Image Colorization
Keyvan Yaghmayi

We write a software that colorizes black-and-white images. The inputs are a black-and-white image and a color image (we may call it reference image). The software looks at the color image and colorizes the black-and-white image.

The idea is based on *histogram matching*: we find the histogram equalizer array for black-and-white image and assume it is the histogram equalizer array of the reference image. Therefore, we need to find the inverse histogram equalizer map from this array to the histogram of the color image. This inverse map does not necessarily exist, however, we make an approximation by choosing the closest value in the histogram equalizer array of black-and-white image as the histogram equalizer of the color image. Finally, we apply the resulting inverse histogram equalizer map of the color image to the histogram equalizer values of the black-and-white image.

Example 1: We want to colorize the following black-and-white image.

Figure 1: African Americans in Utah. Family of a black miner killed in the Castle Gate mine disaster, 1924. Courtesy of Utah State Historical Society.

One can use any color image as a reference image, however, to get realistic results the contents of the black-and-white image and the color image should be similar. The following color image would be a good choice because it has many white, gray, and orange pixels that we guess they existed in real scene of the black-and-white image.

The result is

In general, if all one has is a black-and-white image, then it is impossible to determine the original colors; simply because there are many different combinations of red, green, and blue channels that map to the same grayness level.

Some artists use digital software like Photoshop and guess the true color by eye. They look at the entire image as a whole and combine different layers of color to construct a colored image. In order to have a satisfactory result, it could take hours or even days to finish a single image.

The purpose of this project was to write codes and ask computers to colorize images for us. Note that there are functions, for instance *ind2rgb()* in Matlab, that colorize images for us, however, these functions mostly assign colors just to distinguish different levels of grayness and the result is far from being realistic. For instance, colorization of our figure by function *ind2rgb()* is as the following

In our software we set all red, green, and blue channels equal to the same value (that value was inverse histogram equalizer map of the color image applied to the histogram equalizer values of the black-and-white image). It would be an interesting project to apply machine learning techniques and colorize without having any color image.

In the next pages we provide more examples of colorization by our software.

Example 2: We colorize the following image

Figure 2: Christmas Truce, 1914. On Christmas Day, during World War I, soldiers from the opposing armies played soccer.

The following image will be a good candidate for reference image. I contains many gray, pixels that we guess there was in the black-and-white image.

The result is the following image

Example 3: We convert a color image into black-and-white and colorize it with our software. The original color image is the following

Figure 3: A woman watches a participant of the gay pride parade while posing for a photo in Madrid, Spain, July 2, 2016. Photo by *Daniel Ochoa de Olza*

We convert it into black-and-white

And we use the following image as reference image

The result is