualRed+" "+actualGreen+" "+
actualBlue);
    }

    @Override
    public void onClick(View v) {
        try {
            getRGB();
        } catch (IOException e) {
            e.printStackTrace();
        }


        try {
            Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
            startActivityForResult(intent, CALIBRATE_PIC);
        }catch (ActivityNotFoundException anfe) {
            Toast toast = Toast.makeText(this, "This device doesn't support the crop
action!",
                    Toast.LENGTH_SHORT);
            toast.show();
        }
    }
}
```

Appendix C : Calibration activity source code

```
package com.munsellapp.munsellcolorrecognitionapp;

import android.content.ActivityNotFoundException;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.graphics.drawable.BitmapDrawable;
import android.net.Uri;
import android.provider.MediaStore;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.Toast;

import java.io.ByteArrayOutputStream;
import java.io.IOException;

// This class recieves a bitmap image of a Munsell Chip passed from CalibrateHome,
// which will be compared to it's known Munsell
//RGB values, and an adjustment will be made to the RGB value of image to match what
the RGB values
//are supposed to be. This adustment will be mad on all RGB-Munsell combinations
according to the environment
//each time the camera is calibrated.
public class Calibrate extends AppCompatActivity implements View.OnClickListener {

    Bitmap b;
    ImageView caliPic;
    int actualRed, actualGreen, actualBlue;
    int specRed, specGreen, specBlue;
    static int fixRed, fixGreen, fixBlue;
    ImageButton calibrateButton;
    Button takePic ,choosePic;
    private static int TAKE_PIC = 0;
    private static int SELECT_FILE = 1;
    private Uri photo;
    final int CROP_PIC = 3;
    String calibrateHue,calibrateValue,calibrateChroma;




    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_calibrate);
        caliPic=(ImageView) findViewById(R.id.caliPic);
        calibrateButton=(ImageButton) findViewById(R.id.calibrateImageButton);
        calibrateButton.setOnClickListener(this);
        takePic = (Button) findViewById(R.id.button);
        choosePic = (Button) findViewById(R.id.button2);
        Bundle extras = getIntent().getExtras();
        actualRed=Integer.parseInt(extras.getString("actualRed"));
        actualGreen=Integer.parseInt(extras.getString("actualGreen"));
```

```
        actualBlue=Integer.parseInt(extras.getString("actualBlue"));
        calibrateHue=extras.getString("calibrateHueString");
        calibrateValue=extras.getString("calibrateValueString");
        calibrateChroma=extras.getString("calibrateChromaString");
        System.out.println("actual red: "+ actualRed +" actual green: "+ actualGreen
+" actual blue:  "+ actualBlue);
        if (getIntent().hasExtra("CalibrateImage")) {
            b = BitmapFactory.decodeByteArray(
                    getIntent().getByteArrayExtra("CalibrateImage"), 0,
getIntent().getByteArrayExtra("CalibrateImage").length);

            caliPic.setImageBitmap(b);
        }

    }


    public void getSpecs() {
        //When implementing with camera, change field i to get the image taken from
the camera,
        //so it's no longer pre loaded in with Android Studio
        ImageView i = new ImageView(this);
        i.setImageBitmap(b);
        Bitmap bitmap = ((BitmapDrawable) i.getDrawable()).getBitmap();
        int w = bitmap.getWidth();
        int h = bitmap.getHeight();
        int count=0;
        for(int x = 0; x< w-1; x++){
            for(int y = 0; y< h-1; y++){
                int pixel = bitmap.getPixel(x, y);
                specRed += Color.red(pixel);
                specBlue += Color.blue(pixel);
                specGreen += Color.green(pixel);
                count++;
                //

            }
        }

        specRed = specRed/count;
        specBlue = specBlue/count;
        specGreen = specGreen/count;
       // System.out.println("calibrate image red: "+ specRed + " calibrate image
green: "+ specGreen +" calibrate image blue: "+ specBlue);


    }

    public void fixColors(int Red, int Green, int Blue){
        fixRed=actualRed-Red;
        fixGreen=actualGreen-Green;
        fixBlue=actualBlue-Blue;
    }

    @Override
    public void onClick(View v) {
        if (v.getId()==R.id.calibrateImageButton){
            choosePic.setVisibility(View.VISIBLE);
            takePic.setVisibility(View.VISIBLE);getSpecs();
            fixColors(specRed,specGreen,specBlue);
        }
    }
```

```
    /*Starts camera Intent -JB*/
    public void CamClick(View v) {
        try {
            Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
            startActivityForResult(intent, TAKE_PIC);
            //  will handle the returned data in onActivityResult
        } catch (ActivityNotFoundException anfe) {
            Toast toast = Toast.makeText(this, "This device doesn't support the crop
action!",
                    Toast.LENGTH_SHORT);
            toast.show();
        }
    }

    public void galleryIntent(View v)
    {
        Intent intent = new Intent();
        intent.setType("image/*");
        intent.setAction(Intent.ACTION_GET_CONTENT);//
        startActivityForResult(Intent.createChooser(intent, "Select
File"),SELECT_FILE);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {

        if (requestCode == TAKE_PIC && resultCode == RESULT_OK) {
            photo = data.getData();
            performCrop();
        }

        if (requestCode == SELECT_FILE && resultCode == RESULT_OK) {
            Bitmap bm;
            if (data != null) {
                try {
                    bm =
MediaStore.Images.Media.getBitmap(getApplicationContext().getContentResolver(),
data.getData());
                    PassBitmapToNextActivity(bm,ImageActivity.class,"GalleryImage");
                } catch (IOException e) {
                    e.printStackTrace();
                }}}
        else if (requestCode == CROP_PIC) {
            // get the returned data
            Bundle extras = data.getExtras();
            // get the cropped bitmap
            Bitmap thePic = extras.getParcelable("data");
            PassBitmapToNextActivity(thePic,ImageActivity.class,"CameraImage");

        }
    }
    /*Passes Bitmap from any intent (camera, gallery, or calibrate camera) and passes
it to specified activity)*/
    public void PassBitmapToNextActivity (Bitmap bm, Class myClass, String extraName
){
        Intent intent = new Intent(this, myClass);
        ByteArrayOutputStream stream = new ByteArrayOutputStream();
        bm.compress(Bitmap.CompressFormat.PNG, 100, stream);
        intent.putExtra(extraName, stream.toByteArray());
        intent.putExtra("fixRed",fixRed);
        intent.putExtra("fixBlue",fixBlue);
        intent.putExtra("fixGreen",fixGreen);
        intent.putExtra("calibrateRed",actualRed);
```

```
        intent.putExtra("calibrateGreen",actualGreen);
        intent.putExtra("calibrateBlue",actualBlue);
        intent.putExtra("calibrateHue",calibrateHue);
        intent.putExtra("calibrateValue",calibrateValue);
        intent.putExtra("calibrateChroma",calibrateChroma);



        startActivity(intent);

    }

    private void performCrop() {
        try {
            // call the standard crop action intent (the user device may not
            // support it)
            Intent cropIntent = new Intent("com.android.camera.action.CROP");
            // indicate image type and Uri
            cropIntent.setDataAndType(photo, "image/*");
            // set crop properties
            cropIntent.putExtra("crop", "true");
            // indicate aspect of desired crop
            cropIntent.putExtra("aspectX", 2);
            cropIntent.putExtra("aspectY", 1);
            // indicate output X and Y
            cropIntent.putExtra("outputX", 256);
            cropIntent.putExtra("outputY", 256);
            // retrieve data on return
            cropIntent.putExtra("return-data", true);
            // start the activity - we handle returning in onActivityResult
            startActivityForResult(cropIntent, CROP_PIC);
        }
        // respond to users whose devices do not support the crop action
        catch (ActivityNotFoundException anfe) {
            Toast toast = Toast
                    .makeText(this, "This device doesn't support the crop action!",
Toast.LENGTH_SHORT);
            toast.show();
        }
    }



}
```

Appendix D : Image activity source code

```
package com.munsellapp.munsellcolorrecognitionapp;

import android.content.ActivityNotFoundException;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.graphics.drawable.BitmapDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.RelativeLayout;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
import com.opencsv.CSVReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.text.SimpleDateFormat;
import java.util.Date;
import static com.munsellapp.munsellcolorrecognitionapp.R.id.spinner;
import static com.munsellapp.munsellcolorrecognitionapp.R.id.spinner2;
import static com.munsellapp.munsellcolorrecognitionapp.R.id.spinner3;


//This class takes the image selected from gallery or take by the camera, get's the
image's munsell value,
//displays it on the screen, and changes the background color to match the Munsell
value. Three buttons
//appear at the bottom , home button, submit button ( whic will take the user to the
submitForm activity ), and save
//button (which will screenshot the result and save the image in an album labeled
"photos" in the device gallery)
public class ImageActivity extends AppCompatActivity implements View.OnClickListener {
    private Button calibrate;
    static int TAKE_ANOTHERPIC = 0;
    private ImageView ResultPic;
    private ImageButton saveresult, exportresult, home, takeAnotherPic;
    Bitmap b;
    String munsellValue;
    TextView munsellChip, backgroundWarning;
    TextView iaDataStorage;
    RelativeLayout R1;
    Double smallestDif;
    int fixRed, fixGreen, fixBlue;
    private Uri photo;
    int compareRed, compareGreen, compareBlue;
    final int CROP_PIC = 3;
    int smallRed, smallGreen, smallBlue;
    int calibrateRed,calibrateGreen,calibrateBlue;
    Spinner expectedHue, expectedValue, expectedChroma;
    String expectedHueValue, expectedValueValue, expectedChromaValue;
    int red;
```

```
    int green;
    int blue;
    int i;
    String[] line3;
    String[] line5;
    int expectedRed, expectedGreen, expectedBlue;
    String foundMunsellHue, foundMunsellValue, foundMunsellChroma;
    Double rgbDistancedouble;
    String rgbDistance;
    int finalRed, finalGreen, finalBlue;
    String calibrateHue,calibrateValue,calibrateChroma;
    double calibratergbDistanceDouble;
    String calibratergbDistance;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.image_layout);
        R1=(RelativeLayout)findViewById(R.id.R1);
        home = (ImageButton) findViewById(R.id.homeButton);
        home.setOnClickListener(this);
        calibrate = (Button) findViewById(R.id.button3);
        ResultPic = (ImageView) findViewById(R.id.imageView1);
        saveresult = (ImageButton) findViewById(R.id.saveButton);
        saveresult.setOnClickListener(this);
        exportresult = (ImageButton) findViewById(R.id.submitButton);
        exportresult.setOnClickListener(this);
        takeAnotherPic = (ImageButton) findViewById(R.id.button4);
        Bundle extras = getIntent().getExtras();
        fixRed = extras.getInt("fixRed");
        fixBlue = extras.getInt("fixBlue");
        fixGreen = extras.getInt("fixGreen");
        calibrateRed=extras.getInt("calibrateRed");
        calibrateGreen=extras.getInt("calibrateGreen");
        calibrateBlue=extras.getInt("calibrateBlue");
        calibrateHue=extras.getString("calibrateHue");
        calibrateValue=extras.getString("calibrateValue");
        calibrateChroma=extras.getString("calibrateChroma");
        expectedHue = (Spinner)findViewById(spinner);
        expectedValue = (Spinner)findViewById(spinner2);
        expectedChroma = (Spinner)findViewById(spinner3);



/* Extracts image taken from camera or image selected from gallery and
passes it to imageview -JB
 */
        /* Takes bitmp image from Camera Intent, finds Munsell, and sets bitmap to
imageview -JB*/
        if (getIntent().hasExtra("CameraImage")) {
            b = BitmapFactory.decodeByteArray(
                    getIntent().getByteArrayExtra("CameraImage"), 0,
getIntent().getByteArrayExtra("CameraImage").length);
            try {
                munsell(findViewById(R.id.musellValue),b);
            } catch (IOException e) {
                e.printStackTrace();
            }

            ResultPic.setImageBitmap(b);
        }
        /* Takes bitmp image from gallery, finds Munsell, and resizes image to fit in
```

```
imageview -JB*/
        else if (getIntent().hasExtra("GalleryImage")) {
            b = BitmapFactory.decodeByteArray(
                    getIntent().getByteArrayExtra("GalleryImage"), 0,
getIntent().getByteArrayExtra("GalleryImage").length);

            try {
                munsell(findViewById(R.id.musellValue),b);
            } catch (IOException e) {
                e.printStackTrace();
            }

            ResultPic.setImageBitmap(b);
        }



    }


    /*Starts camera Intent -JB*/
    public void AnotherCameraClick(View v) {
        try {
            Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
            startActivityForResult(intent, TAKE_ANOTHERPIC);
            // we will handle the returned data in onActivityResult
        } catch (ActivityNotFoundException anfe) {
            Toast toast = Toast.makeText(this, "This device doesn't support the crop
action!",
                    Toast.LENGTH_SHORT);
            toast.show();
        }


    }


    /* Distance formula for two 3D point */
    public static double getDistance(float aR, float aG, float aB, float cR, float cG,
float cB) {
        float dx = aR - cR;
        float dy = aG - cG;
        float dz = aB - cB;

        return Math.sqrt(dx * dx + dy * dy + dz * dz);
    }


    /*Goes through the munsell.csv file a first time and calculates the distance
between the actual average RGB
         * and the RGB values in the csv file. If the distance calculated is smaller
than the previous smallest distance,
          * the smallest distance value gets updated along with the smallest red,
green, and blue value.
          * It then goes through the csv file again with the new RGB values and finds
the line with the matching values and
          * returns the munsell color to the phone. It then changes the background to
show the munsell chip color.*/
    public void munsell(View v, Bitmap bit) throws IOException {
        smallestDif=1000.0;


        TextView text = (TextView) findViewById(R.id.musellValue);
```

```
        InputStream csv;

        csv = getAssets().open("munsell.csv");

        InputStreamReader is = new InputStreamReader(csv);

        CSVReader csvReader = new CSVReader(is);
        String[] line;
        csvReader.readNext();
        getSpecs(bit);


        //This should  fix the colors from the calibration activity.
        //UNCOMMENT ONCE WE CAN GET THE SPECS OF A KNOWS COLOR.
        System.out.println("redfactor: "+ fixRed+" greenfactor: "+ fixGreen+"
bluefactor: "+ fixBlue);
        red=red+fixRed;
        green=green+fixGreen;
        blue=blue+fixBlue;
        System.out.println("fixed red: "+ red + " fixed green"+ green +" fixed blue"+
blue);


        while ((line = csvReader.readNext()) != null) {
            compareRed = Integer.parseInt(line[line.length - 3]);
            compareGreen = Integer.parseInt(line[line.length - 2]);
            compareBlue = Integer.parseInt(line[line.length - 1]);
            if (getDistance(red, green, blue, compareRed, compareGreen, compareBlue) <
smallestDif) {
                smallestDif = getDistance(red, green, blue, compareRed, compareGreen,
compareBlue);
                smallRed = Integer.parseInt(line[3]);
                smallGreen = Integer.parseInt(line[4]);
                smallBlue = Integer.parseInt(line[5]);
            } //else
                //csvReader.readNext();
        }
        csvReader.close();
        System.out.println("smalled difference: " + Double.toString(smallestDif) + "/n
smallest red: " + Double.toString(smallRed) + "/n Actual red:"
                + Integer.toString(red) + "/n Smallest green: " +
Double.toString(smallGreen) + "/n Actual Green:" + Integer.toString(green) +
                "/n Actual blue:" + Integer.toString(blue) + "/n Smallest Blue: " +
Double.toString(smallBlue));

        InputStream csv2;

        csv2 = getAssets().open("munsell.csv");

        InputStreamReader is2 = new InputStreamReader(csv2);

        CSVReader csvReader2 = new CSVReader(is2);
        String[] line2;
        csvReader2.readNext();


        while ((line2 = csvReader2.readNext()) != null) {
            if (smallRed == (Integer.parseInt(line2[line2.length - 3]))) {
                if (smallGreen == (Integer.parseInt(line2[line2.length - 2]))) {
                    if (smallBlue == Integer.parseInt(line2[line2.length - 1])) {
                        munsellValue = line2[0] + " " + line2[1] + "/" + line2[2];
                        foundMunsellHue=line2[0];
```

```
                        foundMunsellValue=line2[1];
                        foundMunsellChroma=line2[2];
                        text.setText(munsellValue);

                    }
                }
            }
        }
        csvReader2.close();
        //setBackground(red,green,blue);
        Toast.makeText(getApplicationContext(), " R "+red+" G "+green+" B "+blue,
                Toast.LENGTH_LONG).show();
       // System.out.println("Distance is:
"+getDistance(calibrateRed,calibrateGreen,calibrateBlue,red,green,blue));
    //input expected RGBs ^^
        InputStream csv5;

        csv5 = getAssets().open("munsell.csv");

        InputStreamReader is5 = new InputStreamReader(csv5);

        CSVReader csvReader5 = new CSVReader(is5);
        csvReader5.readNext();


        while ((line5 = csvReader5.readNext()) != null) {
            if (foundMunsellHue.equals((line5[0]))) {
                if (foundMunsellValue.equals((line5[1]))) {
                    if (foundMunsellChroma.equals((line5[2]))) {
                        finalRed = Integer.parseInt(line5[3]);
                        finalGreen = Integer.parseInt(line5[4]);
                        finalBlue = Integer.parseInt(line5[5]);

                    }
                }
            }
        }
        csvReader5.close();
    }


    @Override
    public void onClick(View v) {
        iaDataStorage = (TextView) findViewById(R.id.dataStorage);
        switch (v.getId()) {
            case R.id.homeButton:
                Intent i = new Intent(this, MainActivity.class);
                Bundle bundle = new Bundle();
                bundle.putString("dataList", iaDataStorage.getText().toString());
                i.putExtras(bundle);
                startActivity(i);
                break;
            case R.id.submitButton:
                //uncomment for testing
               /* try {
                    getRGB();
                } catch (IOException e) {
                    e.printStackTrace();
                }*/
                //uncomment for testing

/*rgbDistancedouble=getDistance(finalRed,finalGreen,finalBlue,expectedRed,expectedGree
n,expectedBlue);
```

```
                rgbDistance=Double.toString(rgbDistancedouble);

calibratergbDistanceDouble=getDistance(finalRed,finalGreen,finalBlue,calibrateRed,cali
brateGreen,calibrateBlue);
                calibratergbDistance=Double.toString(calibratergbDistanceDouble);
                String expected= expectedHue.getSelectedItem().toString()+"
"+expectedValue.getSelectedItem().toString()+"/"+expectedChroma.getSelectedItem().toSt
ring();*/
                munsellChip = (TextView) findViewById(R.id.musellValue);
                Intent submitForm = new Intent(this, SubmitForm.class);
                Bundle submitBundle = new Bundle();
                submitBundle.putString("MunsellChip",
munsellChip.getText().toString());
                submitBundle.putString("dataList",
iaDataStorage.getText().toString());
                //Can be uncommented when testing
               /* submitBundle.putString("expectedMunsellChip", expected);
                submitBundle.putString("expectedHue",
expectedHue.getSelectedItem().toString());
                submitBundle.putString("expectedValue",
expectedValue.getSelectedItem().toString());
                submitBundle.putString("expectedChroma",
expectedChroma.getSelectedItem().toString());
                submitBundle.putString("foundMunsellHue", foundMunsellHue);
                submitBundle.putString("foundMunsellValue", foundMunsellValue);
                submitBundle.putString("foundMunsellChroma", foundMunsellChroma);*/
                submitBundle.putString("calibrateHue", calibrateHue);
                submitBundle.putString("calibrateValue", calibrateValue);
                submitBundle.putString("calibrateChroma", calibrateChroma);
                //Can be uncommented when testing
            /*    submitBundle.putString("rgbDistance",rgbDistance);
                submitBundle.putString("calibratergbDistance",calibratergbDistance);*/
                submitForm.putExtras(submitBundle);
                startActivity(submitForm);
                break;
            case R.id.saveButton:
                /*Takes Screenshot of Activity and Saves reading to Gallery */
                View v1 =R1.getRootView();
                v1.setDrawingCacheEnabled(true);
                Bitmap savebm = v1.getDrawingCache();
                BitmapDrawable bitmapDrawable = new BitmapDrawable(savebm);
                String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss").format(new
Date());
                Bitmap combination =savebm;
                MediaStore.Images.Media.insertImage

(getApplicationContext().getContentResolver(),combination,"test_"+ timeStamp +
".jpg",timeStamp.toString());
                Toast.makeText(getApplicationContext(), "Your Image Has Been Saved
Successfully",
                        Toast.LENGTH_LONG).show();
                break;


        }

    }
//USED ONLY FOR TESTING maps expected munsell to rgb
    public void getRGB() throws IOException {
       // System.out.println("expectedssssss"+expectedRed+" "+expectedGreen+" "+
expectedBlue);

        InputStream csv3;
```

```
        csv3 = getAssets().open("munsell.csv");

        InputStreamReader is3 = new InputStreamReader(csv3);

        CSVReader csvReader3 = new CSVReader(is3);
        csvReader3.readNext();
        expectedHueValue =expectedHue.getSelectedItem().toString();
        expectedValueValue=expectedValue.getSelectedItem().toString();
        expectedChromaValue=expectedChroma.getSelectedItem().toString();
        while ((line3 = csvReader3.readNext()) != null) {
            if (expectedHueValue.equals((line3[line3.length - 6]))) {
                if (expectedValueValue.equals((line3[line3.length - 5]))) {
                    if (expectedChromaValue.equals((line3[line3.length - 4]))) {
                        expectedRed=Integer.parseInt(line3[3]);
                        expectedGreen=Integer.parseInt(line3[4]);
                        expectedBlue=Integer.parseInt(line3[5]);

                    }
                }
            }
        }
        csvReader3.close();
        System.out.println("expectedssssss+"+expectedRed+" "+expectedGreen+" "+
expectedBlue);
    }


    /*Changes the RGB values to hex numbers and then creates a HexString to change the
background of the phone.
       * If the R,G, or B value is a single digit, it adds a zero infront. */
    public void setBackground(int red, int green, int blue) {

        View view = this.getWindow().getDecorView();


         if (red > 9 && green > 9 && blue > 9) {
            StringBuilder builder = new StringBuilder();
            builder.append("#");
            builder.append(Integer.toHexString(red));
            builder.append(Integer.toHexString(green));
            builder.append(Integer.toHexString(blue));
            view.setBackgroundColor(Color.parseColor(builder.toString()));
            return;
        }
        else if (green < 10 && blue < 10 && red < 10) {
            StringBuilder builder = new StringBuilder();
            builder.append("#");
            builder.append("0" + Integer.toString(red));
            builder.append("0" + Integer.toString(green));
            builder.append("0" + Integer.toString(blue));

            view.setBackgroundColor(Color.parseColor(builder.toString()));
            return;

        }
        else if (red < 10 && green < 10) {
            StringBuilder builder = new StringBuilder();
            builder.append("#");
            builder.append("0" + Integer.toString(red));
            builder.append("0" + Integer.toString(green));
            builder.append(Integer.toHexString(blue));
```

```
            view.setBackgroundColor(Color.parseColor(builder.toString()));
            return;
        } else if (red < 10 && blue < 10) {
            StringBuilder builder = new StringBuilder();
            builder.append("#");
            builder.append("0" + Integer.toString(red));
            builder.append(Integer.toHexString(green));
            builder.append("0" + Integer.toString(blue));

            view.setBackgroundColor(Color.parseColor(builder.toString()));
            return;
        } else if (green < 10 && blue < 10) {
            StringBuilder builder = new StringBuilder();
            builder.append("#");
            builder.append(Integer.toHexString(red));
            builder.append("0" + Integer.toString(green));
            builder.append("0" + Integer.toString(blue));

            view.setBackgroundColor(Color.parseColor(builder.toString()));
            return;
        }
        else if (red < 10) {
            StringBuilder builder = new StringBuilder();
            builder.append("#");
            builder.append("0" + Integer.toString(red));
            builder.append(Integer.toHexString(green));
            builder.append(Integer.toHexString(blue));

            view.setBackgroundColor(Color.parseColor(builder.toString()));
            return;
        } else if (green < 10) {
            StringBuilder builder = new StringBuilder();
            builder.append("#");
            builder.append(Integer.toHexString(red));
            builder.append("0" + Integer.toString(green));
            builder.append(Integer.toHexString(blue));

            view.setBackgroundColor(Color.parseColor(builder.toString()));
            return;
        } else if (blue < 10) {
            StringBuilder builder = new StringBuilder();
            builder.append("#");
            builder.append(Integer.toHexString(red));
            builder.append(Integer.toHexString(green));
            builder.append("0" + Integer.toString(blue));
            view.setBackgroundColor(Color.parseColor(builder.toString()));
            return;
        }

    }


    public void getSpecs(Bitmap bit) {
        //When implementing with camera, change field i to get the image taken from
the camera,
        //so it's no longer pre loaded in with Android Studio
        ImageView i = new ImageView(this);
        i.setImageBitmap(bit);
        //System.out.println(w +" "+ h);
        Bitmap bitmap = ((BitmapDrawable) i.getDrawable()).getBitmap();
        int w = bitmap.getWidth();
        int h = bitmap.getHeight();
        int count=0;
```

```
        //System.out.cprintln(bitmap.getWidth() + " " + bitmap.getHeight());
        for(int x = 0; x< w-1; x++){
            for(int y = 0; y< h-1; y++){
                int pixel = bitmap.getPixel(x, y);
                red += Color.red(pixel);
                blue += Color.blue(pixel);
                green += Color.green(pixel);
                count++;
                //

            }
        }

        red = red/count;
        blue = blue/count;
        green = green/count;
        System.out.println("image red: "+ red + " image green: "+ green +" image blue:
"+ blue);


    }


    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        if (requestCode == TAKE_ANOTHERPIC && resultCode == RESULT_OK) {
            photo = data.getData();
            performCrop();

        } else if (requestCode == CROP_PIC) {
            // get the returned data
            Bundle extras = data.getExtras();
            // get the cropped bitmap
            Bitmap croppedPic = extras.getParcelable("data");
            try {
                munsell(findViewById(R.id.musellValue), croppedPic);
            } catch (IOException e) {
                e.printStackTrace();
            }

            ResultPic.setImageBitmap(croppedPic);

        }
    }


    private void performCrop() {
        // take care of exceptions
        try {
            // call the standard crop action intent (the user device may not
            // support it)
            Intent cropIntent = new Intent("com.android.camera.action.CROP");
            // indicate image type and Uri
            cropIntent.setDataAndType(photo, "image/*");
            // set crop properties
            cropIntent.putExtra("crop", "true");
            // indicate aspect of desired crop
            cropIntent.putExtra("aspectX", 2);
            cropIntent.putExtra("aspectY", 1);
            // indicate output X and Y
            cropIntent.putExtra("outputX", 256);
            cropIntent.putExtra("outputY", 256);
            // retrieve data on return
```

```
            cropIntent.putExtra("return-data", true);
            // start the activity - we handle returning in onActivityResult
            startActivityForResult(cropIntent, CROP_PIC);
        }
        // respond to users whose devices do not support the crop action
        catch (ActivityNotFoundException anfe) {
            Toast toast = Toast
                    .makeText(this, "This device doesn't support the crop action!",
Toast.LENGTH_SHORT);
            toast.show();
        }
    }

}
```

Appendix E : Submission form activity source code

```
package com.munsellapp.munsellcolorrecognitionapp;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.location.LocationServices;
import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.List;
import java.util.Locale;
import java.lang.Math;
/* This activity allows the user to enter details about the image they have taken.
Munsell Value of the image will be
passed from ImageActivity and location will appear automatically. User has the option
to save the information to the data.txt text
file, which will promp the DataForm class.*/

public class SubmitForm extends AppCompatActivity implements
View.OnClickListener,GoogleApiClient.ConnectionCallbacks,
GoogleApiClient.OnConnectionFailedListener {
    ImageButton save, email;
    EditText idNumber, notes;
    TextView munsell, munsellValueText, updatedText, expectedMunsellValueText,
distanceValueText, rgbDistanceValueText;
    String munsellChip, expectedChip, expectedHue, expectedValue, expectedChroma;
    String foundMunsellHue, foundMunsellValue, foundMunsellChroma;
    double distance, calibrateMunsellDistance;
    TextView location;
    String rgbDistance, calibratergbDistance;
    private GoogleApiClient googleApiClient;
    private static final int PERMISSION_ACCESS_COARSE_LOCATION = 0;
    String calibrateHue, calibrateValue, calibrateChroma, calibrationChip;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_submit_form);
        save = (ImageButton) findViewById(R.id.sfSaveButton);
        save.setOnClickListener(this);
        Bundle getBundle= getIntent().getExtras();
        munsellChip=getBundle.getString("MunsellChip");
        //uncomment when testing
   /*     expectedChip=getBundle.getString("expectedMunsellChip");
```

```
        expectedHue=getBundle.getString("expectedHue");
        expectedValue=getBundle.getString("expectedValue");
        expectedChroma=getBundle.getString("expectedChroma");
        foundMunsellHue=getBundle.getString("foundMunsellHue");
        foundMunsellValue=getBundle.getString("foundMunsellValue");
        foundMunsellChroma=getBundle.getString("foundMunsellChroma");*/
        calibrateHue=getBundle.getString("calibrateHue");
        calibrateValue=getBundle.getString("calibrateValue");
        calibrateChroma=getBundle.getString("calibrateChroma");
        //rgbDistance=getBundle.getString("rgbDistance");
       // calibratergbDistance=getBundle.getString("calibratergbDistance");

//distance=getMunsellDistance(expectedHue,expectedValue,expectedChroma,foundMunsellHue
,foundMunsellValue,foundMunsellChroma);
       //
calibrateMunsellDistance=getMunsellDistance(expectedHue,expectedValue,expectedChroma,c
alibrateHue,calibrateValue,calibrateChroma);
        calibrationChip=calibrateHue+" "+calibrateValue+"/"+calibrateChroma;
        //rgbDistanceValueText=(TextView) findViewById(R.id.textView19);
        System.out.println(rgbDistance);
        //Can be uncommented when testing
        // rgbDistanceValueText.setText(rgbDistance);
        munsellValueText = (TextView) findViewById(R.id.sfMunsellChip);
        munsellValueText.setText(munsellChip);
        //expectedMunsellValueText= (TextView) findViewById(R.id.textView16);
        //Can be uncommented when testing
        //expectedMunsellValueText.setText(expectedChip);
        //distanceValueText=(TextView) findViewById(R.id.textView18);
        //Can be uncommented when testing
        //distanceValueText.setText(Double.toString(distance));
        location = (TextView) findViewById(R.id.textView6);
        googleApiClient = new GoogleApiClient.Builder(this, this,
this).addApi(LocationServices.API).build();
        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION)
                != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.ACCESS_COARSE_LOCATION},
                    PERMISSION_ACCESS_COARSE_LOCATION);
        }
    }

    private double getMunsellDistance(String expectedHue, String expectedValue, String
expectedChroma, String foundMunsellHue, String foundMunsellValue, String
foundMunsellChroma) {
        double distance;
        double expectedHueAngle=findAngle(expectedHue);
        System.out.println(expectedHueAngle);
        double foundMunsellHueAngle=findAngle(foundMunsellHue);
        System.out.println(foundMunsellHueAngle);
        double x1 = Math.sin(expectedHueAngle)*Double.parseDouble(expectedChroma);
        double y1= Math.cos(expectedHueAngle)*Double.parseDouble(expectedChroma);
        double z1=Double.parseDouble(expectedValue);
        double x2 =
Math.sin(foundMunsellHueAngle)*Double.parseDouble(foundMunsellChroma);
        double y2=
Math.cos(foundMunsellHueAngle)*Double.parseDouble(foundMunsellChroma);
        double z2=Double.parseDouble(foundMunsellValue);
        distance=Math.sqrt((x2-x1)*(x2-x1) + (y2-y1)*(y2-y1) + (z2-z1)*(z2-z1));
        System.out.println(x1+" "+y1+" "+z1+" "+x2+" "+y2+" "+z2);
        return distance;
    }
```

```
private double findAngle(String hue) {
    double angle= 0.0;
    switch (hue) {
        case "N":
            angle = 0.0;
            break;
        case "5.0R":
            angle = 0.0;
            break;
        case "7.5R":
            angle = 9.0;
            break;
        case "10.0R":
            angle = 18.0;
            break;
        case "2.5YR":
            angle = 27.0;
            break;
        case "5.0YR":
            angle = 36.0;
            break;
        case "7.5YR":
            angle = 45.0;
            break;
        case "10.0YR":
            angle = 54.0;
            break;
        case "2.5Y":
            angle = 63.0;
            break;
        case "5.0Y":
            angle = 72.0;
            break;
        case "7.5Y":
            angle = 81.0;
            break;
        case "10.0Y":
            angle = 90.0;
            break;
        case "2.5GY":
            angle = 99.0;
            break;
        case "5.0GY":
            angle = 108.0;
            break;
        case "7.5GY":
            angle = 117.0;
            break;
        case "10.0GY":
            angle = 126.0;
            break;
        case "2.5G":
            angle = 135.0;
            break;
        case "5.0G":
            angle = 144.0;
            break;
        case "7.5G":
            angle = 153.0;
            break;
        case "10.0G":
            angle = 162.0;
            break;
```

```
case "2.5BG":
    angle = 171.0;
    break;
case "5.0BG":
    angle = 180.0;
    break;
case "7.5BG":
    angle = 189.0;
    break;
case "10.0BG":
    angle = 198.0;
    break;
case "2.5B":
    angle = 207.0;
    break;
case "5.0B":
    angle = 216.0;
    break;
case "7.5B":
    angle = 225.0;
    break;
case "10.0B":
    angle = 234.0;
    break;
case "2.5PB":
    angle = 243.0;
    break;
case "5.0PB":
    angle = 252.0;
    break;
case "7.5PB":
    angle = 261.0;
    break;
case "10.0PB":
    angle = 270.0;
    break;
case "2.5P":
    angle = 279.0;
    break;
case "5.0P":
    angle = 288.0;
    break;
case "7.5P":
    angle = 297.0;
    break;
case "10.0P":
    angle = 306.0;
    break;
case "2.5RP":
    angle = 315.0;
    break;
case "5.0RP":
    angle = 324.0;
    break;
case "7.5RP":
    angle = 333.0;
    break;
case "10.0RP":
    angle = 342.0;
    break;
case "2.5R":
    angle= 351.0;
    break;
```

```
            default:
                break;
        }
        return angle * 0.0174533;
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, String[] permissions,
int[] grantResults) {
        switch (requestCode) {
            case PERMISSION_ACCESS_COARSE_LOCATION:
                if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                    Toast.makeText(this, "All good?!", Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(this, "Need your location!",
Toast.LENGTH_SHORT).show();
                }
                break;
        }
    }

    private void saveInInternalFolder(String aStringToSave, String aFileName){
        FileOutputStream fos=null;
        aStringToSave=idNumber+" , "+munsellChip+" , "+ notes;
        try{
            fos=openFileOutput(aFileName, this.MODE_PRIVATE);
            fos.write(aStringToSave.getBytes());
            fos.close();
            Toast.makeText(this, "file saved", Toast.LENGTH_LONG).show();

            FileInputStream fis = this.openFileInput(aFileName);
            InputStreamReader isr = new InputStreamReader(fis);
            BufferedReader bufferedReader = new BufferedReader(isr);
            StringBuilder sb = new StringBuilder();
            String line;
            while ((line = bufferedReader.readLine()) != null) {
                sb.append(line);
            }

        }catch (IOException e){
            Toast.makeText(this, "There is a problem saving to the internal file",
Toast.LENGTH_LONG).show();

        }
    }
    @Override
    public void onClick(View v) {
        idNumber=(EditText) findViewById(R.id.sfIdEdit);
        notes=(EditText) findViewById(R.id.sfNotesEdit);
        updatedText=(TextView) findViewById(R.id.sfInfoStorage);
        Intent intent=new Intent(this, DataForm.class);
        Bundle bundle=new Bundle();
        bundle.putString("idNumber", idNumber.getText().toString());
        bundle.putString("munsellChip", munsellValueText.getText().toString());
        bundle.putString("notes", notes.getText().toString());
        bundle.putString("location",location.getText().toString());
        bundle.putString("expectedChip",expectedChip);
        bundle.putString("expectedHue",expectedHue);
        bundle.putString("expectedValue",expectedValue);
        bundle.putString("expectedChroma",expectedChroma);
        bundle.putString("distance",Double.toString(distance));
        bundle.putString("rgbDistance", rgbDistance);
```

```
        bundle.putString("calibratergbDistance",calibratergbDistance);

bundle.putString("calibrateMunsellDistance",Double.toString(calibrateMunsellDistance))
;
        bundle.putString("calibrationChip", calibrationChip);


//        if(updatedText.equals("")){
        intent.putExtras(bundle);
        startActivity(intent);
//        }else{
//            bundle.putString("dataList", updatedText.getText().toString());
//            intent.putExtras(bundle);
        startActivity(intent);
    }


    protected String cityName;

    @Override
    protected void onStart() {
        super.onStart();
        googleApiClient.connect();
    }
    @Override
    public void onConnected(Bundle bundle) {
        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION)
                == PackageManager.PERMISSION_GRANTED) {
            Location lastLocation =
LocationServices.FusedLocationApi.getLastLocation(googleApiClient);
            double lat = lastLocation.getLatitude(), lon =
lastLocation.getLongitude();
            String units = "imperial";
            System.out.println(lat);
            Geocoder gcd = new Geocoder(this, Locale.getDefault());
            List<Address> addresses = null;
            try {
                addresses = gcd.getFromLocation(lat, lon, 1);
            } catch (IOException e) {
                e.printStackTrace();
            }
            if (addresses.size() > 0) {
                System.out.println(addresses.get(0).getLocality());
                cityName = addresses.get(0).getLocality();
            }}
        location.setText(cityName);
    }
    @Override
    public void onConnectionSuspended(int cause) {
        googleApiClient.connect();
    }
    protected String getCityName(){
        return cityName;
    }

    @Override
    public void onConnectionFailed(ConnectionResult connectionResult) {

    }
}
```

Appendix F : Data form activity source code

```
package com.munsellapp.munsellcolorrecognitionapp;

import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.net.Uri;
import android.os.Environment;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.ImageButton;
import android.widget.TextView;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
//This activity appends to a data.txt file each time a new submission is made from
submitform activity,
//from here the user has the option to email the data.txt file.
public class DataForm extends AppCompatActivity implements View.OnClickListener {
    TextView dataListText, savedData;
    String idNumber, munsellChip, notes, dataString, dataListString,
dataListStringBundle;
    ImageButton home, email;
    SharedPreferences savedValues;
    String savedDataString, restoreData;
    String savedDataPref;
    String fileLocation, location, expectedChip;
    String distance ,rgbDistance, calibratergbDistance,calibrateMunsellDistance,
calibrationChip;
    ImageButton takeAnother;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_data_form);
        dataListText = (TextView) findViewById(R.id.dataList);
        home = (ImageButton) findViewById(R.id.dfHome);
        home.setOnClickListener(this);
        email = (ImageButton) findViewById(R.id.dfEmail);
        email.setOnClickListener(this);

//
        takeAnother=(ImageButton) findViewById(R.id.imageButton);
        takeAnother.setOnClickListener(this);
        Bundle getBundle = getIntent().getExtras();
        idNumber = getBundle.getString("idNumber");
        munsellChip = getBundle.getString("munsellChip");
        notes = getBundle.getString("notes");
        location= getBundle.getString("location");
        expectedChip= getBundle.getString("expectedChip");
        calibrationChip=getBundle.getString("calibrationChip");
        distance= getBundle.getString("distance");
        rgbDistance=getBundle.getString("rgbDistance");
```

```
        calibratergbDistance=getBundle.getString("calibratergbDistance");
        calibrateMunsellDistance=getBundle.getString("calibrateMunsellDistance");
        SharedPreferences sp = getSharedPreferences("key", 0);
        savedDataPref = sp.getString("savedDataPref", "");
        //uncomment for final this is just for testing
        dataListText.setText("\""+idNumber+ "\"" + " , " +"\"" + calibrationChip +
"\"" + " , " + "\"" + munsellChip + "\"" + " , " + "\""+location+ "\""+ " , "+"\"" +
notes + "\"" + "\n" + savedDataPref);
        //dataListText.setText( calibrationChip+ " , "+expectedChip  + " , "
+munsellChip+ " , "+distance+ " , "+rgbDistance+" , "+calibrateMunsellDistance+ " ,
"+calibratergbDistance+"\n"+ savedDataPref);

    }

//    @Override
//    public void onSaveInstanceState(Bundle savedInstanceState) {
//        super.onSaveInstanceState(savedInstanceState);
//        // Save UI state changes to the savedInstanceState.
//        // This bundle will be passed to onCreate if the process is
//        // killed and restarted.
//
//        savedInstanceState.putString("dataList", dataListText.getText().toString());
//        // etc.
//    }
//
//
//    @Override
//    public void onRestoreInstanceState(Bundle savedInstanceState) {
//        super.onRestoreInstanceState(savedInstanceState);
//        // Restore UI state from the savedInstanceState.
//        // This bundle has also been passed to onCreate.
//        restoreData=savedInstanceState.getString("dataList");
//    }


    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.dfHome:
                dataListText = (TextView) findViewById(R.id.dataList);
                Intent intent = new Intent(this, MainActivity.class);

                SharedPreferences sp = getSharedPreferences("key", 0);
                SharedPreferences.Editor editor = sp.edit();
                editor.putString("savedDataPref", dataListText.getText().toString());
                editor.commit();
                startActivity(intent);

                break;
            case R.id.imageButton:
                dataListText = (TextView) findViewById(R.id.dataList);
                Intent Newintent = new Intent(this, ImageActivity.class);

                SharedPreferences Newsp = getSharedPreferences("key", 0);
                SharedPreferences.Editor NewEditor = Newsp.edit();
                NewEditor.putString("savedDataPref",
dataListText.getText().toString());
                NewEditor.commit();
                startActivity(Newintent);

                break;
            case R.id.dfEmail:
                new AlertDialog.Builder(DataForm.this)
```

```
                        .setTitle("Warning!")
                        .setMessage("Data list will be deleted upon emailing results.
Are you sure " +
                                "you want to proceed?")
                        .setPositiveButton("YES", new
DialogInterface.OnClickListener() {
                            public void onClick(DialogInterface dialog, int which) {
                                writeFile();
                                Intent emailIntent = new Intent((Intent.ACTION_SEND));
                                emailIntent.setType("text/plain");
                                emailIntent.putExtra(Intent.EXTRA_EMAIL, new
String[]{""});
                                emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Munsell
Data List");
                                emailIntent.putExtra(Intent.EXTRA_TEXT, "");

                                emailIntent.putExtra(Intent.EXTRA_STREAM,
Uri.parse("file://" + fileLocation));
                                startActivity(emailIntent);
                            }
                        })
                        .setNegativeButton("NO", new DialogInterface.OnClickListener()
{
                            public void onClick(DialogInterface dialog, int which) {
                                Log.d("AlertDialog", "Negative");
                            }
                        })
                        .show();
                break;

        }
    }

    public void writeFile() {
        String content = dataListText.getText().toString();
        //This string is used when testing
       // String heading = "\"" + "Calibration Chip" + "\"" + "," + "\""+"\"" +
"Expected Munsell" + "\"" + "," + "\"" + "Found Munsell" + "\"" + ',' + "\"" +
"Distance in Munsell between found and expected" + "\""+ "," + "\"" + "Distance in RGB
between found and expected" + "\""+ "," + "\"" +"Distance in Munsell between
calibration chip and expected"+"\""+ "," + "\"" +"Distance in rgb between calibration
chip and expected"+"\"";
        //This string is what the user should use for transferring data
        String heading = "\"" + "ID"  + "\"" + "," + "\"" +"Calibration Chip" + "\"" +
"," + "\"" + "Munsell Chip" + "\"" + "," + "\"" + "Location" + "\"" + ',' + "\"" +
"Notes" +"\"";
        String fullContent = heading + "\n" + content;
        File file;
        FileOutputStream outputStream;
        try {
            file = new File(Environment.getExternalStorageDirectory(),
"DataList.txt");
            outputStream = new FileOutputStream(file);
            outputStream.write(fullContent.getBytes());
            outputStream.close();
            System.out.println(file);
            fileLocation = file.toString();
        } catch (IOException e) {
            e.printStackTrace();

        }
        dataListText.setText("");
    }
```

```
    public void readFile() {
        File Root = Environment.getExternalStorageDirectory();
        File dir = new File(Root.getAbsolutePath() + "/MyAppFile");
        File dataText = new File(dir, "data.txt");

        String message;
        try {
            FileInputStream fis = new FileInputStream(dataText);
            InputStreamReader isr = new InputStreamReader(fis);
            BufferedReader buffRead = new BufferedReader(isr);
            StringBuffer stringBuff = new StringBuffer();

            while ((message = buffRead.readLine()) != null) {
                stringBuff.append(message + "/n");
            }

            System.out.println(stringBuff.toString());
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }


    }

}
```

Appendix G : Main activity layout

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/leavess"
    android:layout_gravity="center"
    android:orientation="vertical"
    android:paddingBottom="32dp"
    tools:context="com.munsellapp.munsellcolorrecognitionapp.MainActivity">


        <TableRow
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="2"
            android:gravity="center_horizontal">

            <TextView
                android:id="@+id/textView2"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="22dp"
                android:fontFamily="serif"
                android:gravity="center_horizontal"
                android:layout_gravity="center_vertical"
                android:text="Munsell"
                android:textAlignment="center"
                android:textColor="@android:color/black"
                android:textSize="40sp"
                android:textStyle="normal|bold"
                />
        </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="2">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/textView10" />
    </TableRow>

    <TableRow
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="2"
            android:gravity="center_horizontal">

            <Button
                android:id="@+id/button1"
                android:layout_width="200dp"
                android:layout_height="match_parent"
                android:onClick="CameraClick"
                android:text="Take Picture"
```

```
                android:textSize="16sp"
                android:fontFamily="sans-serif-condensed" />
        </TableRow>
    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="2">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/textView12" />


        </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="2"
        android:gravity="center_horizontal"
        android:orientation="vertical">


        <Button
            android:id="@+id/ChooseImage"
            android:layout_width="200dp"
            android:layout_height="match_parent"
            android:textSize="16sp"
            android:text="Select From Gallery"
            android:fontFamily="sans-serif-condensed" />
    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="2">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/textView11" />
    </TableRow>

    <TableRow
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="2"
            android:gravity="center_horizontal">

            <Button
            android:id="@+id/button3"
            android:layout_width="200dp"
            android:layout_height="match_parent"
            android:onClick="CalibrateCameraClick"
            android:text="Calibrate Camera"
                android:textSize="16sp"
                android:fontFamily="sans-serif-condensed" />
        </TableRow>


</TableLayout
```

Appendix H : Calibration home layout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Spinner
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:id="@+id/spinner5"
        android:entries="@array/MunsellHueArray"
        android:prompt="@string/MunsellHuePrompt"/>

    <Spinner
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/spinner"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:id="@+id/spinner6"
        android:entries="@array/MunsellValueArray"
        android:prompt="@string/MunsellValuePrompt"/>

    <Spinner
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/spinner7"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:id="@+id/spinner4"
        android:entries="@array/MunsellChromaArray"
        android:prompt="@string/MunsellChromaPrompt"/>

    <TextView
        android:text="Choose your Calibration Constant and click next"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/textView20" />

    <Button
        android:text="Submit"
        android:layout_width="284dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:id="@+id/button5" />
</LinearLayout>
```

Appendix I : Calibration activity layout

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_calibrate"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.munsellapp.munsellcolorrecognitionapp.Calibrate">

    <ImageView
        android:id="@+id/caliPic"

        android:layout_width="match_parent"
        android:layout_height="300dp"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginBottom="47dp">

    </ImageView>

    <TextView
        android:text="Click to Calibrate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="25sp"
        android:textAlignment="center"
        android:gravity="center_horizontal"
        android:id="@+id/textView3"
        android:fontFamily="sans-serif-condensed"
        android:textColor="@color/common_google_signin_btn_text_dark_focused"
        android:layout_above="@+id/calibrateImageButton"
        android:layout_centerHorizontal="true" />

    <Button
        android:text="Take Picture"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button"
        android:visibility="gone"
        android:onClick="CamClick"
        android:layout_alignParentBottom="true"
        android:layout_toLeftOf="@+id/calibrateImageButton"
        android:layout_toStartOf="@+id/calibrateImageButton" />

    <Button
        android:text="Gallery"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="galleryIntent"
        android:id="@+id/button2"
        android:visibility="gone"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_marginRight="26dp"
```

```
        android:layout_marginEnd="26dp" />

    <ImageButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/calibrate"
        android:id="@+id/calibrateImageButton"
        android:layout_below="@+id/caliPic"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="13dp" />

</RelativeLayout>
```

Appendix J : Image activity layout

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/R1">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/backgroundWarning"
        android:layout_below="@+id/musellValue"
        android:textSize="24dp"
        android:gravity="center_horizontal"/>

    <ImageView
        android:id="@+id/imageView1"

        android:layout_width="match_parent"
        android:layout_height="300dp"
        android:layout_above="@+id/linearLayout"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginBottom="47dp">

    </ImageView>

    <LinearLayout
        android:id="@+id/linearLayout"
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true">

        <ImageButton
            android:text="Button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            app:srcCompat="@drawable/cam"
            android:onClick="AnotherCameraClick"
            android:id="@+id/button4"
            android:layout_weight="0.03" />

        <ImageButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            app:srcCompat="@drawable/homebutt"
            android:id="@+id/homeButton"
            android:layout_below="@+id/imageView1"
            android:layout_toRightOf="@+id/submitButton"
            android:layout_toEndOf="@+id/submitButton"
            android:layout_weight="0.03" />

        <ImageButton
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            app:srcCompat="?attr/actionModeShareDrawable"
            android:id="@+id/submitButton"
```

```
            android:layout_alignParentBottom="true"
            android:layout_toRightOf="@+id/munsellButton"
            android:layout_toEndOf="@+id/munsellButton"
            android:layout_weight="0.03" />

        <ImageButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            app:srcCompat="@drawable/savebutt"
            android:id="@+id/saveButton"
            android:layout_weight="0.03"
            android:layout_alignParentBottom="true"
            android:layout_alignParentLeft="true"
            android:layout_alignParentStart="true" />

        <!--<ImageButton-->
            <!--android:layout_width="wrap_content"-->
            <!--android:layout_height="wrap_content"-->
            <!--app:srcCompat="@drawable/cam"-->
            <!--android:id="@+id/imageButton2"-->
            <!--android:onClick="AnotherCameraClick"-->
            <!--android:layout_weight="1" />-->

    </LinearLayout>

    <TextView
        android:text="TextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/imageView1"
        android:layout_alignLeft="@+id/musellValue"
        android:layout_alignStart="@+id/musellValue"
        android:layout_marginLeft="13dp"
        android:layout_marginStart="13dp"
        android:id="@+id/dataStorage"
        android:visibility="invisible"/>

    <!--The following widget is set to invisible for now, set to visible when testing-
->

    <Spinner
        android:layout_width="match_parent"
        android:visibility="invisible"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:id="@+id/spinner"
        android:entries="@array/MunsellHueArray"
        android:prompt="@string/MunsellHuePrompt"/>
    <!--The following widget is set to invisible for now, set to visible when testing-
->

    <Spinner
        android:layout_width="match_parent"
        android:visibility="invisible"
        android:layout_height="wrap_content"
        android:layout_below="@+id/spinner"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:id="@+id/spinner2"
        android:entries="@array/MunsellValueArray"
        android:prompt="@string/MunsellValuePrompt"/>
```

```
    <!--The following widget is set to invisible for now, set to visible when testing-
->

    <Spinner
        android:layout_width="match_parent"
        android:visibility="invisible"
        android:layout_height="wrap_content"
        android:layout_below="@+id/spinner2"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:id="@+id/spinner3"
        android:entries="@array/MunsellChromaArray"
        android:prompt="@string/MunsellChromaPrompt"/>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/musellValue"
        android:textSize="36sp"
        android:gravity="center_horizontal"
        android:textStyle="bold"
        android:background="#FFFFFF"
        android:layout_centerHorizontal="true"
        android:layout_above="@+id/imageView1" />


</RelativeLayout>
```

Appendix K : Data form layout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_data_form"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:orientation="vertical"
    tools:context="com.munsellapp.munsellcolorrecognitionapp.DataForm">

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TextView
            android:text="ID"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/textView21"
            android:layout_weight="1"
            android:textSize="10sp"
            android:textStyle="bold"/>

        <TextView
            android:text="Calibration Chip"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/textView23"
            android:layout_weight="1"
            android:textSize="10sp"
            android:textStyle="bold"/>

        <TextView
            android:text="Munsell Chip"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/textView13"
            android:layout_weight="1"
            android:textSize="10sp"
            android:textStyle="bold"/>

        <TextView
            android:text="Location"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/textView9"
            android:layout_weight="1"
            android:textSize="10sp"
            android:textStyle="bold"/>

        <!--  <TextView
              android:text="Location,"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
```

```
            android:id="@+id/textView8"
            android:layout_weight="1"
            android:textSize="16sp"
            android:textStyle="bold"/>-->

        <TextView
            android:text="Notes"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/textView7"
            android:layout_weight="1"
            android:textSize="10sp"
            android:textStyle="bold"/>
    </LinearLayout>


    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="376dp"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginBottom="32dp">


        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/dataList" />


    </ScrollView>

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <ImageButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            app:srcCompat="@drawable/cam"
            android:id="@+id/imageButton"
            android:layout_weight="1" />

        <ImageButton
            android:layout_width="122dp"
            android:layout_height="wrap_content"
            app:srcCompat="@drawable/homebutt"
            android:id="@+id/dfHome" />

        <ImageButton
            android:layout_width="131dp"
            android:layout_height="wrap_content"
            app:srcCompat="@drawable/email"
            android:id="@+id/dfEmail" />
    </LinearLayout>

</LinearLayout>
```

Appendix L : Submit form layout

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/activity_submit_form"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.munsellapp.munsellcolorrecognitionapp.SubmitForm">

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/linearLayout">

    </LinearLayout>

    <TextView
        android:text="Munsell Chip:"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/amMunsellText"
        android:layout_below="@+id/sfIdEdit"
        android:textStyle="bold"
        android:textSize="16sp"/>

    <EditText
        android:layout_width="286dp"
        android:layout_height="wrap_content"
        android:inputType="textPersonName"
        android:ems="10"
        android:id="@+id/sfIdEdit"
        android:hint="Enter ID here..."
        android:layout_alignParentTop="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:textSize="16sp"
        android:layout_alignLeft="@+id/sfMunsellChip"
        android:layout_alignStart="@+id/sfMunsellChip" />

    <TextView
        android:text="Testing "
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/amMunsellText"
        android:layout_toRightOf="@+id/amMunsellText"
        android:layout_toEndOf="@+id/amMunsellText"
        android:layout_marginLeft="27dp"
        android:layout_marginStart="27dp"
        android:id="@+id/sfMunsellChip"
        android:textSize="16sp"/>

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```
    android:gravity="center_horizontal"
    android:weightSum="2"
    android:layout_alignParentBottom="true"
    android:layout_toRightOf="@+id/linearLayout"
    android:layout_toEndOf="@+id/linearLayout">

    <ImageButton
        android:text="Save"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_margin="16dp"
        android:id="@+id/sfSaveButton"
        app:srcCompat="@drawable/savebutt"
        android:layout_alignParentBottom="true"
        android:layout_toRightOf="@+id/textView"
        android:layout_toEndOf="@+id/textView"
        android:layout_weight="1.02" />


</LinearLayout>

<TextView
    android:text="Location Not Available"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/textView5"
    android:layout_alignLeft="@+id/sfMunsellChip"
    android:layout_alignStart="@+id/sfMunsellChip"
    android:id="@+id/textView6"
    />

<TextView
    android:text="Location:"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="22dp"
    android:id="@+id/textView5"
    android:textStyle="bold"
    android:textSize="16sp"
    android:layout_below="@+id/amMunsellText"
    android:layout_alignRight="@+id/textView4"
    android:layout_alignEnd="@+id/textView4" />

<TextView
    android:text="Notes:"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/textView"
    android:textSize="16sp"
    android:textStyle="bold"
    android:layout_below="@+id/textView5"
    android:layout_toLeftOf="@+id/sfMunsellChip"
    android:layout_toStartOf="@+id/sfMunsellChip"
    android:layout_marginTop="39dp" />

<TextView
    android:text="ID #:"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/textView4"
    android:layout_weight="1"
    android:textSize="16sp"
    android:textStyle="bold"
```

```
        android:layout_alignBaseline="@+id/sfIdEdit"
        android:layout_alignBottom="@+id/sfIdEdit"
        android:layout_toLeftOf="@+id/sfMunsellChip"
        android:layout_toStartOf="@+id/sfMunsellChip" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="textMultiLine"
        android:ems="10"
        android:id="@+id/sfNotesEdit"
        android:hint="Enter notes here..."
        android:maxLines="2"
        android:textSize="16sp"
        android:layout_alignBaseline="@+id/textView"
        android:layout_alignBottom="@+id/textView"
        android:layout_alignLeft="@+id/textView6"
        android:layout_alignStart="@+id/textView6" />
<!--The following widget is set to invisible for now, set to visible when testing-->
    <TextView
        android:text="Expected Munsell Chip:"
        android:visibility="invisible"
        android:textStyle="bold"
        android:textSize="16sp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="14dp"
        android:id="@+id/textView15"
        android:layout_below="@+id/sfNotesEdit"
        android:layout_alignLeft="@+id/textView5"
        android:layout_alignStart="@+id/textView5"
        android:layout_toLeftOf="@+id/sfNotesEdit"
        android:layout_toStartOf="@+id/sfNotesEdit" />
    <!--The following widget is set to invisible for now, set to visible when testing-
->

    <TextView
        android:text="Testing"
        android:visibility="invisible"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/textView15"
        android:layout_toRightOf="@+id/textView15"
        android:layout_toEndOf="@+id/textView15"
        android:layout_marginBottom="14dp"
        android:id="@+id/textView16"
        android:textSize="16sp"
        />
    <!--The following widget is set to invisible for now, set to visible when testing-
->

    <TextView
        android:text="testing"
        android:visibility="invisible"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@+id/textView17"
        android:layout_toRightOf="@+id/textView15"
        android:layout_toEndOf="@+id/textView15"
        android:id="@+id/textView18"
        android:textSize="16sp" />
    <!--The following widget is set to invisible for now, set to visible when testing-
->
```

```
    <TextView
        android:text="RGB Distance:"
        android:visibility="invisible"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textView17"
        android:layout_alignLeft="@+id/textView17"
        android:layout_alignStart="@+id/textView17"
        android:textSize="16sp"
        android:textStyle="bold"
        android:layout_marginTop="20dp"
        android:id="@+id/textView14" />
    <!--The following widget is set to invisible for now, set to visible when testing-
->

    <TextView
        android:text="Munsell Distance:"
        android:visibility="invisible"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="15dp"
        android:textSize="16sp"
        android:textStyle="bold"
        android:id="@+id/textView17"
        android:layout_below="@+id/textView15"
        android:layout_toLeftOf="@+id/textView18"
        android:layout_toStartOf="@+id/textView18"
        android:layout_marginRight="10dp"
        android:layout_marginEnd="10dp" />
    <!--The following widget is set to invisible for now, set to visible when testing-
->

    <TextView
        android:text="testing"
        android:visibility="invisible"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/textView19"
        android:layout_alignTop="@+id/textView14"
        android:layout_alignLeft="@+id/textView18"
        android:layout_alignStart="@+id/textView18" />

</RelativeLayout>
```

Appendix M : Munsell to RGB conversion chart

Hue	Value	Chroma	Red	Green	Blue
2.5R	1	2	45	21	31
2.5R	1	4	54	14	33
2.5R	1	6	62	3	34
2.5R	1	8	70	0	36
2.5R	1	10	78	0	38
2.5R	2	2	66	43	50
2.5R	2	4	77	36	49
2.5R	2	6	87	28	49
2.5R	2	8	98	14	49
2.5R	2	10	107	0	49
2.5R	2	12	117	0	50
2.5R	2	14	127	0	51
2.5R	3	2	92	65	71
2.5R	3	4	106	58	67
2.5R	3	6	117	50	64
2.5R	3	8	129	39	61
2.5R	3	10	140	21	58
2.5R	3	12	151	0	56
2.5R	3	14	160	0	54
2.5R	3	16	170	0	53
2.5R	4	2	116	90	96
2.5R	4	4	131	83	92
2.5R	4	6	144	76	88
2.5R	4	8	156	68	84
2.5R	4	10	167	57	81
2.5R	4	12	178	44	78
2.5R	4	14	189	18	75
2.5R	4	16	199	0	72
2.5R	4	18	210	0	70
2.5R	5	2	141	116	123
2.5R	5	4	156	109	118
2.5R	5	6	171	102	113
2.5R	5	8	184	95	108
2.5R	5	10	197	86	104
2.5R	5	12	209	74	100
2.5R	5	14	219	62	97

2.5R	5	16	231	42	93
2.5R	5	18	242	0	90
2.5R	5	20	253	0	88
2.5R	6	2	167	141	150
2.5R	6	4	183	135	145
2.5R	6	6	198	129	139
2.5R	6	8	211	122	134
2.5R	6	10	224	114	129
2.5R	6	12	237	104	124
2.5R	6	14	249	94	120
2.5R	6	16	255	79	116
2.5R	6	18	255	62	112
2.5R	7	2	193	167	177
2.5R	7	4	209	162	171
2.5R	7	6	225	155	165
2.5R	7	8	240	148	160
2.5R	7	10	254	140	154
2.5R	7	12	255	130	148
2.5R	7	14	255	120	144
2.5R	7	16	255	107	139
2.5R	8	2	218	194	205
2.5R	8	4	237	188	198
2.5R	8	6	254	181	192
2.5R	8	8	255	173	185
2.5R	8	10	255	165	179
2.5R	9	2	245	221	233
2.5R	9	4	255	214	225
2.5R	9	6	255	206	218
5.0R	1	2	46	21	29
5.0R	1	4	55	14	29
5.0R	1	6	64	2	29
5.0R	1	8	72	0	29
5.0R	1	10	79	0	30
5.0R	2	2	67	43	48
5.0R	2	4	78	36	45
5.0R	2	6	88	28	42
5.0R	2	8	99	13	40
5.0R	2	10	109	0	39
5.0R	2	12	118	0	38

5.0R	2	14	129	0	37
5.0R	3	2	92	65	69
5.0R	3	4	107	58	62
5.0R	3	6	118	51	57
5.0R	3	8	130	40	51
5.0R	3	10	141	23	46
5.0R	3	12	151	0	42
5.0R	3	14	160	0	38
5.0R	3	16	170	0	35
5.0R	4	2	117	90	94
5.0R	4	4	132	83	87
5.0R	4	6	145	76	80
5.0R	4	8	157	68	74
5.0R	4	10	168	58	67
5.0R	4	12	178	46	61
5.0R	4	14	189	24	55
5.0R	4	16	199	0	49
5.0R	4	18	209	0	44
5.0R	5	2	141	116	121
5.0R	5	4	158	109	114
5.0R	5	6	172	103	106
5.0R	5	8	185	95	99
5.0R	5	10	198	86	91
5.0R	5	12	210	75	84
5.0R	5	14	220	63	78
5.0R	5	16	232	45	71
5.0R	5	18	242	2	65
5.0R	5	20	252	0	60
5.0R	6	2	167	141	148
5.0R	6	4	184	135	140
5.0R	6	6	199	129	132
5.0R	6	8	212	122	126
5.0R	6	10	226	114	118
5.0R	6	12	239	105	111
5.0R	6	14	250	95	104
5.0R	6	16	255	82	96
5.0R	6	18	255	65	89
5.0R	7	2	194	167	175
5.0R	7	4	211	162	167

5.0R	7	6	226	155	159
5.0R	7	8	242	148	151
5.0R	7	10	255	140	144
5.0R	7	12	255	131	135
5.0R	7	14	255	121	127
5.0R	8	2	218	194	203
5.0R	8	4	239	188	194
5.0R	8	6	255	181	185
5.0R	8	8	255	173	176
5.0R	8	10	255	165	167
5.0R	9	2	246	221	231
5.0R	9	4	255	214	220
5.0R	9	6	255	206	210
7.5R	1	2	46	21	27
7.5R	1	4	56	14	25
7.5R	1	6	65	2	23
7.5R	1	8	73	0	23
7.5R	1	10	80	0	23
7.5R	2	2	67	43	46
7.5R	2	4	79	37	41
7.5R	2	6	89	29	36
7.5R	2	8	99	16	31
7.5R	2	10	109	0	27
7.5R	2	12	119	0	23
7.5R	2	14	128	0	21
7.5R	3	2	92	65	67
7.5R	3	4	106	59	58
7.5R	3	6	118	51	50
7.5R	3	8	129	42	42
7.5R	3	10	139	28	33
7.5R	3	12	150	0	25
7.5R	3	14	159	0	18
7.5R	3	16	168	0	9
7.5R	4	2	117	90	92
7.5R	4	4	132	83	83
7.5R	4	6	145	77	74
7.5R	4	8	157	69	65
7.5R	4	10	168	60	56
7.5R	4	12	178	49	46

7.5R	4	14	188	32	36
7.5R	4	16	197	0	26
7.5R	4	18	206	0	14
7.5R	4	20	216	0	0
7.5R	5	2	142	116	119
7.5R	5	4	158	109	109
7.5R	5	6	172	103	99
7.5R	5	8	186	96	89
7.5R	5	10	198	87	79
7.5R	5	12	210	77	68
7.5R	5	14	220	66	58
7.5R	5	16	230	51	46
7.5R	5	18	239	29	34
7.5R	5	20	248	0	21
7.5R	6	2	168	141	145
7.5R	6	4	185	136	136
7.5R	6	6	199	129	126
7.5R	6	8	213	123	116
7.5R	6	10	227	115	104
7.5R	6	12	239	107	93
7.5R	6	14	250	97	82
7.5R	6	16	255	86	69
7.5R	6	18	255	73	56
7.5R	7	2	195	167	173
7.5R	7	4	212	162	162
7.5R	7	6	227	156	152
7.5R	7	8	243	148	140
7.5R	7	10	255	141	130
7.5R	7	12	255	133	117
7.5R	7	14	255	124	105
7.5R	7	16	255	113	92
7.5R	8	2	219	194	201
7.5R	8	4	240	188	189
7.5R	8	6	255	181	177
7.5R	8	8	255	174	164
7.5R	8	10	255	166	153
7.5R	9	2	247	221	229
7.5R	9	4	255	214	214
7.5R	9	6	255	207	201

10.0R	1	2	46	22	25
10.0R	1	4	56	14	20
10.0R	1	6	65	2	17
10.0R	1	8	73	0	14
10.0R	1	10	81	0	12
10.0R	2	2	67	43	44
10.0R	2	4	78	37	36
10.0R	2	6	88	30	28
10.0R	2	8	98	18	20
10.0R	2	10	108	0	13
10.0R	2	12	117	0	5
10.0R	2	14	127	0	0
10.0R	3	2	92	65	64
10.0R	3	4	106	59	54
10.0R	3	6	117	53	44
10.0R	3	8	128	44	32
10.0R	3	10	137	33	20
10.0R	3	12	147	12	1
10.0R	3	14	156	0	0
10.0R	4	2	117	90	90
10.0R	4	4	132	84	78
10.0R	4	6	144	78	66
10.0R	4	8	156	71	54
10.0R	4	10	166	63	40
10.0R	4	12	174	55	22
10.0R	4	14	183	44	0
10.0R	4	16	189	32	0
10.0R	5	2	142	116	116
10.0R	5	4	158	110	104
10.0R	5	6	172	104	90
10.0R	5	8	185	97	77
10.0R	5	10	196	90	61
10.0R	5	12	206	82	44
10.0R	5	14	215	75	24
10.0R	5	16	222	66	0
10.0R	5	18	229	56	0
10.0R	6	2	169	141	143
10.0R	6	4	185	136	130
10.0R	6	6	199	130	117

10.0R	6	8	212	124	103
10.0R	6	10	225	117	88
10.0R	6	12	236	110	72
10.0R	6	14	246	102	52
10.0R	6	16	255	95	28
10.0R	6	18	255	86	0
10.0R	7	2	195	167	170
10.0R	7	4	212	162	156
10.0R	7	6	228	156	143
10.0R	7	8	242	150	128
10.0R	7	10	254	144	114
10.0R	7	12	255	136	97
10.0R	7	14	255	129	78
10.0R	7	16	255	121	56
10.0R	8	2	220	194	199
10.0R	8	4	241	188	183
10.0R	8	6	255	182	168
10.0R	8	8	255	176	152
10.0R	8	10	255	169	137
10.0R	9	2	248	221	226
10.0R	9	4	255	214	208
10.0R	9	6	255	208	192
2.5YR	1	2	46	22	21
2.5YR	1	4	57	14	12
2.5YR	1	6	65	4	5
2.5YR	1	8	73	0	0
2.5YR	2	2	66	44	42
2.5YR	2	4	78	38	31
2.5YR	2	6	87	32	20
2.5YR	2	8	96	22	2
2.5YR	3	2	91	66	62
2.5YR	3	4	104	61	49
2.5YR	3	6	114	55	35
2.5YR	3	8	123	49	18
2.5YR	3	10	130	43	0
2.5YR	4	2	117	90	87
2.5YR	4	4	131	85	73
2.5YR	4	6	142	80	58
2.5YR	4	8	152	75	42

2.5YR	4	10	160	69	20
2.5YR	4	12	166	65	0
2.5YR	5	2	142	116	113
2.5YR	5	4	157	111	99
2.5YR	5	6	170	106	83
2.5YR	5	8	182	100	65
2.5YR	5	10	191	95	46
2.5YR	5	12	198	90	21
2.5YR	5	14	204	86	0
2.5YR	5	16	208	82	0
2.5YR	6	2	169	141	139
2.5YR	6	4	184	137	125
2.5YR	6	6	198	132	109
2.5YR	6	8	210	126	93
2.5YR	6	10	221	121	74
2.5YR	6	12	230	116	52
2.5YR	6	14	238	111	21
2.5YR	6	16	243	107	0
2.5YR	6	18	248	103	0
2.5YR	7	2	195	168	167
2.5YR	7	4	211	163	151
2.5YR	7	6	226	158	134
2.5YR	7	8	239	152	117
2.5YR	7	10	250	147	100
2.5YR	7	12	255	141	78
2.5YR	7	14	255	136	51
2.5YR	7	16	255	131	4
2.5YR	7	18	255	128	0
2.5YR	7	20	255	125	0
2.5YR	8	2	221	194	195
2.5YR	8	4	240	189	176
2.5YR	8	6	255	184	159
2.5YR	8	8	255	178	141
2.5YR	8	10	255	173	122
2.5YR	8	12	255	167	101
2.5YR	9	2	249	221	223
2.5YR	9	4	255	215	202
2.5YR	9	6	255	210	182
5.0YR	1	2	45	23	17

5.0YR	1	4	55	16	2
5.0YR	2	2	65	44	39
5.0YR	2	4	75	40	25
5.0YR	2	6	84	35	5
5.0YR	3	2	90	67	60
5.0YR	3	4	101	63	45
5.0YR	3	6	110	58	27
5.0YR	3	8	117	54	1
5.0YR	4	2	116	91	84
5.0YR	4	4	129	87	67
5.0YR	4	6	139	83	50
5.0YR	4	8	147	79	31
5.0YR	4	10	153	75	0
5.0YR	4	12	158	72	0
5.0YR	5	2	141	116	111
5.0YR	5	4	155	112	93
5.0YR	5	6	167	108	74
5.0YR	5	8	177	104	54
5.0YR	5	10	184	100	31
5.0YR	5	12	190	97	0
5.0YR	5	14	194	94	0
5.0YR	6	2	168	142	137
5.0YR	6	4	182	138	119
5.0YR	6	6	195	134	100
5.0YR	6	8	206	130	80
5.0YR	6	10	215	126	58
5.0YR	6	12	222	122	29
5.0YR	6	14	227	119	0
5.0YR	6	16	231	117	0
5.0YR	6	18	234	115	0
5.0YR	7	2	195	168	164
5.0YR	7	4	210	164	145
5.0YR	7	6	223	160	125
5.0YR	7	8	234	156	106
5.0YR	7	10	244	152	85
5.0YR	7	12	253	148	58
5.0YR	7	14	255	144	17
5.0YR	7	16	255	141	0
5.0YR	7	18	255	139	0

5.0YR	7	20	255	138	0
5.0YR	8	2	221	194	191
5.0YR	8	4	238	190	171
5.0YR	8	6	251	186	151
5.0YR	8	8	255	182	129
5.0YR	8	10	255	178	108
5.0YR	8	12	255	174	83
5.0YR	8	14	255	170	53
5.0YR	9	2	249	221	218
5.0YR	9	4	255	217	195
5.0YR	9	6	255	212	174
7.5YR	1	2	43	24	14
7.5YR	2	2	63	45	37
7.5YR	2	4	73	42	20
7.5YR	2	6	80	39	0
7.5YR	3	2	89	67	58
7.5YR	3	4	98	64	41
7.5YR	3	6	106	61	20
7.5YR	3	8	111	59	0
7.5YR	4	2	115	92	82
7.5YR	4	4	127	88	62
7.5YR	4	6	135	85	43
7.5YR	4	8	141	83	19
7.5YR	4	10	146	80	0
7.5YR	5	2	140	117	109
7.5YR	5	4	153	114	89
7.5YR	5	6	163	110	67
7.5YR	5	8	171	107	44
7.5YR	5	10	177	105	10
7.5YR	5	12	181	103	0
7.5YR	5	14	184	102	0
7.5YR	6	2	167	143	134
7.5YR	6	4	180	139	115
7.5YR	6	6	191	136	93
7.5YR	6	8	200	133	70
7.5YR	6	10	207	130	43
7.5YR	6	12	213	128	0
7.5YR	6	14	216	126	0
7.5YR	6	16	219	125	0

7.5YR	7	2	194	169	161
7.5YR	7	4	207	165	140
7.5YR	7	6	219	162	118
7.5YR	7	8	228	159	96
7.5YR	7	10	236	156	72
7.5YR	7	12	244	153	38
7.5YR	7	14	249	151	0
7.5YR	7	16	252	150	0
7.5YR	7	18	255	149	0
7.5YR	8	2	221	195	187
7.5YR	8	4	235	192	166
7.5YR	8	6	247	188	144
7.5YR	8	8	255	185	120
7.5YR	8	10	255	182	96
7.5YR	8	12	255	180	67
7.5YR	8	14	255	177	24
7.5YR	8	16	255	175	0
7.5YR	8	18	255	174	0
7.5YR	8	20	255	173	0
7.5YR	9	2	249	222	213
7.5YR	9	4	255	218	189
7.5YR	9	6	255	215	166
7.5YR	9	8	255	212	143
10.0YR	1	2	42	25	11
10.0YR	2	2	62	46	36
10.0YR	2	4	70	44	15
10.0YR	3	2	86	69	56
10.0YR	3	4	95	66	37
10.0YR	3	6	101	64	12
10.0YR	3	8	106	62	0
10.0YR	4	2	112	93	79
10.0YR	4	4	123	90	58
10.0YR	4	6	130	88	36
10.0YR	4	8	135	86	3
10.0YR	4	10	138	85	0
10.0YR	5	2	138	118	106
10.0YR	5	4	149	116	84
10.0YR	5	6	158	113	59
10.0YR	5	8	164	111	32

10.0YR	5	10	169	110	0
10.0YR	5	12	172	109	0
10.0YR	6	2	165	144	131
10.0YR	6	4	176	141	109
10.0YR	6	6	186	139	85
10.0YR	6	8	193	137	60
10.0YR	6	10	199	135	26
10.0YR	6	12	203	134	0
10.0YR	6	14	206	133	0
10.0YR	7	2	192	170	157
10.0YR	7	4	203	167	134
10.0YR	7	6	213	165	110
10.0YR	7	8	221	163	86
10.0YR	7	10	228	161	58
10.0YR	7	12	233	159	5
10.0YR	7	14	237	158	0
10.0YR	7	16	240	157	0
10.0YR	7	18	242	156	0
10.0YR	8	2	219	196	184
10.0YR	8	4	230	194	160
10.0YR	8	6	241	192	136
10.0YR	8	8	250	189	111
10.0YR	8	10	255	187	84
10.0YR	8	12	255	185	51
10.0YR	8	14	255	184	0
10.0YR	8	16	255	182	0
10.0YR	8	18	255	182	0
10.0YR	8	20	255	181	0
10.0YR	9	2	247	223	209
10.0YR	9	4	255	221	183
10.0YR	9	6	255	219	159
10.0YR	9	8	255	216	133
2.5Y	1	2	39	27	9
2.5Y	2	2	59	47	34
2.5Y	2	4	66	46	10
2.5Y	3	2	84	70	55
2.5Y	3	4	91	68	34
2.5Y	3	6	96	67	4
2.5Y	4	2	110	94	78

2.5Y	4	4	118	92	55
2.5Y	4	6	124	91	30
2.5Y	4	8	128	90	0
2.5Y	4	10	131	89	0
2.5Y	5	2	136	119	104
2.5Y	5	4	145	118	80
2.5Y	5	6	152	116	53
2.5Y	5	8	157	115	21
2.5Y	5	10	160	114	0
2.5Y	5	12	162	113	0
2.5Y	6	2	162	145	129
2.5Y	6	4	171	144	105
2.5Y	6	6	179	142	79
2.5Y	6	8	185	141	50
2.5Y	6	10	190	139	0
2.5Y	6	12	193	139	0
2.5Y	6	14	195	138	0
2.5Y	7	2	189	171	155
2.5Y	7	4	199	169	130
2.5Y	7	6	207	168	105
2.5Y	7	8	213	167	78
2.5Y	7	10	219	165	44
2.5Y	7	12	223	164	0
2.5Y	7	14	226	164	0
2.5Y	7	16	228	163	0
2.5Y	8	2	216	197	181
2.5Y	8	4	225	196	156
2.5Y	8	6	234	195	130
2.5Y	8	8	241	193	104
2.5Y	8	10	247	192	74
2.5Y	8	12	252	191	30
2.5Y	8	14	255	190	0
2.5Y	8	16	255	189	0
2.5Y	8	18	255	188	0
2.5Y	8	20	255	188	0
2.5Y	9	2	244	224	207
2.5Y	9	4	254	223	179
2.5Y	9	6	255	222	153
2.5Y	9	8	255	220	126

2.5Y	9	10	255	219	98
2.5Y	9	12	255	218	63
5.0Y	1	2	37	28	9
5.0Y	2	2	57	48	34
5.0Y	2	4	63	47	6
5.0Y	3	2	81	71	55
5.0Y	3	4	87	70	33
5.0Y	3	6	91	69	0
5.0Y	4	2	107	95	77
5.0Y	4	4	113	94	52
5.0Y	4	6	118	94	25
5.0Y	4	8	121	93	0
5.0Y	5	2	133	120	103
5.0Y	5	4	140	120	77
5.0Y	5	6	145	119	48
5.0Y	5	8	149	118	10
5.0Y	5	10	151	118	0
5.0Y	5	12	153	118	0
5.0Y	6	2	159	146	128
5.0Y	6	4	166	146	102
5.0Y	6	6	172	145	74
5.0Y	6	8	177	144	42
5.0Y	6	10	180	144	0
5.0Y	6	12	182	143	0
5.0Y	6	14	183	143	0
5.0Y	7	2	186	172	153
5.0Y	7	4	193	172	127
5.0Y	7	6	198	171	100
5.0Y	7	8	204	171	71
5.0Y	7	10	208	170	29
5.0Y	7	12	211	170	0
5.0Y	7	14	212	169	0
5.0Y	7	16	214	169	0
5.0Y	8	2	213	198	179
5.0Y	8	4	219	198	153
5.0Y	8	6	225	198	125
5.0Y	8	8	231	198	96
5.0Y	8	10	235	197	63
5.0Y	8	12	238	197	0

5.0Y	8	14	241	196	0
5.0Y	8	16	242	196	0
5.0Y	8	18	243	196	0
5.0Y	9	2	241	225	205
5.0Y	9	4	248	225	177
5.0Y	9	6	253	225	149
5.0Y	9	8	255	225	119
5.0Y	9	10	255	225	89
5.0Y	9	12	255	224	45
5.0Y	9	14	255	224	0
5.0Y	9	16	255	223	0
5.0Y	9	18	255	223	0
5.0Y	9	20	255	223	0
7.5Y	1	2	34	28	11
7.5Y	2	2	55	49	35
7.5Y	2	4	59	49	5
7.5Y	3	2	79	71	55
7.5Y	3	4	82	72	32
7.5Y	3	6	85	71	0
7.5Y	4	2	104	96	77
7.5Y	4	4	108	96	51
7.5Y	4	6	111	96	23
7.5Y	4	8	113	96	0
7.5Y	5	2	130	121	102
7.5Y	5	4	135	121	76
7.5Y	5	6	138	122	46
7.5Y	5	8	141	122	0
7.5Y	5	10	142	121	0
7.5Y	5	12	144	121	0
7.5Y	6	2	157	147	127
7.5Y	6	4	161	147	101
7.5Y	6	6	165	148	72
7.5Y	6	8	168	148	37
7.5Y	6	10	170	148	0
7.5Y	6	12	172	147	0
7.5Y	6	14	173	147	0
7.5Y	7	2	184	173	153
7.5Y	7	4	188	173	126
7.5Y	7	6	192	174	99

7.5Y	7	8	195	174	68
7.5Y	7	10	198	174	19
7.5Y	7	12	200	174	0
7.5Y	7	14	201	174	0
7.5Y	7	16	202	174	0
7.5Y	8	2	211	199	178
7.5Y	8	4	215	200	152
7.5Y	8	6	219	200	124
7.5Y	8	8	222	200	93
7.5Y	8	10	225	201	58
7.5Y	8	12	228	201	0
7.5Y	8	14	229	201	0
7.5Y	8	16	230	200	0
7.5Y	8	18	231	200	0
7.5Y	9	2	239	226	204
7.5Y	9	4	243	227	176
7.5Y	9	6	247	227	147
7.5Y	9	8	251	228	117
7.5Y	9	10	254	228	85
7.5Y	9	12	255	228	36
7.5Y	9	14	255	228	0
7.5Y	9	16	255	228	0
7.5Y	9	18	255	228	0
10.0Y	1	2	31	29	14
10.0Y	2	2	52	50	36
10.0Y	2	4	54	50	8
10.0Y	3	2	76	72	56
10.0Y	3	4	78	73	32
10.0Y	3	6	79	74	0
10.0Y	4	2	101	97	78
10.0Y	4	4	103	98	52
10.0Y	4	6	105	98	22
10.0Y	4	8	106	99	0
10.0Y	5	2	128	122	103
10.0Y	5	4	130	123	77
10.0Y	5	6	131	124	47
10.0Y	5	8	133	124	0
10.0Y	5	10	133	125	0
10.0Y	5	12	134	125	0

10.0Y	6	2	154	148	127
10.0Y	6	4	157	149	102
10.0Y	6	6	159	150	73
10.0Y	6	8	160	150	37
10.0Y	6	10	161	151	0
10.0Y	6	12	161	151	0
10.0Y	6	14	162	151	0
10.0Y	7	2	181	174	153
10.0Y	7	4	184	175	126
10.0Y	7	6	186	176	99
10.0Y	7	8	187	176	68
10.0Y	7	10	189	177	14
10.0Y	7	12	189	177	0
10.0Y	7	14	190	178	0
10.0Y	7	16	190	178	0
10.0Y	8	2	208	200	178
10.0Y	8	4	211	201	152
10.0Y	8	6	213	202	123
10.0Y	8	8	215	203	93
10.0Y	8	10	216	204	57
10.0Y	8	12	217	204	0
10.0Y	8	14	218	204	0
10.0Y	8	16	218	205	0
10.0Y	8	18	218	205	0
10.0Y	9	2	236	227	204
10.0Y	9	4	239	228	175
10.0Y	9	6	241	229	147
10.0Y	9	8	243	230	116
10.0Y	9	10	245	231	83
10.0Y	9	12	246	231	32
10.0Y	9	14	247	232	0
10.0Y	9	16	247	232	0
10.0Y	9	18	248	232	0
2.5GY	1	2	28	30	17
2.5GY	2	2	50	50	38
2.5GY	2	4	48	52	15
2.5GY	3	2	73	73	57
2.5GY	3	4	72	75	35
2.5GY	3	6	71	76	3

2.5GY	4	2	98	98	80
2.5GY	4	4	97	99	55
2.5GY	4	6	96	101	28
2.5GY	4	8	95	102	0
2.5GY	5	2	124	123	105
2.5GY	5	4	123	125	80
2.5GY	5	6	123	126	52
2.5GY	5	8	122	127	11
2.5GY	5	10	120	128	0
2.5GY	5	12	120	129	0
2.5GY	6	2	151	149	129
2.5GY	6	4	150	151	104
2.5GY	6	6	149	152	77
2.5GY	6	8	148	154	43
2.5GY	6	10	148	155	0
2.5GY	6	12	147	155	0
2.5GY	6	14	146	156	0
2.5GY	7	2	178	175	154
2.5GY	7	4	177	177	129
2.5GY	7	6	177	178	104
2.5GY	7	8	176	180	73
2.5GY	7	10	175	181	27
2.5GY	7	12	174	182	0
2.5GY	7	14	174	182	0
2.5GY	7	16	173	183	0
2.5GY	8	2	205	201	179
2.5GY	8	4	205	203	154
2.5GY	8	6	204	205	127
2.5GY	8	8	203	206	99
2.5GY	8	10	203	208	62
2.5GY	8	12	202	209	0
2.5GY	8	14	201	209	0
2.5GY	8	16	201	210	0
2.5GY	8	18	201	210	0
2.5GY	9	2	233	228	204
2.5GY	9	4	233	230	177
2.5GY	9	6	233	232	149
2.5GY	9	8	232	234	119
2.5GY	9	10	231	235	87

2.5GY	9	12	230	236	37
2.5GY	9	14	229	237	0
2.5GY	9	16	229	238	0
2.5GY	9	18	229	238	0
5.0GY	1	2	26	30	19
5.0GY	1	4	18	33	0
5.0GY	2	2	48	51	40
5.0GY	2	4	43	53	21
5.0GY	2	6	37	55	0
5.0GY	3	2	70	74	59
5.0GY	3	4	66	76	40
5.0GY	3	6	61	78	15
5.0GY	3	8	57	80	0
5.0GY	4	2	95	98	82
5.0GY	4	4	91	101	60
5.0GY	4	6	87	103	36
5.0GY	4	8	83	105	0
5.0GY	4	10	79	106	0
5.0GY	5	2	121	124	107
5.0GY	5	4	117	126	85
5.0GY	5	6	113	129	61
5.0GY	5	8	109	131	30
5.0GY	5	10	105	132	0
5.0GY	5	12	102	133	0
5.0GY	6	2	148	149	131
5.0GY	6	4	144	152	108
5.0GY	6	6	140	155	85
5.0GY	6	8	136	157	56
5.0GY	6	10	132	158	7
5.0GY	6	12	129	160	0
5.0GY	6	14	126	161	0
5.0GY	7	2	175	175	156
5.0GY	7	4	171	178	133
5.0GY	7	6	167	181	110
5.0GY	7	8	164	183	83
5.0GY	7	10	160	185	48
5.0GY	7	12	156	186	0
5.0GY	7	14	153	187	0
5.0GY	7	16	151	188	0

5.0GY	8	2	202	202	181
5.0GY	8	4	199	204	157
5.0GY	8	6	196	207	132
5.0GY	8	8	192	209	107
5.0GY	8	10	188	211	76
5.0GY	8	12	184	213	25
5.0GY	8	14	181	215	0
5.0GY	8	16	178	216	0
5.0GY	8	18	176	216	0
5.0GY	8	20	176	217	0
5.0GY	9	2	230	229	206
5.0GY	9	4	227	232	179
5.0GY	9	6	224	234	153
5.0GY	9	8	220	237	125
5.0GY	9	10	216	239	95
5.0GY	9	12	212	241	55
5.0GY	9	14	209	242	0
5.0GY	9	16	206	244	0
5.0GY	9	18	204	244	0
7.5GY	1	2	24	31	22
7.5GY	1	4	9	34	5
7.5GY	2	2	45	51	43
7.5GY	2	4	36	54	28
7.5GY	2	6	25	57	10
7.5GY	2	8	2	59	0
7.5GY	3	2	67	74	63
7.5GY	3	4	58	77	47
7.5GY	3	6	49	80	30
7.5GY	3	8	37	82	2
7.5GY	3	10	17	84	0
7.5GY	4	2	91	99	86
7.5GY	4	4	82	103	68
7.5GY	4	6	73	105	49
7.5GY	4	8	62	108	25
7.5GY	4	10	50	110	0
7.5GY	4	12	33	112	0
7.5GY	5	2	117	125	111
7.5GY	5	4	109	128	93
7.5GY	5	6	99	131	74

7.5GY	5	8	90	134	52
7.5GY	5	10	79	136	20
7.5GY	5	12	68	138	0
7.5GY	5	14	56	140	0
7.5GY	6	2	142	151	135
7.5GY	6	4	134	154	117
7.5GY	6	6	126	157	98
7.5GY	6	8	117	160	75
7.5GY	6	10	107	163	50
7.5GY	6	12	97	165	4
7.5GY	6	14	86	167	0
7.5GY	6	16	77	169	0
7.5GY	7	2	169	177	160
7.5GY	7	4	161	180	142
7.5GY	7	6	153	183	122
7.5GY	7	8	144	187	101
7.5GY	7	10	135	189	77
7.5GY	7	12	125	192	46
7.5GY	7	14	115	194	0
7.5GY	7	16	106	196	0
7.5GY	7	18	99	197	0
7.5GY	8	2	196	203	186
7.5GY	8	4	187	207	166
7.5GY	8	6	179	210	145
7.5GY	8	8	171	213	124
7.5GY	8	10	162	216	101
7.5GY	8	12	152	219	72
7.5GY	8	14	142	222	26
7.5GY	8	16	133	224	0
7.5GY	8	18	125	225	0
7.5GY	8	20	119	226	0
7.5GY	9	2	223	230	211
7.5GY	9	4	214	234	189
7.5GY	9	6	206	238	166
7.5GY	9	8	197	242	144
7.5GY	9	10	188	245	119
7.5GY	9	12	179	247	92
7.5GY	9	14	169	250	57
7.5GY	9	16	159	252	0

7.5GY	9	18	150	254	0
10.0GY	1	2	22	31	24
10.0GY	1	4	1	35	13
10.0GY	1	6	0	37	2
10.0GY	2	2	43	51	45
10.0GY	2	4	30	55	35
10.0GY	2	6	11	57	24
10.0GY	2	8	0	60	10
10.0GY	2	10	0	62	0
10.0GY	2	12	0	64	0
10.0GY	3	2	64	75	66
10.0GY	3	4	52	78	54
10.0GY	3	6	37	81	41
10.0GY	3	8	10	84	27
10.0GY	3	10	0	86	9
10.0GY	3	12	0	88	0
10.0GY	3	14	0	90	0
10.0GY	4	2	88	99	89
10.0GY	4	4	76	103	75
10.0GY	4	6	61	107	61
10.0GY	4	8	42	110	45
10.0GY	4	10	0	113	27
10.0GY	4	12	0	115	0
10.0GY	4	14	0	117	0
10.0GY	4	16	0	119	0
10.0GY	5	2	113	125	114
10.0GY	5	4	101	129	100
10.0GY	5	6	88	133	86
10.0GY	5	8	72	136	70
10.0GY	5	10	51	139	53
10.0GY	5	12	4	142	30
10.0GY	5	14	0	144	0
10.0GY	5	16	0	146	0
10.0GY	5	18	0	148	0
10.0GY	6	2	138	151	139
10.0GY	6	4	127	155	124
10.0GY	6	6	114	159	109
10.0GY	6	8	99	163	93
10.0GY	6	10	82	166	77

10.0GY	6	12	60	169	58
10.0GY	6	14	11	172	33
10.0GY	6	16	0	174	0
10.0GY	6	18	0	176	0
10.0GY	6	20	0	178	0
10.0GY	7	2	164	177	165
10.0GY	7	4	153	182	149
10.0GY	7	6	141	185	135
10.0GY	7	8	127	189	119
10.0GY	7	10	111	192	102
10.0GY	7	12	92	196	83
10.0GY	7	14	68	199	63
10.0GY	7	16	21	201	35
10.0GY	7	18	0	204	0
10.0GY	7	20	0	206	0
10.0GY	7	22	0	208	0
10.0GY	8	2	191	204	190
10.0GY	8	4	179	208	174
10.0GY	8	6	166	213	158
10.0GY	8	8	153	216	143
10.0GY	8	10	137	220	125
10.0GY	8	12	120	223	106
10.0GY	8	14	99	226	87
10.0GY	8	16	71	229	64
10.0GY	8	18	12	232	28
10.0GY	8	20	0	235	0
10.0GY	8	22	0	237	0
10.0GY	8	24	0	239	0
10.0GY	9	2	217	231	216
10.0GY	9	4	204	236	199
10.0GY	9	6	189	241	180
10.0GY	9	8	176	245	164
10.0GY	9	10	160	249	145
10.0GY	9	12	144	252	128
10.0GY	9	14	124	255	108
10.0GY	9	16	99	255	84
10.0GY	9	18	64	255	55
2.5G	1	2	20	31	26
2.5G	1	4	0	35	19

2.5G	1	6	0	38	13
2.5G	1	8	0	41	8
2.5G	2	2	41	52	47
2.5G	2	4	24	55	40
2.5G	2	6	0	58	34
2.5G	2	8	0	60	29
2.5G	2	10	0	63	24
2.5G	2	12	0	65	19
2.5G	2	14	0	66	16
2.5G	2	16	0	68	13
2.5G	3	2	62	75	68
2.5G	3	4	45	79	61
2.5G	3	6	18	82	54
2.5G	3	8	0	85	48
2.5G	3	10	0	88	42
2.5G	3	12	0	90	36
2.5G	3	14	0	91	32
2.5G	3	16	0	93	28
2.5G	3	18	0	94	25
2.5G	3	20	0	95	22
2.5G	3	22	0	97	20
2.5G	4	2	85	100	93
2.5G	4	4	68	104	84
2.5G	4	6	43	108	75
2.5G	4	8	0	112	68
2.5G	4	10	0	115	60
2.5G	4	12	0	117	53
2.5G	4	14	0	119	47
2.5G	4	16	0	121	42
2.5G	4	18	0	123	37
2.5G	4	20	0	124	34
2.5G	4	22	0	125	30
2.5G	4	24	0	127	27
2.5G	4	26	0	128	24
2.5G	5	2	110	126	118
2.5G	5	4	94	130	109
2.5G	5	6	74	134	100
2.5G	5	8	46	138	91
2.5G	5	10	0	141	83

2.5G	5	12	0	144	74
2.5G	5	14	0	147	67
2.5G	5	16	0	149	59
2.5G	5	18	0	152	53
2.5G	5	20	0	153	47
2.5G	5	22	0	155	42
2.5G	5	24	0	156	37
2.5G	5	26	0	158	33
2.5G	5	28	0	159	29
2.5G	6	2	135	152	143
2.5G	6	4	118	157	133
2.5G	6	6	100	161	123
2.5G	6	8	78	164	114
2.5G	6	10	43	168	105
2.5G	6	12	0	171	97
2.5G	6	14	0	174	87
2.5G	6	16	0	177	79
2.5G	6	18	0	179	71
2.5G	6	20	0	182	64
2.5G	6	22	0	184	57
2.5G	6	24	0	186	50
2.5G	6	26	0	188	43
2.5G	6	28	0	189	37
2.5G	7	2	161	178	169
2.5G	7	4	144	183	159
2.5G	7	6	128	187	149
2.5G	7	8	107	191	138
2.5G	7	10	80	195	128
2.5G	7	12	32	198	118
2.5G	7	14	0	202	109
2.5G	7	16	0	205	100
2.5G	7	18	0	207	91
2.5G	7	20	0	210	82
2.5G	7	22	0	212	73
2.5G	7	24	0	214	66
2.5G	7	26	0	217	57
2.5G	8	2	186	205	195
2.5G	8	4	170	210	184
2.5G	8	6	151	215	173

2.5G	8	8	132	218	162
2.5G	8	10	109	222	152
2.5G	8	12	74	226	141
2.5G	8	14	0	229	131
2.5G	8	16	0	233	120
2.5G	8	18	0	236	110
2.5G	8	20	0	239	100
2.5G	8	22	0	241	91
2.5G	8	24	0	243	82
2.5G	9	2	212	232	222
2.5G	9	4	195	237	210
2.5G	9	6	174	243	196
2.5G	9	8	154	247	185
2.5G	9	10	130	251	173
2.5G	9	12	102	255	163
2.5G	9	14	56	255	151
2.5G	9	16	0	255	140
5.0G	1	2	19	31	27
5.0G	1	4	0	35	23
5.0G	1	6	0	38	20
5.0G	1	8	0	41	18
5.0G	2	2	40	52	48
5.0G	2	4	21	55	44
5.0G	2	6	0	58	40
5.0G	2	8	0	61	38
5.0G	2	10	0	63	35
5.0G	2	12	0	65	34
5.0G	2	14	0	66	33
5.0G	2	16	0	68	32
5.0G	3	2	60	75	70
5.0G	3	4	40	79	65
5.0G	3	6	0	83	61
5.0G	3	8	0	85	58
5.0G	3	10	0	88	55
5.0G	3	12	0	90	53
5.0G	3	14	0	92	51
5.0G	3	16	0	93	50
5.0G	3	18	0	94	49
5.0G	3	20	0	96	48

5.0G	3	22	0	97	47
5.0G	4	2	84	100	95
5.0G	4	4	63	105	89
5.0G	4	6	33	109	84
5.0G	4	8	0	112	79
5.0G	4	10	0	115	76
5.0G	4	12	0	118	73
5.0G	4	14	0	120	70
5.0G	4	16	0	122	69
5.0G	4	18	0	123	67
5.0G	4	20	0	125	66
5.0G	4	22	0	126	65
5.0G	4	24	0	127	65
5.0G	4	26	0	128	64
5.0G	5	2	108	126	121
5.0G	5	4	89	130	114
5.0G	5	6	66	135	109
5.0G	5	8	19	138	103
5.0G	5	10	0	142	99
5.0G	5	12	0	145	95
5.0G	5	14	0	148	92
5.0G	5	16	0	150	90
5.0G	5	18	0	152	88
5.0G	5	20	0	154	86
5.0G	5	22	0	155	85
5.0G	5	24	0	157	83
5.0G	5	26	0	158	82
5.0G	5	28	0	160	82
5.0G	6	2	133	152	146
5.0G	6	4	113	157	139
5.0G	6	6	91	161	133
5.0G	6	8	62	165	128
5.0G	6	10	0	169	124
5.0G	6	12	0	172	119
5.0G	6	14	0	175	115
5.0G	6	16	0	178	112
5.0G	6	18	0	180	110
5.0G	6	20	0	183	107
5.0G	6	22	0	185	105

5.0G	6	24	0	187	103
5.0G	6	26	0	188	102
5.0G	6	28	0	190	101
5.0G	7	2	158	178	173
5.0G	7	4	139	183	166
5.0G	7	6	119	188	160
5.0G	7	8	93	192	154
5.0G	7	10	51	196	148
5.0G	7	12	0	199	143
5.0G	7	14	0	203	138
5.0G	7	16	0	206	135
5.0G	7	18	0	209	132
5.0G	7	20	0	211	129
5.0G	7	22	0	214	126
5.0G	7	24	0	216	124
5.0G	7	26	0	217	123
5.0G	8	2	184	205	199
5.0G	8	4	165	210	192
5.0G	8	6	142	215	184
5.0G	8	8	117	220	178
5.0G	8	10	83	224	172
5.0G	8	12	0	228	167
5.0G	8	14	0	231	162
5.0G	8	16	0	234	158
5.0G	8	18	0	237	154
5.0G	8	20	0	240	151
5.0G	8	22	0	243	148
5.0G	9	2	210	232	226
5.0G	9	4	189	238	218
5.0G	9	6	163	244	209
5.0G	9	8	137	249	203
5.0G	9	10	105	253	197
5.0G	9	12	46	255	191
7.5G	1	2	18	31	28
7.5G	1	4	0	35	26
7.5G	1	6	0	38	25
7.5G	1	8	0	41	24
7.5G	2	2	39	52	49
7.5G	2	4	17	56	47

7.5G	2	6	0	58	45
7.5G	2	8	0	61	44
7.5G	2	10	0	63	43
7.5G	2	12	0	65	42
7.5G	2	14	0	67	42
7.5G	2	16	0	68	42
7.5G	3	2	59	75	72
7.5G	3	4	36	80	69
7.5G	3	6	0	83	66
7.5G	3	8	0	85	65
7.5G	3	10	0	88	63
7.5G	3	12	0	90	62
7.5G	3	14	0	92	62
7.5G	3	16	0	93	61
7.5G	3	18	0	95	61
7.5G	3	20	0	96	60
7.5G	3	22	0	97	60
7.5G	4	2	82	100	97
7.5G	4	4	60	105	93
7.5G	4	6	23	109	90
7.5G	4	8	0	112	87
7.5G	4	10	0	115	85
7.5G	4	12	0	118	84
7.5G	4	14	0	120	83
7.5G	4	16	0	122	82
7.5G	4	18	0	124	81
7.5G	4	20	0	125	81
7.5G	4	22	0	126	80
7.5G	4	24	0	127	80
7.5G	4	26	0	128	80
7.5G	5	2	107	126	123
7.5G	5	4	86	131	118
7.5G	5	6	60	135	115
7.5G	5	8	0	139	112
7.5G	5	10	0	142	109
7.5G	5	12	0	145	107
7.5G	5	14	0	148	105
7.5G	5	16	0	150	104
7.5G	5	18	0	152	103

7.5G	5	20	0	154	102
7.5G	5	22	0	156	101
7.5G	5	24	0	157	101
7.5G	5	26	0	158	100
7.5G	5	28	0	160	100
7.5G	6	2	132	152	149
7.5G	6	4	110	157	144
7.5G	6	6	86	161	140
7.5G	6	8	51	165	137
7.5G	6	10	0	169	134
7.5G	6	12	0	172	131
7.5G	6	14	0	175	129
7.5G	6	16	0	178	127
7.5G	6	18	0	181	125
7.5G	6	20	0	183	124
7.5G	6	22	0	185	123
7.5G	6	24	0	187	122
7.5G	6	26	0	188	121
7.5G	6	28	0	190	120
7.5G	7	2	157	178	175
7.5G	7	4	137	183	170
7.5G	7	6	115	188	166
7.5G	7	8	85	192	162
7.5G	7	10	26	196	158
7.5G	7	12	0	200	155
7.5G	7	14	0	203	152
7.5G	7	16	0	206	150
7.5G	7	18	0	209	148
7.5G	7	20	0	212	146
7.5G	7	22	0	214	145
7.5G	7	24	0	216	144
7.5G	7	26	0	217	143
7.5G	8	2	183	205	202
7.5G	8	4	162	210	197
7.5G	8	6	137	215	192
7.5G	8	8	110	220	188
7.5G	8	10	70	224	183
7.5G	8	12	0	228	180
7.5G	8	14	0	231	177

7.5G	8	16	0	235	174
7.5G	8	18	0	238	171
7.5G	8	20	0	240	169
7.5G	9	2	208	232	229
7.5G	9	4	186	238	223
7.5G	9	6	158	244	217
7.5G	9	8	128	249	212
7.5G	9	10	92	253	208
7.5G	9	12	0	255	204
10.0G	1	2	17	31	30
10.0G	1	4	0	35	29
10.0G	1	6	0	38	30
10.0G	1	8	0	41	31
10.0G	2	2	38	52	51
10.0G	2	4	13	56	50
10.0G	2	6	0	58	49
10.0G	2	8	0	61	49
10.0G	2	10	0	63	50
10.0G	2	12	0	65	50
10.0G	2	14	0	67	51
10.0G	2	16	0	68	52
10.0G	3	2	58	75	74
10.0G	3	4	32	80	72
10.0G	3	6	0	83	72
10.0G	3	8	0	86	71
10.0G	3	10	0	88	71
10.0G	3	12	0	90	71
10.0G	3	14	0	92	71
10.0G	3	16	0	94	72
10.0G	3	18	0	95	72
10.0G	3	20	0	96	73
10.0G	3	22	0	97	73
10.0G	4	2	81	100	99
10.0G	4	4	57	105	97
10.0G	4	6	10	109	96
10.0G	4	8	0	112	95
10.0G	4	10	0	115	94
10.0G	4	12	0	118	94
10.0G	4	14	0	120	94

10.0G	4	16	0	122	94
10.0G	4	18	0	124	94
10.0G	4	20	0	125	94
10.0G	4	22	0	126	95
10.0G	4	24	0	127	95
10.0G	4	26	0	128	95
10.0G	5	2	106	126	125
10.0G	5	4	84	131	123
10.0G	5	6	55	135	121
10.0G	5	8	0	139	119
10.0G	5	10	0	142	118
10.0G	5	12	0	146	117
10.0G	5	14	0	148	117
10.0G	5	16	0	150	117
10.0G	5	18	0	153	117
10.0G	5	20	0	154	116
10.0G	5	22	0	156	117
10.0G	5	24	0	157	117
10.0G	5	26	0	158	117
10.0G	5	28	0	159	117
10.0G	6	2	131	152	151
10.0G	6	4	107	157	148
10.0G	6	6	82	161	146
10.0G	6	8	38	165	145
10.0G	6	10	0	169	143
10.0G	6	12	0	173	142
10.0G	6	14	0	175	141
10.0G	6	16	0	178	140
10.0G	6	18	0	181	140
10.0G	6	20	0	183	139
10.0G	6	22	0	185	139
10.0G	6	24	0	187	139
10.0G	6	26	0	188	139
10.0G	7	2	156	178	178
10.0G	7	4	135	183	175
10.0G	7	6	110	188	172
10.0G	7	8	77	192	170
10.0G	7	10	0	196	168
10.0G	7	12	0	200	167

10.0G	7	14	0	203	166
10.0G	7	16	0	206	165
10.0G	7	18	0	209	164
10.0G	7	20	0	212	163
10.0G	7	22	0	214	162
10.0G	7	24	0	216	162
10.0G	8	2	182	205	205
10.0G	8	4	160	210	201
10.0G	8	6	134	215	199
10.0G	8	8	104	220	196
10.0G	8	10	56	224	194
10.0G	8	12	0	228	192
10.0G	8	14	0	231	191
10.0G	8	16	0	235	189
10.0G	8	18	0	238	188
10.0G	8	20	0	241	187
10.0G	9	2	208	232	232
10.0G	9	4	183	238	228
10.0G	9	6	154	244	225
10.0G	9	8	120	249	222
10.0G	9	10	77	253	220
10.0G	9	12	0	255	218
2.5BG	1	2	15	32	31
2.5BG	1	4	0	35	33
2.5BG	1	6	0	38	36
2.5BG	1	8	0	41	39
2.5BG	2	2	37	52	52
2.5BG	2	4	8	56	54
2.5BG	2	6	0	58	55
2.5BG	2	8	0	61	57
2.5BG	2	10	0	63	58
2.5BG	2	12	0	65	60
2.5BG	2	14	0	67	61
2.5BG	3	2	57	75	76
2.5BG	3	4	28	80	77
2.5BG	3	6	0	83	78
2.5BG	3	8	0	86	79
2.5BG	3	10	0	88	81
2.5BG	3	12	0	90	82

2.5BG	3	14	0	92	83
2.5BG	3	16	0	94	85
2.5BG	3	18	0	95	86
2.5BG	3	20	0	96	87
2.5BG	4	2	80	100	101
2.5BG	4	4	54	105	102
2.5BG	4	6	0	109	102
2.5BG	4	8	0	112	103
2.5BG	4	10	0	115	105
2.5BG	4	12	0	118	106
2.5BG	4	14	0	120	108
2.5BG	4	16	0	122	109
2.5BG	4	18	0	124	110
2.5BG	4	20	0	125	111
2.5BG	4	22	0	126	112
2.5BG	4	24	0	127	113
2.5BG	5	2	105	126	127
2.5BG	5	4	82	131	127
2.5BG	5	6	51	135	127
2.5BG	5	8	0	139	128
2.5BG	5	10	0	142	129
2.5BG	5	12	0	146	130
2.5BG	5	14	0	148	131
2.5BG	5	16	0	150	133
2.5BG	5	18	0	153	134
2.5BG	5	20	0	155	135
2.5BG	5	22	0	156	136
2.5BG	5	24	0	157	137
2.5BG	6	2	130	152	153
2.5BG	6	4	105	157	153
2.5BG	6	6	78	161	152
2.5BG	6	8	19	165	153
2.5BG	6	10	0	169	154
2.5BG	6	12	0	172	154
2.5BG	6	14	0	175	155
2.5BG	6	16	0	178	156
2.5BG	6	18	0	181	157
2.5BG	6	20	0	183	159
2.5BG	6	22	0	185	160

2.5BG	7	2	156	178	180
2.5BG	7	4	133	183	179
2.5BG	7	6	108	188	178
2.5BG	7	8	69	192	178
2.5BG	7	10	0	196	179
2.5BG	7	12	0	200	179
2.5BG	7	14	0	203	180
2.5BG	7	16	0	206	181
2.5BG	7	18	0	209	181
2.5BG	7	20	0	211	182
2.5BG	7	22	0	214	183
2.5BG	8	2	181	205	207
2.5BG	8	4	158	210	205
2.5BG	8	6	131	215	205
2.5BG	8	8	98	220	204
2.5BG	8	10	36	224	204
2.5BG	8	12	0	228	204
2.5BG	8	14	0	231	205
2.5BG	8	16	0	234	205
2.5BG	8	18	0	238	206
2.5BG	9	2	207	232	234
2.5BG	9	4	182	238	233
2.5BG	9	6	150	244	232
2.5BG	9	8	113	249	231
2.5BG	9	10	61	253	231
5.0BG	1	2	14	32	34
5.0BG	1	4	0	35	39
5.0BG	1	6	0	38	44
5.0BG	2	2	36	52	55
5.0BG	2	4	4	55	58
5.0BG	2	6	0	58	62
5.0BG	2	8	0	61	66
5.0BG	2	10	0	63	70
5.0BG	2	12	0	65	73
5.0BG	3	2	56	75	78
5.0BG	3	4	24	80	82
5.0BG	3	6	0	83	85
5.0BG	3	8	0	85	89
5.0BG	3	10	0	88	93

5.0BG	3	12	0	90	96
5.0BG	3	14	0	92	99
5.0BG	3	16	0	94	102
5.0BG	3	18	0	95	104
5.0BG	4	2	80	100	104
5.0BG	4	4	52	105	107
5.0BG	4	6	0	109	110
5.0BG	4	8	0	112	114
5.0BG	4	10	0	115	117
5.0BG	4	12	0	118	121
5.0BG	4	14	0	120	124
5.0BG	4	16	0	122	127
5.0BG	4	18	0	124	129
5.0BG	4	20	0	125	132
5.0BG	5	2	104	126	130
5.0BG	5	4	80	130	132
5.0BG	5	6	45	134	135
5.0BG	5	8	0	138	139
5.0BG	5	10	0	142	142
5.0BG	5	12	0	145	145
5.0BG	5	14	0	148	148
5.0BG	5	16	0	150	151
5.0BG	5	18	0	153	154
5.0BG	5	20	0	155	157
5.0BG	5	22	0	156	159
5.0BG	6	2	129	152	156
5.0BG	6	4	104	157	158
5.0BG	6	6	72	161	161
5.0BG	6	8	0	165	164
5.0BG	6	10	0	169	167
5.0BG	6	12	0	172	169
5.0BG	6	14	0	175	172
5.0BG	6	16	0	178	175
5.0BG	6	18	0	181	178
5.0BG	6	20	0	183	180
5.0BG	7	2	155	178	183
5.0BG	7	4	131	183	185
5.0BG	7	6	105	187	186
5.0BG	7	8	60	192	189

5.0BG	7	10	0	196	191
5.0BG	7	12	0	199	194
5.0BG	7	14	0	203	196
5.0BG	7	16	0	206	199
5.0BG	7	18	0	209	202
5.0BG	7	20	0	211	204
5.0BG	8	2	181	205	210
5.0BG	8	4	157	210	211
5.0BG	8	6	129	215	213
5.0BG	8	8	89	219	215
5.0BG	8	10	0	224	217
5.0BG	8	12	0	228	219
5.0BG	8	14	0	231	221
5.0BG	8	16	0	234	223
5.0BG	9	2	207	232	237
5.0BG	9	4	180	238	238
5.0BG	9	6	147	244	240
5.0BG	9	8	105	249	241
5.0BG	9	10	37	253	243
7.5BG	1	2	13	31	36
7.5BG	1	4	0	35	42
7.5BG	1	6	0	38	49
7.5BG	2	2	36	52	56
7.5BG	2	4	1	55	62
7.5BG	2	6	0	58	67
7.5BG	2	8	0	61	73
7.5BG	2	10	0	63	78
7.5BG	2	12	0	65	82
7.5BG	3	2	56	75	81
7.5BG	3	4	21	79	87
7.5BG	3	6	0	82	92
7.5BG	3	8	0	85	97
7.5BG	3	10	0	88	102
7.5BG	3	12	0	90	106
7.5BG	3	14	0	92	110
7.5BG	3	16	0	94	114
7.5BG	4	2	79	100	106
7.5BG	4	4	51	104	111
7.5BG	4	6	0	108	117

7.5BG	4	8	0	112	123
7.5BG	4	10	0	115	128
7.5BG	4	12	0	117	133
7.5BG	4	14	0	120	138
7.5BG	4	16	0	122	142
7.5BG	4	18	0	124	145
7.5BG	5	2	104	126	132
7.5BG	5	4	79	130	137
7.5BG	5	6	40	134	142
7.5BG	5	8	0	138	147
7.5BG	5	10	0	141	153
7.5BG	5	12	0	145	158
7.5BG	5	14	0	147	163
7.5BG	5	16	0	150	167
7.5BG	5	18	0	153	172
7.5BG	6	2	129	152	159
7.5BG	6	4	103	157	164
7.5BG	6	6	70	161	168
7.5BG	6	8	0	165	173
7.5BG	6	10	0	168	178
7.5BG	6	12	0	172	183
7.5BG	6	14	0	175	187
7.5BG	6	16	0	177	191
7.5BG	6	18	0	180	196
7.5BG	7	2	155	178	185
7.5BG	7	4	130	183	190
7.5BG	7	6	103	187	194
7.5BG	7	8	55	191	198
7.5BG	7	10	0	195	203
7.5BG	7	12	0	199	208
7.5BG	7	14	0	202	212
7.5BG	7	16	0	205	217
7.5BG	7	18	0	208	221
7.5BG	8	2	181	204	213
7.5BG	8	4	157	209	216
7.5BG	8	6	125	214	221
7.5BG	8	8	84	219	226
7.5BG	8	10	0	223	230
7.5BG	8	12	0	227	234

7.5BG	8	14	0	230	238
7.5BG	8	16	0	234	242
7.5BG	9	2	207	231	241
7.5BG	9	4	179	238	245
7.5BG	9	6	145	243	249
7.5BG	9	8	97	248	254
7.5BG	9	10	0	252	255
10.0BG	1	2	12	31	38
10.0BG	1	4	0	34	46
10.0BG	1	6	0	37	55
10.0BG	2	2	35	52	58
10.0BG	2	4	0	55	66
10.0BG	2	6	0	57	74
10.0BG	2	8	0	60	82
10.0BG	2	10	0	63	89
10.0BG	3	2	55	75	83
10.0BG	3	4	21	79	91
10.0BG	3	6	0	82	99
10.0BG	3	8	0	84	106
10.0BG	3	10	0	87	114
10.0BG	3	12	0	90	120
10.0BG	3	14	0	92	127
10.0BG	4	2	79	100	108
10.0BG	4	4	51	104	116
10.0BG	4	6	0	107	124
10.0BG	4	8	0	111	132
10.0BG	4	10	0	114	140
10.0BG	4	12	0	117	147
10.0BG	4	14	0	119	154
10.0BG	4	16	0	121	159
10.0BG	5	2	105	125	134
10.0BG	5	4	80	129	142
10.0BG	5	6	39	133	150
10.0BG	5	8	0	137	157
10.0BG	5	10	0	140	165
10.0BG	5	12	0	144	173
10.0BG	5	14	0	146	179
10.0BG	5	16	0	149	186
10.0BG	6	2	130	151	161

10.0BG	6	4	105	156	168
10.0BG	6	6	70	160	176
10.0BG	6	8	0	164	183
10.0BG	6	10	0	167	190
10.0BG	6	12	0	170	197
10.0BG	6	14	0	173	204
10.0BG	6	16	0	176	211
10.0BG	6	18	0	179	218
10.0BG	7	2	156	177	188
10.0BG	7	4	131	182	195
10.0BG	7	6	103	186	201
10.0BG	7	8	52	190	209
10.0BG	7	10	0	194	216
10.0BG	7	12	0	198	223
10.0BG	7	14	0	201	230
10.0BG	7	16	0	205	238
10.0BG	8	2	183	204	215
10.0BG	8	4	157	209	222
10.0BG	8	6	125	214	228
10.0BG	8	8	82	218	235
10.0BG	8	10	0	222	242
10.0BG	8	12	0	226	249
10.0BG	8	14	0	230	255
10.0BG	9	2	209	231	243
10.0BG	9	4	180	237	250
10.0BG	9	6	144	242	255
2.5B	1	2	12	31	39
2.5B	1	4	0	34	49
2.5B	1	6	0	36	59
2.5B	2	2	35	52	60
2.5B	2	4	0	54	69
2.5B	2	6	0	57	79
2.5B	2	8	0	60	89
2.5B	2	10	0	62	99
2.5B	3	2	55	75	85
2.5B	3	4	22	78	95
2.5B	3	6	0	81	105
2.5B	3	8	0	84	114
2.5B	3	10	0	86	124

2.5B	3	12	0	89	133
2.5B	4	2	80	99	110
2.5B	4	4	53	103	120
2.5B	4	6	0	106	130
2.5B	4	8	0	109	140
2.5B	4	10	0	112	151
2.5B	4	12	0	115	159
2.5B	4	14	0	118	169
2.5B	4	16	0	120	175
2.5B	5	2	106	125	136
2.5B	5	4	82	129	146
2.5B	5	6	43	132	155
2.5B	5	8	0	135	165
2.5B	5	10	0	139	175
2.5B	5	12	0	142	186
2.5B	5	14	0	145	194
2.5B	5	16	0	148	204
2.5B	6	2	131	151	163
2.5B	6	4	107	155	172
2.5B	6	6	73	159	182
2.5B	6	8	0	162	192
2.5B	6	10	0	165	201
2.5B	6	12	0	169	211
2.5B	6	14	0	172	220
2.5B	6	16	0	175	230
2.5B	7	2	158	177	190
2.5B	7	4	133	181	199
2.5B	7	6	105	185	208
2.5B	7	8	58	189	217
2.5B	7	10	0	193	227
2.5B	7	12	0	196	238
2.5B	7	14	0	199	247
2.5B	7	16	0	203	255
2.5B	8	2	185	203	217
2.5B	8	4	158	208	226
2.5B	8	6	127	213	235
2.5B	8	8	84	217	245
2.5B	8	10	0	221	255
2.5B	8	12	0	225	255

2.5B	9	2	211	230	245
2.5B	9	4	181	236	255
5.0B	1	2	12	31	41
5.0B	1	4	0	33	52
5.0B	1	6	0	35	63
5.0B	2	2	36	51	61
5.0B	2	4	3	54	72
5.0B	2	6	0	56	84
5.0B	2	8	0	58	97
5.0B	2	10	0	60	108
5.0B	3	2	56	74	87
5.0B	3	4	26	77	99
5.0B	3	6	0	80	110
5.0B	3	8	0	82	122
5.0B	3	10	0	84	134
5.0B	3	12	0	86	145
5.0B	4	2	81	99	112
5.0B	4	4	57	102	123
5.0B	4	6	7	105	135
5.0B	4	8	0	107	148
5.0B	4	10	0	110	159
5.0B	4	12	0	112	171
5.0B	4	14	0	115	183
5.0B	5	2	107	124	138
5.0B	5	4	85	128	149
5.0B	5	6	53	131	161
5.0B	5	8	0	133	173
5.0B	5	10	0	136	185
5.0B	5	12	0	139	198
5.0B	5	14	0	141	210
5.0B	5	16	0	144	223
5.0B	6	2	134	150	164
5.0B	6	4	112	154	175
5.0B	6	6	82	157	188
5.0B	6	8	25	160	200
5.0B	6	10	0	163	212
5.0B	6	12	0	165	224
5.0B	6	14	0	168	237
5.0B	6	16	0	172	250

5.0B	7	2	160	176	191
5.0B	7	4	138	180	203
5.0B	7	6	111	183	214
5.0B	7	8	72	187	226
5.0B	7	10	0	190	240
5.0B	7	12	0	193	254
5.0B	7	14	0	196	255
5.0B	8	2	188	202	218
5.0B	8	4	163	207	230
5.0B	8	6	135	210	243
5.0B	8	8	92	214	255
5.0B	9	2	214	229	247
5.0B	9	4	185	235	255
7.5B	1	2	14	30	42
7.5B	1	4	0	32	54
7.5B	1	6	0	33	65
7.5B	1	8	0	35	77
7.5B	2	2	37	51	63
7.5B	2	4	10	53	75
7.5B	2	6	0	55	88
7.5B	2	8	0	56	101
7.5B	2	10	0	58	114
7.5B	3	2	57	74	88
7.5B	3	4	34	76	102
7.5B	3	6	0	78	114
7.5B	3	8	0	80	126
7.5B	3	10	0	82	139
7.5B	3	12	0	84	152
7.5B	4	2	83	98	113
7.5B	4	4	63	101	126
7.5B	4	6	33	103	138
7.5B	4	8	0	105	152
7.5B	4	10	0	107	164
7.5B	4	12	0	109	177
7.5B	4	14	0	111	190
7.5B	5	2	109	124	139
7.5B	5	4	90	126	152
7.5B	5	6	65	129	164
7.5B	5	8	3	131	177

7.5B	5	10	0	133	191
7.5B	5	12	0	135	205
7.5B	5	14	0	138	218
7.5B	5	16	0	140	233
7.5B	6	2	136	149	165
7.5B	6	4	118	152	178
7.5B	6	6	93	155	191
7.5B	6	8	57	157	205
7.5B	6	10	0	160	218
7.5B	6	12	0	162	232
7.5B	6	14	0	165	247
7.5B	6	16	0	167	255
7.5B	7	2	163	175	192
7.5B	7	4	143	179	205
7.5B	7	6	120	181	218
7.5B	7	8	88	184	232
7.5B	7	10	0	187	247
7.5B	7	12	0	190	255
7.5B	8	2	190	201	219
7.5B	8	4	168	205	233
7.5B	8	6	143	209	247
7.5B	8	8	106	212	255
7.5B	9	2	217	228	247
7.5B	9	4	191	233	255
10.0B	1	2	16	30	44
10.0B	1	4	0	31	55
10.0B	1	6	0	32	66
10.0B	1	8	0	33	79
10.0B	2	2	39	50	64
10.0B	2	4	19	52	77
10.0B	2	6	0	53	90
10.0B	2	8	0	55	104
10.0B	2	10	0	56	117
10.0B	3	2	60	73	90
10.0B	3	4	41	75	103
10.0B	3	6	0	76	116
10.0B	3	8	0	78	129
10.0B	3	10	0	79	142
10.0B	3	12	0	81	156

10.0B	3	14	0	82	169
10.0B	4	2	86	97	114
10.0B	4	4	69	99	127
10.0B	4	6	47	101	140
10.0B	4	8	0	103	154
10.0B	4	10	0	104	167
10.0B	4	12	0	106	180
10.0B	4	14	0	108	194
10.0B	4	16	0	109	208
10.0B	5	2	112	123	140
10.0B	5	4	97	125	153
10.0B	5	6	77	127	167
10.0B	5	8	45	129	180
10.0B	5	10	0	130	194
10.0B	5	12	0	132	208
10.0B	5	14	0	134	222
10.0B	5	16	0	136	238
10.0B	5	18	0	138	251
10.0B	6	2	139	148	166
10.0B	6	4	124	151	179
10.0B	6	6	104	153	194
10.0B	6	8	78	155	208
10.0B	6	10	31	157	221
10.0B	6	12	0	159	237
10.0B	6	14	0	161	253
10.0B	6	16	0	163	255
10.0B	7	2	166	174	193
10.0B	7	4	149	177	206
10.0B	7	6	130	179	220
10.0B	7	8	106	181	235
10.0B	7	10	67	183	252
10.0B	7	12	0	186	255
10.0B	8	2	193	200	220
10.0B	8	4	175	203	235
10.0B	8	6	154	206	250
10.0B	8	8	124	209	255
10.0B	9	2	220	227	248
10.0B	9	4	197	231	255
2.5PB	1	2	19	29	44

2.5PB	1	4	3	30	56
2.5PB	1	6	0	30	68
2.5PB	1	8	0	30	80
2.5PB	2	2	41	50	65
2.5PB	2	4	28	50	78
2.5PB	2	6	0	51	92
2.5PB	2	8	0	52	106
2.5PB	2	10	0	52	119
2.5PB	2	12	0	52	132
2.5PB	3	2	63	72	91
2.5PB	3	4	50	73	105
2.5PB	3	6	29	74	118
2.5PB	3	8	0	75	131
2.5PB	3	10	0	75	144
2.5PB	3	12	0	76	158
2.5PB	3	14	0	77	172
2.5PB	4	2	89	97	115
2.5PB	4	4	77	98	129
2.5PB	4	6	62	99	142
2.5PB	4	8	35	100	156
2.5PB	4	10	0	100	169
2.5PB	4	12	0	101	183
2.5PB	4	14	0	102	197
2.5PB	4	16	0	103	212
2.5PB	4	18	0	103	226
2.5PB	5	2	115	122	141
2.5PB	5	4	103	123	154
2.5PB	5	6	89	124	169
2.5PB	5	8	70	125	183
2.5PB	5	10	39	126	197
2.5PB	5	12	0	127	211
2.5PB	5	14	0	128	225
2.5PB	5	16	0	129	241
2.5PB	5	18	0	130	255
2.5PB	6	2	142	147	167
2.5PB	6	4	131	149	180
2.5PB	6	6	116	150	195
2.5PB	6	8	99	151	210
2.5PB	6	10	76	152	225

2.5PB	6	12	21	154	241
2.5PB	6	14	155	255	
2.5PB	7	2	169	173	193
2.5PB	7	4	157	175	208
2.5PB	7	6	142	176	222
2.5PB	7	8	125	178	238
2.5PB	7	10	100	179	255
2.5PB	8	2	196	199	220
2.5PB	8	4	182	201	236
2.5PB	8	6	166	203	253
2.5PB	9	2	224	226	248
5.0PB	1	2	23	28	45
5.0PB	1	4	13	28	56
5.0PB	1	6	0	28	68
5.0PB	1	8	0	27	81
5.0PB	1	10	0	25	94
5.0PB	2	2	44	49	66
5.0PB	2	4	35	49	79
5.0PB	2	6	20	49	92
5.0PB	2	8	0	49	106
5.0PB	2	10	0	48	119
5.0PB	2	12	0	47	132
5.0PB	2	14	0	45	146
5.0PB	3	2	66	71	91
5.0PB	3	4	57	72	105
5.0PB	3	6	45	72	119
5.0PB	3	8	26	72	132
5.0PB	3	10	0	72	145
5.0PB	3	12	0	71	159
5.0PB	3	14	0	71	173
5.0PB	3	16	0	70	187
5.0PB	3	18	0	69	202
5.0PB	4	2	92	96	116
5.0PB	4	4	84	96	129
5.0PB	4	6	74	96	143
5.0PB	4	8	59	97	157
5.0PB	4	10	38	97	170
5.0PB	4	12	0	97	183
5.0PB	4	14	0	97	198

5.0PB	4	16	0	96	212
5.0PB	4	18	0	96	228
5.0PB	4	20	0	95	245
5.0PB	5	2	119	121	141
5.0PB	5	4	111	121	155
5.0PB	5	6	101	122	169
5.0PB	5	8	88	122	183
5.0PB	5	10	72	122	198
5.0PB	5	12	46	123	212
5.0PB	5	14	0	123	226
5.0PB	5	16	0	123	243
5.0PB	5	18	0	123	255
5.0PB	6	2	145	147	167
5.0PB	6	4	137	147	181
5.0PB	6	6	127	148	196
5.0PB	6	8	115	148	211
5.0PB	6	10	101	148	225
5.0PB	6	12	78	149	242
5.0PB	6	14	39	149	255
5.0PB	7	2	172	172	193
5.0PB	7	4	163	173	208
5.0PB	7	6	152	174	223
5.0PB	7	8	141	174	239
5.0PB	7	10	124	175	255
5.0PB	8	2	198	199	220
5.0PB	8	4	189	200	236
5.0PB	8	6	177	200	254
5.0PB	9	2	226	225	248
7.5PB	1	2	27	27	45
7.5PB	1	4	25	25	56
7.5PB	1	6	23	23	68
7.5PB	1	8	24	19	78
7.5PB	1	10	25	14	87
7.5PB	1	12	28	6	95
7.5PB	1	14	31	0	104
7.5PB	1	16	36	0	113
7.5PB	1	18	39	0	120
7.5PB	1	20	43	0	128
7.5PB	1	22	47	0	136

7.5PB	1	24	51	0	145
7.5PB	1	26	56	0	155
7.5PB	1	28	61	0	165
7.5PB	1	30	66	0	176
7.5PB	1	32	72	0	189
7.5PB	1	34	77	0	199
7.5PB	1	36	82	0	210
7.5PB	1	38	88	0	223
7.5PB	2	2	48	48	66
7.5PB	2	4	45	46	79
7.5PB	2	6	42	45	91
7.5PB	2	8	39	42	104
7.5PB	2	10	37	39	116
7.5PB	2	12	37	35	126
7.5PB	2	14	39	30	136
7.5PB	2	16	41	22	145
7.5PB	2	18	44	8	155
7.5PB	2	20	48	0	164
7.5PB	2	22	51	0	174
7.5PB	2	24	55	0	182
7.5PB	2	26	59	0	190
7.5PB	2	28	62	0	198
7.5PB	2	30	66	0	208
7.5PB	2	32	70	0	217
7.5PB	2	34	74	0	227
7.5PB	2	36	79	0	237
7.5PB	2	38	83	0	248
7.5PB	3	2	71	70	92
7.5PB	3	4	67	69	105
7.5PB	3	6	64	68	119
7.5PB	3	8	60	66	131
7.5PB	3	10	55	64	144
7.5PB	3	12	52	61	156
7.5PB	3	14	51	57	168
7.5PB	3	16	51	52	180
7.5PB	3	18	52	47	191
7.5PB	3	20	53	40	200
7.5PB	3	22	56	30	211
7.5PB	3	24	59	13	221

7.5PB	3	26	61	0	230
7.5PB	3	28	64	0	239
7.5PB	3	30	68	0	250
7.5PB	3	32	72	0	255
7.5PB	3	34	75	0	255
7.5PB	4	2	96	94	116
7.5PB	4	4	93	94	129
7.5PB	4	6	89	93	142
7.5PB	4	8	84	91	156
7.5PB	4	10	79	90	169
7.5PB	4	12	74	88	182
7.5PB	4	14	70	86	195
7.5PB	4	16	66	82	208
7.5PB	4	18	64	79	221
7.5PB	4	20	62	73	235
7.5PB	4	22	62	68	245
7.5PB	4	24	63	62	255
7.5PB	4	26	64	54	255
7.5PB	5	2	122	120	141
7.5PB	5	4	118	119	154
7.5PB	5	6	114	118	169
7.5PB	5	8	110	117	183
7.5PB	5	10	105	116	196
7.5PB	5	12	99	114	210
7.5PB	5	14	93	113	225
7.5PB	5	16	86	110	239
7.5PB	5	18	80	107	254
7.5PB	5	20	74	104	255
7.5PB	6	2	148	146	167
7.5PB	6	4	144	145	181
7.5PB	6	6	140	144	196
7.5PB	6	8	136	143	209
7.5PB	6	10	131	142	223
7.5PB	6	12	124	141	240
7.5PB	6	14	116	139	255
7.5PB	7	2	175	171	193
7.5PB	7	4	171	171	208
7.5PB	7	6	166	170	223
7.5PB	7	8	160	169	240

7.5PB	7	10	155	168	255
7.5PB	8	2	202	198	220
7.5PB	8	4	197	197	236
7.5PB	8	6	191	197	254
7.5PB	9	2	229	225	248
10.0PB	1	2	32	26	45
10.0PB	1	4	33	23	55
10.0PB	1	6	35	19	65
10.0PB	1	8	38	14	73
10.0PB	1	10	41	7	81
10.0PB	1	12	44	0	90
10.0PB	1	14	48	0	98
10.0PB	1	16	53	0	106
10.0PB	1	18	56	0	113
10.0PB	1	20	61	0	121
10.0PB	1	22	65	0	129
10.0PB	1	24	70	0	138
10.0PB	1	26	75	0	149
10.0PB	1	28	82	0	160
10.0PB	1	30	88	0	172
10.0PB	2	2	52	46	66
10.0PB	2	4	53	44	77
10.0PB	2	6	54	41	88
10.0PB	2	8	56	38	99
10.0PB	2	10	59	33	110
10.0PB	2	12	62	27	118
10.0PB	2	14	65	17	128
10.0PB	2	16	69	2	137
10.0PB	2	18	73	0	146
10.0PB	2	20	77	0	155
10.0PB	2	22	82	0	164
10.0PB	2	24	86	0	173
10.0PB	2	26	91	0	182
10.0PB	2	28	95	0	191
10.0PB	2	30	101	0	202
10.0PB	2	32	107	0	212
10.0PB	2	34	112	0	222
10.0PB	3	2	75	69	91
10.0PB	3	4	75	67	104

10.0PB	3	6	76	64	116
10.0PB	3	8	78	61	127
10.0PB	3	10	79	57	139
10.0PB	3	12	81	53	149
10.0PB	3	14	84	47	160
10.0PB	3	16	88	39	171
10.0PB	3	18	91	30	180
10.0PB	3	20	95	16	189
10.0PB	3	22	99	0	199
10.0PB	3	24	103	0	208
10.0PB	3	26	107	0	218
10.0PB	3	28	112	0	227
10.0PB	3	30	117	0	237
10.0PB	3	32	123	0	248
10.0PB	3	34	127	0	255
10.0PB	4	2	99	93	116
10.0PB	4	4	100	91	128
10.0PB	4	6	100	89	141
10.0PB	4	8	101	87	153
10.0PB	4	10	101	84	165
10.0PB	4	12	102	80	177
10.0PB	4	14	104	77	188
10.0PB	4	16	107	72	199
10.0PB	4	18	109	66	210
10.0PB	4	20	113	58	222
10.0PB	4	22	116	51	230
10.0PB	4	24	119	41	240
10.0PB	4	26	124	22	251
10.0PB	4	28	127	0	255
10.0PB	4	30	131	0	255
10.0PB	5	2	125	119	140
10.0PB	5	4	125	117	153
10.0PB	5	6	125	115	168
10.0PB	5	8	125	113	180
10.0PB	5	10	126	110	192
10.0PB	5	12	126	107	205
10.0PB	5	14	127	104	218
10.0PB	5	16	128	100	230
10.0PB	5	18	129	96	241

10.0PB	5	20	131	91	253
10.0PB	5	22	135	84	255
10.0PB	6	2	151	145	166
10.0PB	6	4	151	143	180
10.0PB	6	6	150	141	194
10.0PB	6	8	150	139	207
10.0PB	6	10	150	137	220
10.0PB	6	12	150	134	234
10.0PB	6	14	150	131	249
10.0PB	6	16	151	127	255
10.0PB	7	2	177	171	193
10.0PB	7	4	176	169	207
10.0PB	7	6	176	167	221
10.0PB	7	8	176	165	236
10.0PB	7	10	175	163	252
10.0PB	7	12	175	159	255
10.0PB	8	2	204	197	219
10.0PB	8	4	203	196	234
10.0PB	8	6	202	194	251
10.0PB	8	8	202	191	255
10.0PB	9	2	232	224	247
10.0PB	9	4	230	222	255
2.5P	1	2	35	25	44
2.5P	1	4	38	21	53
2.5P	1	6	42	16	61
2.5P	1	8	46	9	69
2.5P	1	10	50	0	76
2.5P	1	12	55	0	84
2.5P	1	14	59	0	92
2.5P	1	16	64	0	99
2.5P	1	18	68	0	106
2.5P	1	20	73	0	114
2.5P	1	22	78	0	122
2.5P	1	24	83	0	129
2.5P	1	26	90	0	140
2.5P	2	2	55	46	65
2.5P	2	4	58	43	75
2.5P	2	6	62	39	84
2.5P	2	8	67	34	94

2.5P	2	10	71	27	103
2.5P	2	12	76	17	112
2.5P	2	14	80	1	120
2.5P	2	16	85	0	129
2.5P	2	18	90	0	137
2.5P	2	20	95	0	146
2.5P	2	22	100	0	155
2.5P	2	24	106	0	164
2.5P	2	26	111	0	173
2.5P	2	28	118	0	184
2.5P	2	30	126	0	196
2.5P	3	2	78	68	90
2.5P	3	4	82	65	102
2.5P	3	6	85	61	113
2.5P	3	8	89	57	123
2.5P	3	10	94	51	133
2.5P	3	12	98	45	142
2.5P	3	14	103	37	152
2.5P	3	16	108	25	161
2.5P	3	18	113	4	170
2.5P	3	20	117	0	178
2.5P	3	22	123	0	188
2.5P	3	24	128	0	197
2.5P	3	26	133	0	206
2.5P	3	28	138	0	215
2.5P	3	30	144	0	225
2.5P	3	32	150	0	235
2.5P	3	34	156	0	244
2.5P	4	2	102	93	115
2.5P	4	4	106	90	127
2.5P	4	6	109	87	137
2.5P	4	8	112	83	148
2.5P	4	10	116	79	158
2.5P	4	12	121	74	169
2.5P	4	14	125	69	179
2.5P	4	16	130	62	189
2.5P	4	18	135	53	199
2.5P	4	20	140	42	209
2.5P	4	22	144	29	217

2.5P	4	24	149	0	226
2.5P	4	26	154	0	237
2.5P	4	28	159	0	246
2.5P	4	30	164	0	255
2.5P	4	32	171	0	255
2.5P	5	2	128	118	139
2.5P	5	4	130	116	152
2.5P	5	6	133	113	164
2.5P	5	8	137	109	176
2.5P	5	10	141	106	186
2.5P	5	12	145	101	198
2.5P	5	14	149	97	208
2.5P	5	16	153	91	218
2.5P	5	18	158	86	228
2.5P	5	20	162	78	239
2.5P	5	22	167	69	250
2.5P	5	24	172	58	255
2.5P	5	26	178	42	255
2.5P	6	2	153	144	166
2.5P	6	4	156	142	178
2.5P	6	6	159	139	191
2.5P	6	8	162	136	203
2.5P	6	10	166	132	215
2.5P	6	12	170	128	227
2.5P	6	14	174	124	238
2.5P	6	16	179	118	251
2.5P	6	18	183	113	255
2.5P	7	2	180	170	192
2.5P	7	4	182	168	205
2.5P	7	6	185	165	218
2.5P	7	8	188	162	231
2.5P	7	10	192	158	245
2.5P	7	12	196	154	255
2.5P	8	2	206	196	218
2.5P	8	4	209	194	233
2.5P	8	6	211	191	248
2.5P	8	8	214	188	255
2.5P	9	2	234	223	247
2.5P	9	4	236	221	255

5.0P	1	2	37	24	43
5.0P	1	4	42	20	51
5.0P	1	6	47	14	58
5.0P	1	8	51	5	65
5.0P	1	10	56	0	72
5.0P	1	12	61	0	79
5.0P	1	14	67	0	87
5.0P	1	16	72	0	94
5.0P	1	18	77	0	100
5.0P	1	20	82	0	107
5.0P	1	22	88	0	115
5.0P	2	2	57	45	64
5.0P	2	4	62	41	73
5.0P	2	6	67	37	80
5.0P	2	8	73	30	90
5.0P	2	10	78	22	98
5.0P	2	12	84	8	106
5.0P	2	14	90	0	115
5.0P	2	16	95	0	122
5.0P	2	18	100	0	130
5.0P	2	20	106	0	138
5.0P	2	22	112	0	146
5.0P	2	24	118	0	155
5.0P	2	26	126	0	165
5.0P	2	28	133	0	175
5.0P	3	2	81	67	89
5.0P	3	4	87	63	99
5.0P	3	6	92	59	108
5.0P	3	8	97	54	118
5.0P	3	10	103	47	127
5.0P	3	12	109	39	136
5.0P	3	14	115	28	145
5.0P	3	16	120	7	153
5.0P	3	18	126	0	161
5.0P	3	20	131	0	168
5.0P	3	22	137	0	177
5.0P	3	24	143	0	186
5.0P	3	26	149	0	195
5.0P	3	28	156	0	204

5.0P	3	30	162	0	213
5.0P	3	32	170	0	223
5.0P	4	2	105	92	113
5.0P	4	4	110	88	124
5.0P	4	6	115	85	133
5.0P	4	8	121	81	143
5.0P	4	10	126	75	152
5.0P	4	12	132	70	162
5.0P	4	14	138	63	171
5.0P	4	16	144	53	181
5.0P	4	18	150	42	189
5.0P	4	20	156	25	198
5.0P	4	22	161	0	206
5.0P	4	24	167	0	214
5.0P	4	26	173	0	223
5.0P	4	28	179	0	233
5.0P	4	30	185	0	242
5.0P	4	32	193	0	253
5.0P	5	2	130	118	138
5.0P	5	4	135	114	149
5.0P	5	6	141	111	160
5.0P	5	8	146	106	170
5.0P	5	10	152	102	180
5.0P	5	12	157	97	190
5.0P	5	14	163	91	199
5.0P	5	16	168	85	209
5.0P	5	18	174	77	218
5.0P	5	20	180	68	228
5.0P	5	22	186	56	237
5.0P	5	24	193	38	247
5.0P	5	26	199	0	255
5.0P	5	28	206	0	255
5.0P	6	2	156	143	165
5.0P	6	4	161	140	175
5.0P	6	6	166	137	187
5.0P	6	8	171	133	197
5.0P	6	10	177	129	208
5.0P	6	12	183	124	218
5.0P	6	14	189	118	229

5.0P	6	16	195	112	240
5.0P	6	18	200	106	250
5.0P	6	20	207	97	255
5.0P	7	2	182	170	191
5.0P	7	4	187	166	203
5.0P	7	6	192	163	214
5.0P	7	8	197	159	226
5.0P	7	10	203	155	238
5.0P	7	12	209	150	250
5.0P	7	14	215	145	255
5.0P	8	2	208	196	217
5.0P	8	4	213	193	230
5.0P	8	6	218	189	243
5.0P	8	8	225	185	255
5.0P	8	10	231	180	255
5.0P	9	2	235	223	245
5.0P	9	4	241	219	255
7.5P	1	2	39	23	42
7.5P	1	4	44	19	49
7.5P	1	6	49	12	56
7.5P	1	8	55	3	62
7.5P	1	10	60	0	69
7.5P	1	12	65	0	76
7.5P	1	14	71	0	83
7.5P	1	16	77	0	90
7.5P	1	18	81	0	95
7.5P	1	20	87	0	102
7.5P	2	2	59	44	62
7.5P	2	4	65	40	70
7.5P	2	6	71	35	77
7.5P	2	8	78	28	86
7.5P	2	10	84	17	93
7.5P	2	12	90	0	101
7.5P	2	14	96	0	108
7.5P	2	16	102	0	116
7.5P	2	18	108	0	123
7.5P	2	20	114	0	130
7.5P	2	22	121	0	139
7.5P	2	24	128	0	146

7.5P	3	2	84	66	87
7.5P	3	4	91	62	96
7.5P	3	6	98	57	104
7.5P	3	8	105	50	113
7.5P	3	10	111	43	121
7.5P	3	12	118	32	129
7.5P	3	14	125	14	137
7.5P	3	16	131	0	144
7.5P	3	18	137	0	152
7.5P	3	20	143	0	158
7.5P	3	22	149	0	166
7.5P	3	24	156	0	174
7.5P	3	26	163	0	182
7.5P	3	28	171	0	191
7.5P	3	30	178	0	199
7.5P	4	2	108	91	111
7.5P	4	4	115	87	120
7.5P	4	6	122	82	128
7.5P	4	8	129	77	137
7.5P	4	10	136	71	145
7.5P	4	12	143	64	153
7.5P	4	14	150	55	161
7.5P	4	16	157	43	169
7.5P	4	18	164	25	178
7.5P	4	20	171	0	186
7.5P	4	22	177	0	193
7.5P	4	24	183	0	200
7.5P	4	26	191	0	209
7.5P	4	28	197	0	217
7.5P	4	30	205	0	226
7.5P	4	32	213	0	236
7.5P	5	2	133	117	136
7.5P	5	4	141	113	145
7.5P	5	6	148	108	154
7.5P	5	8	156	103	163
7.5P	5	10	163	98	171
7.5P	5	12	170	91	180
7.5P	5	14	178	84	188
7.5P	5	16	185	76	196

7.5P	5	18	192	66	204
7.5P	5	20	199	53	213
7.5P	5	22	207	30	222
7.5P	5	24	213	0	229
7.5P	5	26	220	0	238
7.5P	5	28	228	0	246
7.5P	5	30	236	0	255
7.5P	6	2	159	143	162
7.5P	6	4	167	139	171
7.5P	6	6	175	134	180
7.5P	6	8	183	129	189
7.5P	6	10	191	124	198
7.5P	6	12	198	118	206
7.5P	6	14	206	112	215
7.5P	6	16	213	104	224
7.5P	6	18	221	96	232
7.5P	6	20	230	83	243
7.5P	6	22	238	69	252
7.5P	6	24	246	50	255
7.5P	7	2	185	169	188
7.5P	7	4	193	165	197
7.5P	7	6	202	160	206
7.5P	7	8	210	155	216
7.5P	7	10	219	150	225
7.5P	7	12	227	144	234
7.5P	7	14	235	137	244
7.5P	7	16	244	130	253
7.5P	7	18	252	121	255
7.5P	8	2	211	195	215
7.5P	8	4	221	191	225
7.5P	8	6	229	187	234
7.5P	8	8	240	181	245
7.5P	8	10	249	175	255
7.5P	8	12	255	169	255
7.5P	9	2	238	222	243
7.5P	9	4	250	217	254
7.5P	9	6	255	212	255
10.0P	1	2	40	23	41
10.0P	1	4	46	18	47

10.0P	1	6	52	10	54
10.0P	1	8	57	0	60
10.0P	1	10	63	0	66
10.0P	1	12	68	0	72
10.0P	1	14	75	0	79
10.0P	1	16	80	0	85
10.0P	1	18	85	0	91
10.0P	2	2	61	44	61
10.0P	2	4	68	39	67
10.0P	2	6	75	34	74
10.0P	2	8	82	25	81
10.0P	2	10	89	12	88
10.0P	2	12	96	0	95
10.0P	2	14	103	0	102
10.0P	2	16	109	0	109
10.0P	2	18	116	0	116
10.0P	2	20	122	0	122
10.0P	2	22	129	0	130
10.0P	3	2	86	66	85
10.0P	3	4	95	61	92
10.0P	3	6	102	55	99
10.0P	3	8	111	47	107
10.0P	3	10	118	38	114
10.0P	3	12	126	23	121
10.0P	3	14	133	0	128
10.0P	3	16	140	0	134
10.0P	3	18	147	0	142
10.0P	3	20	154	0	148
10.0P	3	22	161	0	154
10.0P	3	24	168	0	162
10.0P	3	26	175	0	168
10.0P	4	2	110	91	109
10.0P	4	4	119	86	116
10.0P	4	6	128	80	123
10.0P	4	8	136	74	130
10.0P	4	10	144	68	136
10.0P	4	12	152	59	143
10.0P	4	14	160	48	150
10.0P	4	16	167	34	156

10.0P	4	18	176	0	164
10.0P	4	20	185	0	172
10.0P	4	22	191	0	178
10.0P	4	24	198	0	184
10.0P	4	26	206	0	192
10.0P	4	28	213	0	198
10.0P	4	30	223	0	208
10.0P	5	2	134	117	134
10.0P	5	4	144	112	141
10.0P	5	6	154	106	148
10.0P	5	8	164	100	155
10.0P	5	10	172	94	162
10.0P	5	12	181	87	169
10.0P	5	14	189	78	176
10.0P	5	16	197	68	182
10.0P	5	18	205	55	189
10.0P	5	20	214	34	196
10.0P	5	22	221	0	203
10.0P	5	24	229	0	210
10.0P	5	26	238	0	217
10.0P	5	28	246	0	224
10.0P	5	30	254	0	232
10.0P	6	2	161	142	160
10.0P	6	4	170	138	167
10.0P	6	6	181	132	174
10.0P	6	8	190	127	181
10.0P	6	10	200	121	188
10.0P	6	12	208	114	195
10.0P	6	14	218	106	202
10.0P	6	16	227	97	209
10.0P	6	18	235	88	216
10.0P	6	20	244	73	224
10.0P	6	22	253	55	232
10.0P	6	24	255	24	239
10.0P	6	26	255	0	247
10.0P	7	2	187	168	187
10.0P	7	4	197	164	193
10.0P	7	6	208	159	200
10.0P	7	8	219	153	208

10.0P	7	10	229	147	215
10.0P	7	12	238	140	222
10.0P	7	14	247	133	230
10.0P	7	16	255	123	238
10.0P	7	18	255	113	246
10.0P	7	20	255	101	253
10.0P	7	22	255	85	255
10.0P	8	2	212	195	213
10.0P	8	4	224	190	220
10.0P	8	6	236	185	228
10.0P	8	8	248	178	236
10.0P	8	10	255	172	244
10.0P	8	12	255	165	252
10.0P	8	14	255	156	255
10.0P	9	2	240	222	241
10.0P	9	4	254	216	249
10.0P	9	6	255	210	255
2.5RP	1	2	41	22	39
2.5RP	1	4	48	17	45
2.5RP	1	6	54	9	51
2.5RP	1	8	60	0	57
2.5RP	1	10	66	0	62
2.5RP	1	12	72	0	68
2.5RP	1	14	78	0	74
2.5RP	1	16	84	0	80
2.5RP	2	2	63	43	58
2.5RP	2	4	71	38	63
2.5RP	2	6	78	32	68
2.5RP	2	8	87	22	74
2.5RP	2	10	95	4	80
2.5RP	2	12	102	0	86
2.5RP	2	14	110	0	92
2.5RP	2	16	118	0	98
2.5RP	2	18	125	0	104
2.5RP	2	20	132	0	109
2.5RP	3	2	88	65	82
2.5RP	3	4	98	59	87
2.5RP	3	6	108	53	92
2.5RP	3	8	117	44	97

2.5RP	3	10	126	32	103
2.5RP	3	12	134	12	108
2.5RP	3	14	143	0	113
2.5RP	3	16	151	0	119
2.5RP	3	18	159	0	124
2.5RP	3	20	166	0	129
2.5RP	3	22	175	0	135
2.5RP	4	2	112	90	106
2.5RP	4	4	123	85	111
2.5RP	4	6	133	79	116
2.5RP	4	8	143	72	121
2.5RP	4	10	152	64	126
2.5RP	4	12	161	54	131
2.5RP	4	14	169	40	136
2.5RP	4	16	178	18	141
2.5RP	4	18	187	0	147
2.5RP	4	20	197	0	153
2.5RP	4	22	204	0	157
2.5RP	4	24	212	0	163
2.5RP	4	26	220	0	167
2.5RP	5	2	136	116	132
2.5RP	5	4	148	111	136
2.5RP	5	6	159	105	141
2.5RP	5	8	171	98	147
2.5RP	5	10	180	91	151
2.5RP	5	12	190	82	156
2.5RP	5	14	200	72	161
2.5RP	5	16	208	60	166
2.5RP	5	18	217	42	171
2.5RP	5	20	226	2	176
2.5RP	5	22	235	0	182
2.5RP	5	24	243	0	186
2.5RP	5	26	252	0	192
2.5RP	6	2	163	142	158
2.5RP	6	4	174	137	162
2.5RP	6	6	186	131	167
2.5RP	6	8	197	125	172
2.5RP	6	10	207	119	176
2.5RP	6	12	218	111	181

2.5RP	6	14	228	101	186
2.5RP	6	16	239	91	192
2.5RP	6	18	248	79	197
2.5RP	6	20	255	60	202
2.5RP	6	22	255	38	207
2.5RP	6	24	255	0	213
2.5RP	7	2	188	168	185
2.5RP	7	4	201	163	189
2.5RP	7	6	214	157	194
2.5RP	7	8	226	151	198
2.5RP	7	10	237	144	203
2.5RP	7	12	248	136	208
2.5RP	7	14	255	128	213
2.5RP	7	16	255	117	219
2.5RP	7	18	255	104	225
2.5RP	7	20	255	88	231
2.5RP	8	2	214	195	211
2.5RP	8	4	228	189	216
2.5RP	8	6	242	183	220
2.5RP	8	8	255	176	225
2.5RP	8	10	255	169	230
2.5RP	8	12	255	161	236
2.5RP	8	14	255	151	241
2.5RP	9	2	241	222	239
2.5RP	9	4	255	216	244
2.5RP	9	6	255	209	249
5.0RP	1	2	43	22	37
5.0RP	1	4	50	16	42
5.0RP	1	6	57	7	47
5.0RP	1	8	63	0	52
5.0RP	1	10	70	0	57
5.0RP	1	12	76	0	62
5.0RP	1	14	82	0	67
5.0RP	2	2	64	43	56
5.0RP	2	4	73	38	60
5.0RP	2	6	81	31	63
5.0RP	2	8	91	19	68
5.0RP	2	10	99	0	73
5.0RP	2	12	107	0	77

5.0RP	2	14	115	0	82
5.0RP	2	16	124	0	87
5.0RP	2	18	131	0	91
5.0RP	3	2	89	65	78
5.0RP	3	4	101	59	81
5.0RP	3	6	112	51	84
5.0RP	3	8	122	42	88
5.0RP	3	10	131	28	92
5.0RP	3	12	140	1	95
5.0RP	3	14	149	0	100
5.0RP	3	16	158	0	104
5.0RP	3	18	166	0	108
5.0RP	3	20	174	0	112
5.0RP	4	2	114	90	103
5.0RP	4	4	126	84	105
5.0RP	4	6	138	77	108
5.0RP	4	8	149	69	110
5.0RP	4	10	158	61	113
5.0RP	4	12	169	49	116
5.0RP	4	14	178	32	119
5.0RP	4	16	187	0	122
5.0RP	4	18	197	0	126
5.0RP	4	20	207	0	130
5.0RP	4	22	215	0	133
5.0RP	5	2	138	116	129
5.0RP	5	4	151	110	130
5.0RP	5	6	164	104	132
5.0RP	5	8	177	96	135
5.0RP	5	10	188	88	137
5.0RP	5	12	199	78	139
5.0RP	5	14	209	66	142
5.0RP	5	16	219	50	144
5.0RP	5	18	229	25	147
5.0RP	5	20	239	0	150
5.0RP	5	22	248	0	153
5.0RP	5	24	255	0	156
5.0RP	6	2	164	142	155
5.0RP	6	4	177	136	157
5.0RP	6	6	191	130	158

5.0RP	6	8	203	123	160
5.0RP	6	10	214	116	162
5.0RP	6	12	226	108	164
5.0RP	6	14	238	97	167
5.0RP	6	16	249	85	169
5.0RP	6	18	255	70	172
5.0RP	6	20	255	44	175
5.0RP	6	22	255	0	177
5.0RP	7	2	190	168	182
5.0RP	7	4	204	163	183
5.0RP	7	6	219	156	185
5.0RP	7	8	232	149	186
5.0RP	7	10	244	142	188
5.0RP	7	12	255	133	190
5.0RP	7	14	255	123	192
5.0RP	7	16	255	111	195
5.0RP	7	18	255	97	198
5.0RP	8	2	215	195	209
5.0RP	8	4	231	189	210
5.0RP	8	6	247	182	212
5.0RP	8	8	255	174	213
5.0RP	8	10	255	167	215
5.0RP	8	12	255	157	217
5.0RP	9	2	243	221	238
5.0RP	9	4	255	215	238
5.0RP	9	6	255	207	240
7.5RP	1	2	44	22	36
7.5RP	1	4	52	15	40
7.5RP	1	6	59	5	43
7.5RP	1	8	66	0	47
7.5RP	1	10	73	0	52
7.5RP	1	12	80	0	56
7.5RP	2	2	65	43	54
7.5RP	2	4	75	37	56
7.5RP	2	6	84	30	59
7.5RP	2	8	93	17	62
7.5RP	2	10	102	0	66
7.5RP	2	12	111	0	69
7.5RP	2	14	119	0	73

7.5RP	2	16	127	0	77
7.5RP	3	2	90	65	76
7.5RP	3	4	103	58	77
7.5RP	3	6	114	50	78
7.5RP	3	8	125	40	79
7.5RP	3	10	135	25	81
7.5RP	3	12	145	0	83
7.5RP	3	14	155	0	85
7.5RP	3	16	164	0	88
7.5RP	3	18	173	0	91
7.5RP	4	2	115	90	101
7.5RP	4	4	128	84	100
7.5RP	4	6	141	77	100
7.5RP	4	8	153	68	101
7.5RP	4	10	162	59	102
7.5RP	4	12	173	46	103
7.5RP	4	14	183	26	104
7.5RP	4	16	193	0	106
7.5RP	4	18	203	0	108
7.5RP	4	20	214	0	110
7.5RP	5	2	139	116	127
7.5RP	5	4	154	110	126
7.5RP	5	6	167	103	126
7.5RP	5	8	181	95	126
7.5RP	5	10	191	87	126
7.5RP	5	12	204	75	126
7.5RP	5	14	214	64	127
7.5RP	5	16	224	46	128
7.5RP	5	18	234	12	129
7.5RP	5	20	245	0	131
7.5RP	5	22	254	0	132
7.5RP	6	2	165	142	153
7.5RP	6	4	179	136	152
7.5RP	6	6	195	129	152
7.5RP	6	8	206	123	151
7.5RP	6	10	219	115	151
7.5RP	6	12	231	106	151
7.5RP	6	14	243	95	152
7.5RP	6	16	255	81	153

7.5RP	6	18	255	66	153
7.5RP	6	20	255	37	155
7.5RP	7	2	191	168	180
7.5RP	7	4	206	162	179
7.5RP	7	6	221	155	178
7.5RP	7	8	235	148	178
7.5RP	7	10	248	141	177
7.5RP	7	12	255	131	177
7.5RP	7	14	255	121	177
7.5RP	7	16	255	108	178
7.5RP	8	2	216	194	208
7.5RP	8	4	233	188	206
7.5RP	8	6	250	181	205
7.5RP	8	8	255	174	204
7.5RP	8	10	255	166	204
7.5RP	8	12	255	155	203
7.5RP	9	2	244	221	236
7.5RP	9	4	255	214	234
7.5RP	9	6	255	207	232
10.0RP	1	2	44	21	34
10.0RP	1	4	53	14	36
10.0RP	1	6	60	4	39
10.0RP	1	8	68	0	43
10.0RP	1	10	75	0	46
10.0RP	1	12	83	0	50
10.0RP	2	2	66	43	52
10.0RP	2	4	76	37	53
10.0RP	2	6	86	29	54
10.0RP	2	8	95	15	56
10.0RP	2	10	105	0	58
10.0RP	2	12	114	0	61
10.0RP	2	14	123	0	63
10.0RP	3	2	91	65	73
10.0RP	3	4	105	58	72
10.0RP	3	6	116	50	70
10.0RP	3	8	128	39	70
10.0RP	3	10	139	22	69
10.0RP	3	12	149	0	69
10.0RP	3	14	158	0	69

10.0RP	3	16	169	0	70
10.0RP	4	2	116	90	98
10.0RP	4	4	130	83	96
10.0RP	4	6	143	76	94
10.0RP	4	8	155	68	92
10.0RP	4	10	165	58	91
10.0RP	4	12	177	43	90
10.0RP	4	14	187	20	89
10.0RP	4	16	197	0	89
10.0RP	4	18	208	0	89
10.0RP	4	20	218	0	89
10.0RP	5	2	140	116	125
10.0RP	5	4	155	109	122
10.0RP	5	6	170	103	119
10.0RP	5	8	183	95	117
10.0RP	5	10	195	86	115
10.0RP	5	12	207	74	113
10.0RP	5	14	217	62	112
10.0RP	5	16	229	42	110
10.0RP	5	18	239	0	109
10.0RP	5	20	250	0	108
10.0RP	6	2	166	141	151
10.0RP	6	4	181	136	148
10.0RP	6	6	197	129	145
10.0RP	6	8	209	122	142
10.0RP	6	10	222	114	140
10.0RP	6	12	235	105	137
10.0RP	6	14	246	94	135
10.0RP	6	16	255	79	133
10.0RP	6	18	255	62	132
10.0RP	7	2	192	168	179
10.0RP	7	4	208	162	175
10.0RP	7	6	223	155	172
10.0RP	7	8	238	148	168
10.0RP	7	10	252	140	165
10.0RP	7	12	255	130	162
10.0RP	7	14	255	120	159
10.0RP	7	16	255	107	157
10.0RP	8	2	217	194	206

10.0RP	8	4	235	188	202
10.0RP	8	6	252	181	198
10.0RP	8	8	255	173	194
10.0RP	8	10	255	165	191
10.0RP	9	2	244	221	234
10.0RP	9	4	255	214	229
10.0RP	9	6	255	206	225
N	10	0	255	255	255
N	9	0	229	229	229
N	8	0	204	204	204
N	7	0	178	178	178
N	6	0	153	153	153
N	5	0	127	127	127
N	4	0	102	102	102
N	3	0	76	76	76
N	2	0	51	51	51
N	1	0	25	25	25
N	0	0	0	0	0

Appendix N : Complete Dataset of Munsell and RGB Distances

Calibration Chip	Expected Munsell"	Found Munsell	Distance in Munsell between found and expected	Distance in RGB between found and expected	Distance in Munsell between calibration chip and expected	Distance in rgb between calibration chip and expected
5.0B 7/8	5.0B 7/8	5.0B 7/8	0	0	0	0
5.0B 7/8	5.0BG 7/8	5.0BG 7/8	0	0	4.944273959	40.23679908
5.0B 7/8	5.0B 7/2	10.0BG 7/2	0.6257381261	5.385164807	6	89.02246907
5.0B 7/8	5.0PB 5/2	7.5B 5/2	0.9337818482	9.273618495	6.790542044	112.0089282
5.0B 7/8	5.0P 5/10	2.5P 5/10	1.569182586	12.84523258	10.88840497	111.4001795
5.0B 7/8	5.0RP 5/2	N 5/0	2	12.84523258	9.049230266	122.5846646
5.0B 7/8	5.0P 5/2	2.5PB 5/2	1.530734351	14.69693846	7.88108321	113.8507795
5.0B 7/8	5.0RP 5/12	2.5RP 5/12	1.883019103	18.60107524	16.47214043	168.4933233
5.0B 7/8	5.0G 7/10	7.5G 7/10	1.569182586	19.41648784	10.70314733	81.20960534
5.0B 7/8	5.0YR 5/2	2.5GY 5/2	2.089995062	19.89974874	10.19803903	142.5622671
5.0B 7/8	5.0PB 7/2	5.0BG 7/2	2.351141881	20.34698995	6.489334422	89.52094727
5.0B 7/8	5.0YR 5/8	10.0YR 5/6	2.949335252	20.73644135	16.1245155	194.8691869
5.0B 7/8	5.0GY 5/2	10.0GY 5/4	2.187029128	20.80865205	9.049230266	137.5645303
5.0B 7/8	5.0G 7/2	5.0G 7/4	2	21.56385865	7.623088126	86.37129153
5.0B 7/8	5.0G 4/2	7.5G 4/4	2.048654856	22.64950331	8.192159213	153.3264491
5.0B 7/8	5.0Y 5/6	2.5GY 5/6	2.801345545	24.37211521	13.4783416	183.6463994
5.0B 7/8	5.0PB 5/12	5.0PB 5/10	2	24.41311123	7.527866063	70.80254233
5.0B 7/8	5.0R 5/14	2.5R 5/12	2.852469029	24.53568829	21.10022735	211.5395944
5.0B 7/8	5.0BG 4/2	7.5B 4/4	2.947252162	25.25866188	7.149228017	130.6292463
5.0B 7/8	5.0RP 7/2	10.0B 7/2	2.828428077	27.82085549	8.82545004	96.02603814
5.0B 7/8	5.0GY 5/8	7.5GY 5/8	1.255346069	28.58321186	13.09787287	176.0710084
5.0B 7/8	5.0R 7/2	10.0GY 7/2	3.564026953	33.1662479	9.689608528	106.6348911
5.0B 7/8	5.0R 4/2	2.5GY 5/2	3.201793493	34.05877273	10.14339753	142.5622671
5.0B 7/8	N 5/0	7.5BG 6/2	2.236067977	34.66987165	8.246211251	93.27915094
5.0B 7/8	5.0Y 5/2	7.5GY 6/4	3.112281367	36.41428291	9.893864972	126.1031324
5.0B 7/8	5.0B 4/2	7.5BG 5/4	2.596901438	38.47076812	6.708203932	104.7329938
5.0B 7/8	5.0G 4/8	5.0G 5/8	1	40.162171	9.871468497	180.049993
5.0B 7/8	5.0BG 7/2	10.0B 6/4	3.405207951	40.5092582	6.489334422	77.32399369
5.0B 7/8	5.0P 7/8	2.5P 6/8	1.604959112	47.80167361	9.404567522	103.4359705
5.0B 7/8	5.0P 7/2	5.0PB 6/2	1.589926197	49.73932046	7.623088126	99.2068546
5.0B 7/8	5.0Y 8/2	7.5GY 7/2	1.828427645	52.81098371	9.741076125	113.0884609
5.0B 7/8	5.0BG 4/8	10.0BG 5/6	3.114254073	54.313902	5.783238278	152.816884
5.0B 7/8	5.0PB 7/8	2.5PB 6/6	2.486346781	56.00892786	4.944273959	63.98437309

5.0B 7/8	5.0YR 7/12	5.0YR 6/8	4.123105626	57.30619513	20	199.8524456
5.0B 7/8	5.0GY 8/10	7.5GY 7/10	1.860734798	57.81003373	14.64386704	156.4001279
5.0B 7/8	5.0RP 7/8	10.0RP 6/4	4.486912705	58.55766389	12.94427571	139.5886815
5.0B 7/8	5.0R 7/10	7.5R 6/6	4.29853468	60.73713856	17.13016542	167.2782114
5.0B 7/8	5.0GY 8/2	10.0GY 7/4	2.404806937	61.97580173	8.881923688	106.8924693
5.0B 7/8	N 10/0	7.5BG 8/2	2.236067977	62.11280061	8.246211251	107.4476617
5.0B 7/8	5.0YR 7/2	7.5GY 7/2	2.783617982	63.33245613	10.04987562	113.0884609
5.0B 7/8	5.0B 4/8	2.5PB 5/10	4.73697908	65.444633	3	132.921029
5.0B 7/8	5.0Y 8/12	10.0Y 7/8	5.137816351	87.555697	19.08746669	280.592231
5.0B 7/8	N 0/0	5.0PB 3/2	3.605551275	133.0751667	10.63014581	172.9768771
5.0G 7/10	5.0G 7/10	5.0G 7/10	0	0	0	0
5.0G 7/10	5.0B 7/2	7.5BG 7/2	0.9337818482	8.062257748	9.57284392	115.5724881
5.0G 7/10	5.0BG 7/8	2.5BG 7/8	1.255346069	13.60147051	5.878546268	37.41657387
5.0G 7/10	5.0PB 7/2	5.0BG 7/2	2.351141881	20.34698995	10.78706218	114.0087716
5.0G 7/10	5.0YR 5/8	10.0YR 5/6	2.949335252	20.73644135	14.74594324	161.0403676
5.0G 7/10	5.0G 7/2	7.5G 7/4	2.048654856	22.75961335	8	92.46080251
5.0G 7/10	5.0BG 7/2	5.0BG 7/4	2	23.55843798	8.464001805	92.30926281
5.0G 7/10	5.0Y 5/6	2.5GY 5/6	2.801345545	24.37211521	10.144852	135.8160521
5.0G 7/10	5.0R 5/14	2.5R 5/12	2.852469029	24.53568829	22.94613117	205.6283054
5.0G 7/10	5.0YR 7/12	7.5YR 7/10	2.637195629	25.35744467	17.83715961	206.1407286
5.0G 7/10	5.0P 7/2	2.5BG 7/2	3.410561572	30.5122926	11.67735865	113.9166362
5.0G 7/10	N 5/0	10.0G 6/2	2.236067977	31.06444913	10.19803903	94.91048414
5.0G 7/10	5.0P 7/8	7.5PB 7/4	4.793285754	33.74907406	17.13017281	140.0035714
5.0G 7/10	5.0B 7/8	7.5BG 8/8	3.86667514	33.9705755	10.70314733	88.73556221
5.0G 7/10	5.0R 4/2	7.5GY 5/2	3.554142687	36.68787266	12.05656274	106.0188662
5.0G 7/10	5.0R 7/2	5.0G 7/2	3.804226731	36.74234614	11.67735865	115.04347
5.0G 7/10	5.0P 5/2	2.5BG 6/2	3.554142687	36.78314832	11.84739233	94.1328848
5.0G 7/10	5.0R 7/10	7.5YR 7/4	7.709181315	36.82390528	19.02113365	162.9386388
5.0G 7/10	5.0BG 4/2	2.5BG 5/4	2.279690049	37.25587202	8.979940231	78.01281946
5.0G 7/10	5.0P 5/10	5.0P 6/6	4.123105626	37.42993454	19.12599084	139.2910622
5.0G 7/10	5.0YR 5/2	7.5GY 6/4	4.300820635	38.18376618	10.9709029	100.8662481
5.0G 7/10	5.0RP 5/2	2.5G 6/2	4.111143996	38.39270764	12.16552506	98.468269
5.0G 7/10	5.0B 4/2	7.5BG 5/4	2.596901438	38.47076812	10.0319161	76.12489737
5.0G 7/10	5.0G 4/8	5.0G 5/8	1	40.162171	3.605551275	120.074977
5.0G 7/10	5.0PB 5/2	2.5BG 6/4	4.300820635	40.16217126	10.9709029	70.9013399
5.0G 7/10	5.0PB 5/12	10.0B 6/8	5.137816351	40.80441153	17.94893487	81.60882305
5.0G 7/10	5.0G 4/2	5.0G 5/6	4.123105626	41.06093034	8.544003745	74.43117626
5.0G 7/10	5.0RP 5/12	7.5RP 6/10	2.820425639	41.31585652	22.09072203	189.3171941
5.0G 7/10	5.0RP 7/2	N 6/0	2.236067977	42.26109322	12	119.1847306

5.0G 7/10	5.0Y 5/2	7.5GY 6/2	1.828427645	44.06812907	9.779536835	105.8017013
5.0G 7/10	5.0GY 5/2	2.5G 6/6	4.428977943	45.11097427	8.697087246	68.0734897
5.0G 7/10	5.0GY 8/2	10.0GY 8/6	4.144230279	45.76024475	8.522870793	117.7327482
5.0G 7/10	5.0YR 7/2	10.0GY 8/4	4.472137159	48.86716689	10.83331484	133.3116649
5.0G 7/10	5.0RP 7/8	7.5YR 7/2	7.936877923	50.85272854	18	149.0436178
5.0G 7/10	5.0BG 4/8	10.0BG 5/6	3.114254073	54.313902	6.599795923	103.985576
5.0G 7/10	5.0Y 8/2	10.0GY 7/2	2.073094449	55.50675635	9.624933284	118.6128155
5.0G 7/10	5.0PB 7/8	2.5PB 6/6	2.486346781	56.00892786	14.60968316	96.7780967
5.0G 7/10	5.0GY 8/10	7.5GY 7/10	1.860734798	57.81003373	6.260721427	109.0183471
5.0G 7/10	5.0B 4/8	2.5PB 5/10	4.73697908	65.444633	11.11563596	102.5768
5.0G 7/10	N 10/0	5.0G 8/2	2.236067977	66.54321904	10.19803903	145.5918954
5.0G 7/10	5.0GY 5/8	10.0GY 6/12	5.137816351	68.70953355	6.209452973	90.04998612
5.0G 7/10	5.0Y 8/12	2.5GY 8/10	5.491671531	72.041655	13.07042632	238.482704
5.0G 7/10	N 0/0	2.5BG 4/2	4.472135955	160.7824617	12.20655562	111.1800342
5.0R 5/14	5.0G 7/10	5.0G 7/10	0	0	22.94613117	224.4482123
5.0R 5/14	5.0G 7/2	5.0G 7/2	0	0	15.78939477	154.7578754
5.0R 5/14	5.0PB 5/2	5.0PB 5/2	0	0	14.74126356	127.6048588
5.0R 5/14	5.0R 5/14	5.0R 5/14	0	0	0	0
5.0R 5/14	5.0YR 5/2	5.0YR 5/2	0	0	12.43764677	97.01030873
5.0R 5/14	5.0YR 5/8	5.0YR 5/8	0	0	8.875822706	61.47357156
5.0R 5/14	N 5/0	N 5/0	0	0	14	118.0677771
5.0R 5/14	5.0R 4/2	2.5R 4/2	0.3138311481	2	12.04159458	106.4847407
5.0R 5/14	5.0BG 7/2	7.5BG 7/2	0.3138365171	2.236067977	16.1245155	162.5546062
5.0R 5/14	5.0G 4/2	10.0G 4/2	0.6257381261	4.472135955	15.69410676	141.665098
5.0R 5/14	5.0B 4/2	10.0B 4/2	0.6257381261	5.477225575	15.69410394	139.1545903
5.0R 5/14	5.0GY 5/2	7.5GY 5/2	0.3138365171	5.744562647	14.74126842	121.2146856
5.0R 5/14	5.0PB 7/2	10.0B 7/2	0.6257381261	6.403124237	14.87631847	161.3753389
5.0R 5/14	5.0Y 5/6	7.5Y 5/6	0.9415095514	7.874007874	13.41958386	102.4353455
5.0R 5/14	5.0P 5/2	N 5/0	2	9.433981132	13.51646898	118.0677771
5.0R 5/14	5.0BG 4/2	10.0B 4/2	1.815962719	12.36931688	16.03121954	139.1545903
5.0R 5/14	5.0RP 5/12	7.5RP 5/12	1.883019103	12.80624847	8.256501151	50.04997502
5.0R 5/14	5.0RP 5/2	N 5/0	2	12.84523258	12.43764321	118.0677771
5.0R 5/14	5.0P 5/10	7.5P 5/10	1.569182586	14.17744688	14.47323972	109.5947079
5.0R 5/14	5.0B 7/2	5.0G 7/2	2.351141881	18.35755975	15.78939196	154.7578754
5.0R 5/14	5.0R 7/2	2.5GY 7/2	3.041624824	26.55183609	12.16552506	134.7033778
5.0R 5/14	5.0P 7/2	2.5G 7/2	3.889480216	32.12475681	13.66363545	150.9337603
5.0R 5/14	5.0G 4/8	5.0G 4/6	2	33.511192	21.02902633	225.39299
5.0R 5/14	5.0PB 5/12	2.5PB 5/8	4.285305197	34.2636834	21.06725202	184.5264209
5.0R 5/14	5.0B 4/8	2.5PB 4/8	4.630223623	36.578682	22.88065761	235.023403

5.0R 5/14	5.0GY 5/8	7.5GY 6/10	2.640050604	36.83748091	18.14442005	150.6784656
5.0R 5/14	5.0B 7/8	10.0BG 7/6	2.949335252	37.54996671	21.10022735	202.654879
5.0R 5/14	5.0BG 7/8	7.5BG 6/6	2.486346781	39.30648801	22.09072203	194.1571528
5.0R 5/14	5.0YR 7/12	2.5YR 6/12	2.132079019	40.42276586	8.495313006	54.8634669
5.0R 5/14	5.0Y 5/2	10.0Y 6/2	1.179639014	41.88078318	13.51647428	112.6454615
5.0R 5/14	5.0RP 7/2	N 7/0	2.236067977	44.48595284	12.79433345	153.5545506
5.0R 5/14	5.0GY 8/2	7.5GY 7/2	1.048090339	46.10856753	15.04343693	142.6253834
5.0R 5/14	5.0YR 7/2	5.0YR 7/2	1	46.19523785	12.79433692	130.7325514
5.0R 5/14	5.0P 7/8	5.0P 6/8	1	46.21688003	13.95635104	139.6173342
5.0R 5/14	5.0Y 8/2	10.0Y 7/2	1.179639014	48.2182538	13.84539913	132.4311142
5.0R 5/14	5.0PB 7/8	7.5PB 6/6	2.486346781	53.01886457	18.25429827	156.9872606
5.0R 5/14	5.0RP 7/8	7.5RP 6/6	2.486346781	54.15717866	9.098344584	95.79144012
5.0R 5/14	5.0BG 4/8	10.0BG 5/6	3.114254073	54.313902	22.02271555	228.247672
5.0R 5/14	5.0Y 8/12	10.0Y 7/10	4.092242182	54.744863	15.6579201	156.089718
5.0R 5/14	5.0R 7/10	5.0R 6/8	2.236067977	67.83067153	4.472135955	70.23531875
5.0R 5/14	N 10/0	7.5GY 8/2	2.236067977	69.28924881	14.56021978	170.5549765
5.0R 5/14	5.0GY 8/10	7.5GY 7/12	2.820425639	72.9451849	19.78699001	161.1614098
5.0R 5/14	N 0/0	10.0GY 3/4	5	109.6357606	14.86606875	166.5743077
N 0/0	N 0/0	N 0/0	0	0	0	0
N 0/0	5.0R 5/14	7.5R 4/16	3.242780926	72.01388755	14.86606875	196.2574839
N 0/0	5.0BG 4/8	7.5BG 2/4	4.559380098	77.162167	8.94427191	159.81239
N 0/0	5.0G 7/10	5.0G 5/8	2.828427125	78.84795495	12.20655562	170.3672504
N 0/0	5.0Y 5/2	5.0Y 3/2	2	80.95677859	5.385164807	121.4660446
N 0/0	5.0Y 8/2	7.5Y 4/2	6.115061197	82.13403679	9.433981132	159.0251552
N 0/0	5.0GY 5/8	7.5GY 3/8	2.361333046	87.28688332	9.433981132	89.98333179
N 0/0	N 5/0	N 3/0	2	88.33459119	5	131.6358614
N 0/0	5.0P 5/10	7.5P 3/8	3.15750965	92.31467922	11.18033989	159.7028491
N 0/0	5.0B 7/8	2.5B 5/6	3.030168364	92.87626177	10.63014581	204.0833163
N 0/0	5.0PB 5/12	5.0PB 3/8	4.472135955	94.26027795	13	150.0266643
N 0/0	5.0B 4/8	7.5BG 2/4	5.193802877	100.503731	8.94427191	182.628037
N 0/0	5.0G 4/8	10.0GY 1/4	5.303997136	101.419919	8.94427191	137.058382
N 0/0	5.0RP 5/12	10.0RP 3/10	4.443697343	105.1950569	13	156.7354459
N 0/0	5.0G 4/2	2.5BG 1/2	3.141965713	105.2710786	4.472135955	54.4242593
N 0/0	5.0YR 7/12	7.5YR 5/10	3.309803738	105.6314347	13.89244399	202.6055281
N 0/0	5.0BG 4/2	7.5BG 1/2	3.016370892	106.8971468	4.472135955	55.68662317
N 0/0	5.0B 4/2	10.0B 1/2	3.064563297	106.9252075	4.472135955	61.10646447
N 0/0	5.0R 4/2	7.5R 1/2	3.016370892	109.2382717	4.472135955	63.63175308
N 0/0	5.0BG 7/8	5.0B 4/6	5.597712366	110.1317393	10.63014581	167.0987732
N 0/0	5.0Y 5/6	7.5Y 2/4	3.686591943	112.7696768	7.810249676	79.2779919

N 0/0	5.0YR 5/2	10.0R 2/2	3.064563297	116.8845584	5.385164807	93.90420651
N 0/0	5.0Y 8/12	5.0Y 5/8	5	119.423616	14.4222051	308.954689
N 0/0	5.0RP 5/2	7.5RP 2/2	3.016370892	119.5198728	5.385164807	97.37042672
N 0/0	5.0PB 5/2	5.0PB 2/2	3	120.0916317	5.385164807	95.59288676
N 0/0	5.0P 5/2	2.5P 2/2	3.016370892	120.1082845	5.385164807	98.65089964
N 0/0	5.0BG 7/2	10.0B 4/2	3.506810602	122.3968954	7.280109889	170.020587
N 0/0	5.0B 7/2	N 4/0	3.605551275	122.8901949	7.280109889	176.6691824
N 0/0	5.0YR 5/8	2.5YR 2/8	3.252059924	123.8224535	9.433981132	98.50888285
N 0/0	5.0GY 5/2	7.5GY 2/4	3.632765712	124.3905141	5.385164807	76.11832894
N 0/0	5.0R 7/2	N 4/0	3.605551275	127.6048588	7.280109889	176.6691824
N 0/0	5.0G 7/2	2.5BG 4/2	3.141965713	128.4133949	7.280109889	160.7824617
N 0/0	5.0PB 7/2	7.5PB 4/2	3.016370892	129.9576854	7.280109889	174.2785127
N 0/0	5.0P 7/2	N 4/0	3.605551275	130.7593209	7.280109889	176.6691824
N 0/0	5.0RP 7/2	7.5RP 4/2	3.016370892	132.2459829	7.280109889	173.8505105
N 0/0	5.0YR 7/2	2.5Y 4/2	3.141965713	138.1122732	7.280109889	161.5518493
N 0/0	5.0GY 8/10	7.5GY 5/10	3.385606886	140.5916071	12.80624847	157.3372175
N 0/0	5.0GY 8/2	7.5GY 5/4	3.632765712	144.8378404	8.246211251	188.6451696
N 0/0	5.0R 7/10	7.5R 4/8	3.869091261	145.7978052	12.20655562	181.6975509
N 0/0	5.0P 7/8	5.0P 4/4	5	149.1241094	10.63014581	183.5429105
N 0/0	5.0PB 7/8	10.0B 4/4	5.303997136	149.268215	10.63014581	172.580416
N 0/0	5.0RP 7/8	5.0R 4/4	6.101050827	151.8584868	10.63014581	176.3320731
N 0/0	N 10/0	5.0GY 6/2	3.605551275	157.0413958	9	241.5056935
N 10/0	5.0GY 8/10	5.0GY 8/10	0	0	10.04987562	166.7633053
N 10/0	N 10/0	N 10/0	0	0	0	0
N 10/0	5.0RP 7/8	2.5RP 7/8	1.255346069	13.56465997	8.246211251	85.63293759
N 10/0	5.0Y 8/2	2.5YR 8/2	1.530734351	18.33030278	2.236067977	54.3783045
N 10/0	5.0GY 8/2	N 8/0	2	32.75667871	2.236067977	43.30127019
N 10/0	5.0P 7/8	5.0P 8/4	4.123105626	37.49666652	8.246211251	40.06245125
N 10/0	5.0YR 7/12	10.0R 8/14	4.630877293	38.37968212	12.16552506	153.8603263
N 10/0	5.0P 5/2	5.0P 6/2	1	43.88621651	4.472135955	134.792433
N 10/0	5.0P 7/2	10.0PB 8/2	1.179639014	44.68780594	2.828427125	45.84757355
N 10/0	5.0RP 7/2	2.5RP 8/2	1.048090339	46.41120554	2.828427125	45.82575695
N 10/0	5.0PB 7/2	7.5PB 8/2	1.048090339	48.09365863	2.828427125	46.10856753
N 10/0	5.0PB 7/8	2.5PB 8/4	4.217575948	48.35286961	8.246211251	57.25382083
N 10/0	5.0R 7/2	5.0RP 8/2	1.589922215	49.2341345	2.828427125	46.17358552
N 10/0	5.0G 7/10	5.0G 8/10	1	49.40647731	10.19803903	167.182535
N 10/0	5.0YR 7/2	7.5R 8/2	1.368191704	51.4781507	2.828427125	50.88221693
N 10/0	5.0BG 4/2	10.0B 5/2	2.073094449	52.70673581	5.385164807	184.9621583
N 10/0	5.0RP 5/2	7.5P 6/4	2.596901438	54.49770637	4.472135955	128.9689885

N 10/0	5.0B 7/2	7.5PB 8/2	1.828427645	55.47071299	2.828427125	46.10856753
N 10/0	5.0PB 5/2	2.5P 6/4	2.596901438	55.73149917	4.472135955	129.0465032
N 10/0	5.0R 5/14	2.5R 6/14	2.413712801	56.08921465	14.56021978	174.3961009
N 10/0	5.0BG 7/2	10.0B 8/2	2.073094449	57.5847202	2.828427125	51.77837386
N 10/0	5.0G 7/2	5.0B 8/2	2.554969304	57.90509477	2.828427125	56.04462508
N 10/0	5.0Y 8/12	7.5Y 8/10	3.968179189	59.573484	12.04159458	231.400086
N 10/0	5.0Y 5/6	5.0Y 6/4	2.236067977	60.24118193	7.211102551	169.6231116
N 10/0	5.0YR 5/8	10.0R 6/8	2.695323958	62.07253821	8.94427191	167.3499328
N 10/0	5.0B 7/8	10.0B 8/8	2.695323958	66.31741853	8.246211251	111.5750868
N 10/0	5.0R 7/10	7.5R 8/6	4.29853468	66.73829485	10.19803903	79.0569415
N 10/0	N 5/0	2.5G 7/2	2.828427125	68.56383887	4	111.5930105
N 10/0	5.0RP 5/12	7.5RP 7/16	4.972693561	76.22991539	12.64911064	131.8408131
N 10/0	5.0GY 5/8	7.5GY 7/10	3.15750965	77.62087348	8.94427191	190.5098423
N 10/0	5.0BG 7/8	10.0BG 8/6	3.114254073	78.39642849	8.246211251	109.1512712
N 10/0	5.0YR 5/2	5.0Y 7/2	2.351141278	80.46117076	4.472135955	111.0360302
N 10/0	5.0R 4/2	5.0YR 6/2	2.351141278	81.85963596	5.385164807	146.3625635
N 10/0	5.0B 4/2	7.5BG 6/2	2.207249089	82.19489035	5.385164807	150.4061169
N 10/0	5.0G 4/2	5.0G 6/2	2	84.95292814	5.385164807	154.5186073
N 10/0	5.0P 5/10	7.5P 7/8	3.15750965	86.42337647	10.77032961	78.16009212
N 10/0	5.0GY 5/2	7.5GY 7/2	2.024473601	87.89766777	4.472135955	112.25863
N 10/0	5.0B 4/8	5.0PB 5/10	5.962994736	88.932559	9.433981132	271.819793
N 10/0	5.0PB 5/12	2.5PB 7/8	4.729042253	92.59049627	12.64911064	117.2049487
N 10/0	5.0Y 5/2	5.0YR 7/2	2.351141278	97.39609848	4.472135955	101.6415269
N 10/0	5.0G 4/8	5.0BG 5/4	5.405823017	97.637083	9.433981132	297.707911
N 10/0	5.0BG 4/8	2.5B 6/6	4.296903825	110.281458	9.433981132	281.700195
N 10/0	N 0/0	10.0PB 4/2	4.472135955	179.6997496	9	218.2544387
N 5/0	5.0G 4/2	5.0G 4/2	0	0	2.236067977	63.22183167
N 5/0	5.0GY 5/8	5.0GY 5/8	0	0	8	100.7422454
N 5/0	5.0YR 5/2	5.0YR 5/2	0	0	2	25.35744467
N 5/0	5.0YR 5/8	5.0YR 5/8	0	0	8	89.9499861
N 5/0	N 5/0	N 5/0	0	0	0	0
N 5/0	5.0R 4/2	7.5R 4/2	0.3138365171	1	2.236067977	53.27288241
N 5/0	5.0GY 5/2	2.5GY 5/2	0.3138365171	3.605551275	2	28.05352028
N 5/0	5.0BG 4/2	10.0G 4/2	0.6257381261	4.472135955	2.236067977	62.52199613
N 5/0	5.0Y 5/6	7.5Y 5/6	0.9415095514	7.874007874	6	82.39538822
N 5/0	5.0P 5/2	5.0RP 5/2	1.23606849	11.5758369	2	12.84523258
N 5/0	5.0B 4/2	5.0PB 4/2	1.23606849	12.08304597	2.236067977	49.51767361
N 5/0	5.0RP 5/2	10.0R 5/2	1.81595792	12.64911064	2	21.84032967
N 5/0	5.0YR 7/2	5.0Y 7/2	1.23606849	13.78404875	2.828427125	73.43704787

N 5/0	5.0R 7/2	7.5YR 7/2	1.530734351	14.03566885	2.828427125	79.80601481
N 5/0	5.0RP 5/12	5.0RP 5/14	2	14.28285686	12	96.02603814
N 5/0	5.0BG 7/2	2.5G 7/2	1.530734351	15.23154621	2.828427125	68.56383887
N 5/0	5.0P 5/10	7.5P 5/8	2.443331167	16.52271164	10	44.73253849
N 5/0	5.0RP 7/2	5.0YR 7/2	2.351137524	18.46618531	2.828427125	81.60269603
N 5/0	5.0R 5/14	5.0R 5/16	2	19.54482029	14	136.4038123
N 5/0	5.0B 7/2	N 7/0	2	22.58317958	2.828427125	88.33459119
N 5/0	5.0PB 5/2	2.5G 5/2	3.410561572	24.10394159	2	25.01999201
N 5/0	5.0PB 5/12	5.0PB 5/10	2	24.41311123	12	82.42572414
N 5/0	5.0G 7/2	N 6/0	2.236067977	29.563491	2.828427125	45.033321
N 5/0	5.0G 7/10	7.5G 6/8	2.640050604	32.81767816	10.19803903	89.05054744
N 5/0	5.0G 4/8	5.0G 4/6	2	33.511192	8.062257748	136.59429
N 5/0	5.0B 7/8	2.5B 7/6	2.276383165	35.56683849	8.246211251	96.38464608
N 5/0	5.0Y 5/2	10.0Y 6/4	2.404806937	37.02701716	2	45.32107677
N 5/0	5.0BG 7/8	10.0BG 6/6	3.114254073	37.13488926	8.246211251	79.06326581
N 5/0	5.0YR 7/12	2.5YR 6/12	2.132079019	40.42276586	12.16552506	125.6741819
N 5/0	5.0PB 7/2	N 6/0	2.236067977	40.75536774	2.828427125	45.033321
N 5/0	5.0P 7/2	N 6/0	2.236067977	43	2.828427125	45.033321
N 5/0	5.0B 4/8	5.0PB 4/10	5.878546268	45.033321	8.062257748	130.268953
N 5/0	5.0GY 8/2	5.0GY 7/2	1	45.09988914	3.605551275	69.20982589
N 5/0	5.0Y 8/2	5.0Y 7/2	1	46.23851209	3.605551275	73.43704787
N 5/0	5.0GY 8/10	5.0GY 7/10	1	47.92702787	10.44030651	105.8914539
N 5/0	5.0P 7/8	7.5P 6/8	1.604959112	49.2036584	8.246211251	75.07329752
N 5/0	5.0Y 8/12	5.0Y 7/10	2.236067977	49.699095	12.36931688	182.619824
N 5/0	5.0BG 4/8	10.0BG 4/4	4.374058255	52.933921	8.062257748	128.541822
N 5/0	5.0RP 7/8	2.5R 6/6	3.932350758	61.16371473	8.246211251	67.30527468
N 5/0	5.0PB 7/8	7.5PB 6/4	4.217575948	65.36818798	8.246211251	51.70106382
N 5/0	5.0R 7/10	5.0R 6/8	2.236067977	67.83067153	10.19803903	81.41867108
N 5/0	N 10/0	N 7/0	2	88.33459119	4	88.33459119
N 5/0	N 0/0	2.5P 2/2	2.828427125	98.65089964	5	122.225202