FotoColor Design Document

Nicasia Beebe-Wang

CS50 Fall 2015

TF: Anita Xu

For uploading images, I adapted code from https://gist.github.com/no1k/8492575. Once the picture is uploaded, $_SESSION stores R, G, and B arrays for each pixel, and also stores the width and height of the image. In views/uploaded.php, you will find the document that displays the original image and shows options on the right side. Because COLORSELECT and COLORPOP requires the user to select colors, these had to be sent to new forms views/colorselect.php and views/colorpop.php respectively. GRAYSCALE opens in a new tab via a link to views/grayscale.php. Finally, for the K-MEANS COMPRESSION feature, I adapted code from http://harthur.github.io/clusterfck/demos/colors/clusterfck.js, and thus used JavaScript to dynamically display the reduced image on the uploaded page.

For both COLORSELECT and COLORPOP, the user is required to select colors to create their new image. I implemented the color selection tool from jscolor.com (which can be seen in the file js/jscolor.js).

GRAYSCALE – when brought to views/grayscale.php, first, the code checks to see if the $_SESSION data contains the R, G, and B arrays as well as height and width of the image. If not, it redirects to the main page for the user to upload an image. If this data does exist, a new image is created where for each pixel, the original R, G, and B values at that pixel are averaged so that each R, G, and B now contain the average value (equal values of R, G and B make gray).

COLORSELECT – Colors for COLORSELECT are passed into views/colorselect.php via a POST request, and information for the image itself is contained in $_SESSION as explained earlier. The four colors selected by the user are then transformed into an array of R, G, B values. For each pixel, we use the distance formula to calculate its distance from each of the user-selected color. We then identify which user-selected color is closest (has a minimum distance) to the current pixel. We then save that color in a new array for the COLORSELECT image, and then create and display the new image on the screen alongside the old image.

COLORPOP – This feature is similar to COLORSELECT in that it uses the R, G, B array and user-inputted colors and calculates distances among them. However, this feature also incorporates a user-selected similarity threshold from 0 to 10. The limitation is such that for each threshold value x, the pixel color is accepted if it's distance is less than or equal to 5000*x from any of the colors. If the color is

accepted, it is simply copied over to the COLORPOP image, however, if it is not within the threshold's limitation, it is converted to a gray-scale pixel.

K-MEANS COMPRESSION – For this feature, I adapted code I found from http://harthur.github.io/clusterfck/demos/colors/clusterfck.js (saved as js/kmeans_clustering.js). This function uses an array where each element is an array of R, G, and B values at each pixel. Functions saved in js/functions.js pass the image's RGB values array into the clustering function, and then draw a new image with new pixels where each pixel is converted to it's nearest mean.