Machine Learning Algorithms to Correct Images to Help the Color Blind: Draft II

Jimmy Hickey

April 15, 2018

Abstract

Many people suffer from color deficient vision; though they learn to cope, they still have many issues distinguishing between colors. In this research, machine learning algorithms are combinted to help to eliminate some of the problems faced by these individuals. As a prototype, specialized data with intentional color blind issues is generated, transformed, and used to train the system. The colors in these images are then segmented using k-means clustering. The image is then analyzed and corrected if neighboring clusters are found to be conflicting. After correcting, the image is more color blind friendly. Omitted from the prototype is a supervised neural network that will be trained to predict these clusters, allowing this process to be applied on any image.

1   Introduction

Color blindness is a deficiency in a person's color vision. A color blind person will often confuse different colors that a person with normal vision would not have trouble distinguishing. The most common type of color blindness is red-green, followed by blue-yellow, and the rare full lack of color vision. Red-green color blindness affects "as many as 8 percent of men and 0.5 percent of women with Northern European ancestry." [3] The effects of color blindness range from minor disruptions such as choosing a mismatched outfit to more major difficulties such as discerning between the colors of traffic lights. There is no cure for color blindness, however there exist some methods to alleviate symptoms.

One popular solution to color blindness is specialized glasses. [1] The company enchroma sells a variety of these glasses. Color vision starts in the cones of the eye, of which people have three varieties: green, red, and blue. These detect and distinguish between different wavelengths of light; however, in the eye of a color blind person, this discrimination is not always made properly. The enchroma glasses filter out the overlapping wavelengths, making distinctions (particularly between red and green) clearer. These glasses range from $349 to $429 without prescriptions and are not "intended to help pass color blindness tests for occupational purposes."[2] Additionally, they (enchroma brand) are not useful to people with blue-yellow color blindness. A more permanent solution is currently being tested as well.

There is experimental gene therapy that has produced promising results in animals.[5] In one study squirrel monkeys, which are naturally red-green color blind, were made able to perceive differences in the two colors after treatment. These animals were monitored and "retained their new tricolor sensory capacity for more than two years." [5]. Additionally, there have been no detected harmful side effects from the treatment; four more animals have been successfully treated since. These auspicious trials suggest that there may be a permanent fix to color blindness in the near future; however, gene therapy is often prohibitively expensive, costings hundreds of thousands of dollars per patient. A short-term, cheap solution to some of the issues faced by the color blind is needed.

The burgeoning field of computer vision lends itself well to this problem. Much work has been done in the area of feature recognition. Projects like Google's Deep Dream and work in robotic vision are some of the wide applications of computer vision. Computers' abilities to analyze images is greatly enhancing. These methods can be used as the small, cheap fix to some color blind issues. A first step would be to examine images.

Instead of recognizing complex features, a machine needs only to examine the colors in an image. With appropriate knowledge of the problem domain, it should be able to diagnose whether there is areas of the image that may cause issues for the color blind. For example, given a set of color blind tests, it would be able to tell which slides a color blind person would have trouble with. This process can then be incrementally furthered. The system can

1. mark all problem areas.

2. fix the problem areas without losing too much of the context of the image.

3. perform the preceding actions on a video.

4. perform the preceding actions on a live video feed.

For the scope of this project, the first and second object will be addressed. Unlike the glasses, the machine is not dealing with the light itself, but how it is displayed. Thus, this could be easily expanded to support all of the common types of color blindness.

The overall goal of this research is to create a prototype to prove that a system can be trained to identify and resolve areas of potential color blind confusion in images.

2   Methods

2.1   Data Creation

2.1.1   Image Creation

Initially, it was planned to use standard Ishihara Color Test plates, such as those used by eye doctors, as the data for this experiment. See figure 1 for an example. However, this afforded little control of the data. The learning process would rely on only the Ishihara samples publicly available online.
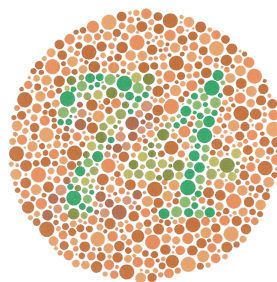
Figure 1: A red-green Ishihara color blind plate.

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

Instead of this, new data was generated for the sake of this experiment. This allows for a wider range of data to be created and manipulated to help the system learn more color combinations. The data created using Adobe Illustrator. A background was chosen and then a foreground image was drawn. The colors of the background and foreground were tweaked until a red-green color blind person started having trouble distinguishing between the two. Figure 2 shows pink sailboat on a gray background.

Figure 2: A sample image created in Adobe Photoshop.

For a color blind person (particularly red-green color blind), it may be very difficult to discriminate between the background and foreground (at least for myself, a person with red-green color-blindness). This process will be continued to create a plethora of color combinations such as blue-purple, brown-red, orange-yellow, red-green, etc. These color combinations will then be altered and repeated. For example, dark blue and dark purple will be used in one image a lighter tint of each color will be used in another. Issues with the devised data creation system will be explored in the Future Improvements section.

2.1.2   From Image to Data

The image data needs to be represented in a more quantitative format while still maintaining the information stored in the picture. Namely, the information to retain is the color and location of each pixel. Thus, the x and y position will be stored as well as the red (R), green (G), and blue (B) values for each pixel. Table 1 shows a few rows of the data table representing the sailboat image from figure 2.

Table 1: Data representation of sail boat picture.

x	y	R	G	B
0	0	205	205	205
1	0	205	205	205
2	0	205	205	205
⋮				
57	116	233	184	217
⋮				
300	300	205	205	205

With the primitive creation method used to form this data, some editing needs to be done yet. One may notice that all of the background pixels will have the exact same red, green, and blue values. At that, the foreground also only include a narrow range of colors. This is not an issue immediately, but looking forward to the color detection and generalization methods that will be used, this will cause problems. These issues and solutions to them will be expounded upon in future sections.

2.2   Color Detection

2.2.1   k-means

An unsupervised learning algorithm for pattern recognition will be used to segment the image. Differences in color are detected in each image. K-means clustering [4] is being used to find distinct trends in the pixel data. Since each image is created to depict a combination of two colors, the k-means algorithm is instructed to dichotomize the data.

The Python SKLearn package was used to implement k-means clustering. The algorithm works as follows:

1. Determine how many clusters the data will be separated into. For this project, two clusters will be used.

2. Randomly select a mean for each cluster.

3. Classify each piece of data according to each mean.

4. Use these new classifications to recompute the mean of each cluster.

5. Repeat 3-4 until the clusters converge (that is, they do not change more than a threshold between iterations).

Using this algorithm on the sail boat image shown in figure 2 successfully bisects the data into clusters. The clusters have been changed to black and white to clearly illustrate the difference:

Figure 3: Image after segmentation by k-means algorithm.

It is expected that the system will identify issues with the red and green more so than the blue since it will be given data for a red-green color blind person. Though k-means is sensitive to outliers, since multiple images will be created for each color combination, local maxima convergence will, hopefully, be avoided.

2.3   Generalization

Identifying the clusters in images is but the first step in the detection process. Next, we want to generalize these clusters. The first step is to generalize within each image and then across multiple images.

2.3.1   Adding Background Noise

As previously mentioned, each image's foreground and background are very distinct, but in fact only have a few different RGB in each of them. For example, the background cluster of the sailboard image will contain only (205, 205, 205) values in the RGB columns. If a single pixel's colors were changed, the k-means algorithm may have a hard time classifying it. However, if the colors were less uniform, the k-means algorithm could train itself on more varied, yet still distinct data. Thus random normal noise is added to each pixel's B, G, and R values. The parameters for this noise were arbitrarily chosen to be a

mean of 0 and a standard deviation of 3. This these changes to the data, the image that the algorithm trains on looks as follows:
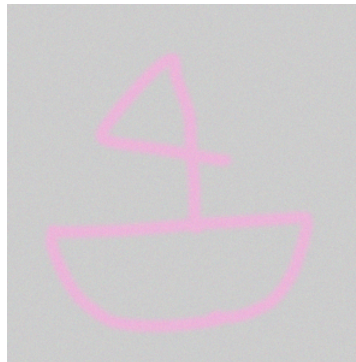

Figure 4: Normal noise added to each pixel of an image.

After running the k-means algorithm on this new data, a few cases were noticeably misclassified in this example; however, overall it did about as well.
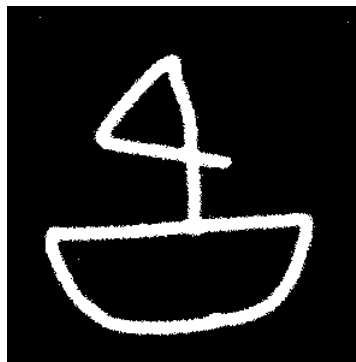

Figure 5: Clustering on data with normal noise.

2.3.2   Across Image Generalization

This idea of generalization is then brought further. Say there are two images, one with a gray background and a pink foreground and the other with a pink background and a gray foreground. In the long run, it would be desired for both of these pinks to be in recognized as the same cluster and the grays to be combined as another single cluster.

This can be achieved by stacking the data together. Only the color data of each pixel is used as input to the k-means algorithm. Thus, the data from many images can be combined together and trained on

simultaneously. This offers the benefit of controlling the amount of clusters across multiple images. Furthermore, data with different clusters can be stacked.

Data with reds and greens can be combined with data with grays and pinks and the algorithm can be configured to look for 4 clusters. For the most general case, the data from all images could be stacked, and clusters can be found and combined across the whole test data set.

2.3.3   Clustering Generalization

With our current methodology, to categorize a new image, we would need to stack its data with the data from all of the other images. This is extremely computationally intesive and inefficient. Instead, we use a supervised learning algorithm to learn how to cluster.

With the clusters already known for the training data, a neural network will be trained to learn how these categories were formed. This is particularly useful for images without a binary color scheme. If the image passed has red, blue, and pink in it, the supervised neural network will categorize all of these. This is a necessary step in making the system work for real images.

2.4   Color Correction

One simple correction method that could be employed is through the use of edge detection. Once the system is properly trained to determine problem areas in images, an edge could be detected. Tracing this area with a thin black line would offer a visible separation between the two competing colors, while trying not to lose the information stored in the image.
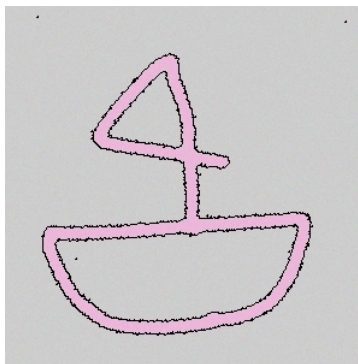
Figure 6: A thin black border is drawn between clusters.

Another possible correction procedure is to change the colors or contrast of the image itself. For example. in the sailboat data point in figure 2, the pink or the gray could be darkened. This method would leave the picture more intact as it wouldn't draw black lines, which could make the image more cumbersome to look at. However, this color changing would have to be very precise as to not completely alter the image. Changing the boat to black would assuredly make it more visible, but the image is no longer of a pink boat and a gray background. Subtly changing the color values or contrast would ensure that the image is still in tact while being more accessible. With improvements to the system such as user testing this mechanism could be fine tuned to translate an image into a user and color blind friendly picture.
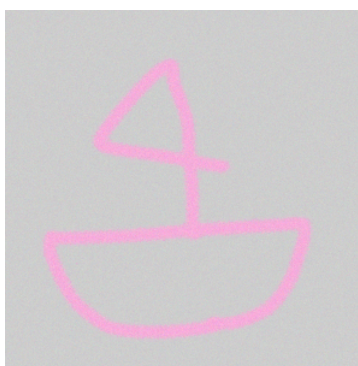

Figure 7: Cluster colors slightly adjusted.

3   Results Analysis

The output of this system will be the corrected images. For the scope of this research, wide, comprehensive results will not be provided. Only one person will evaluate whether the objects are easier to distinguish; and indeed they are. Peers were consulted as to how destructive the algorithm is, that is how much it alters the colors. A proper analysis would involve both groups of color blind and non color blind subjects. They would examine the image before and after processing and comment on the effectiveness and destructiveness.

4   Conclusion

On the whole, this research has outlined the prototype to a system for assisting people with color blindness. Images are ubiquitous on the Internet. This system provides a method for those affected to enjoy and appreciate this information as anyone else could. An ensemble of machine learning mechanisms provide color clustering; these clusters are then used to determine if there will be any potential confusion in the image and corrects accordingly. Importantly, this correction process does not destroy the image, so the message being communicated is not lost to the viewer. The process can be followed in full through the in figure 8. Further additions have been suggested throughout this paper, but there are yet more that would bring this research to greater heights.
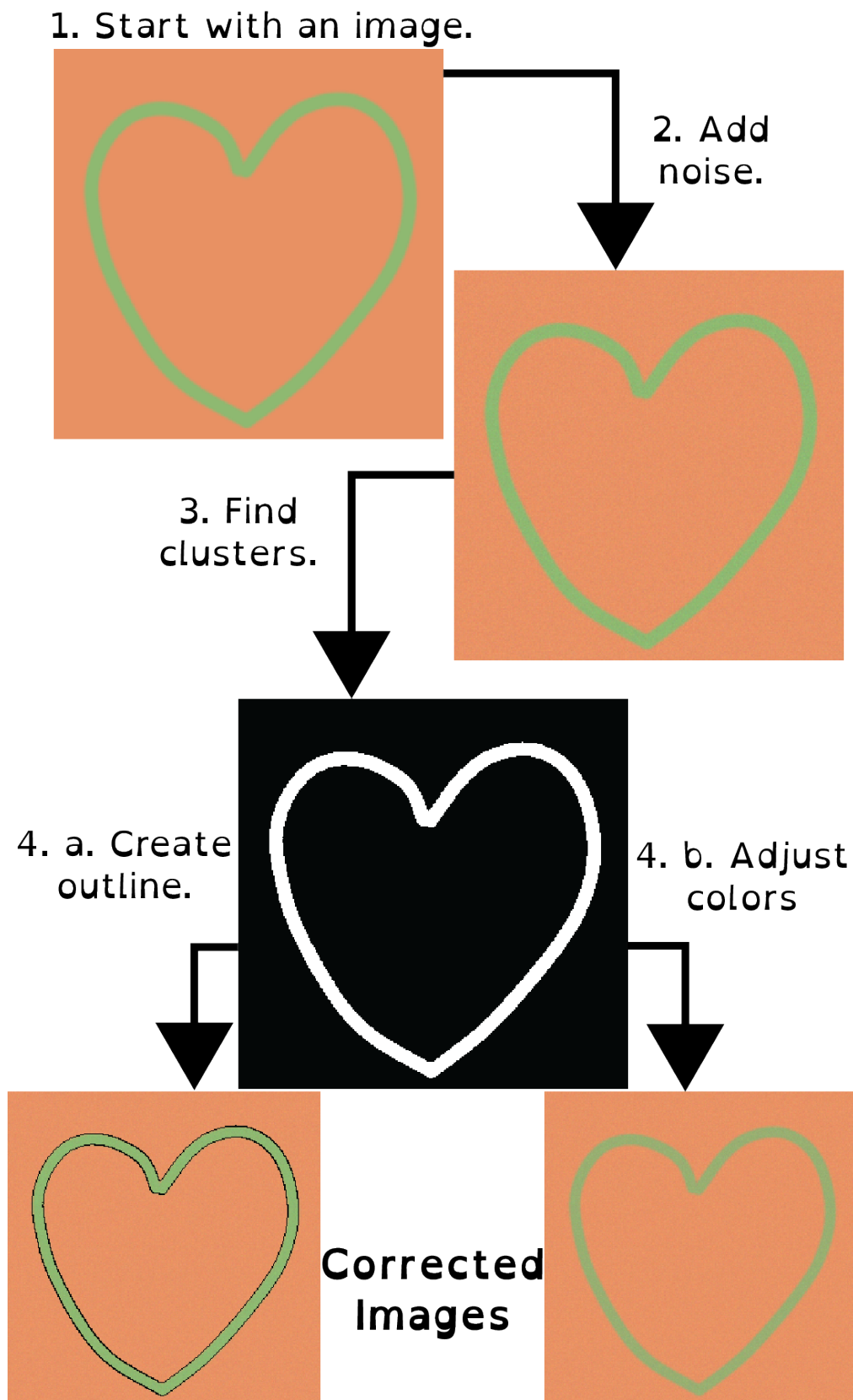
1. Start with an image.

2. Add noise.

3. Find clusters.

4. a. Create outline.

4. b. Adjust colors

Corrected Images

Figure 8: The correction process.

5   Future Improvements

5.1   Better Data

The data is currently created by one person that is red-green color blind. This is a humble start, everyone's color blindness is a little different. Thus, creating data that caters to only one person may leave out issues experienced by many other people with red-green and blue-yellow color deficiencies.

To solve this problem, the National Eye Institute or even an eye doctor could be inquired about the data. They could judge the data set created to see if there are any glaring cases missing from examination. Alternatively, they could provide the data themselves. These expert analysis of the problem would make the system more reliable.

5.2   Including Users in the Development

Any good software is going to include some sort of user testing. In this case, the users could help refine both the data and the correction functionality. Similar to an expert's opinion, getting feedback from many users will help create reliable data. This will ensure that no color combinations go un-learned by the system.

After the system has learned with a more inclusive data set, the users could also help judge the color correction. They could be given before and after pictures; offering their input as to whether the system actually accomplished its goal of helping them distinguish the colors in the image. Additionally, a non-color blind group of people could be brought in, again given the before and after pictures. These participants would be able to determine how drastic a change the system made to the image. Combining feedback from these two groups of people would help reach an equilibrium between making an image more accessible while not altering its meaning.

6   References

References

[1] Ryan Cross. How enchroma's glasses correct color-blindness, June 2016.

[2] enchroma. Frequently asked questions.

[3] National Eye Institute. Facts about color blindness, February 2015.

[4] Thales Sehn Körting. How k-means algorithm works, July 2013.

[5] Emily Singer. Color-blind monkeys get full color vision, September 2009.