Palette Scoring

- Jacob Kattampilly
- Eric Fernandez

Here, we present our efforts in the creation and evaluation of color palettes. Color palettes can be generated using several techniques. Usually designers look for inspirations for generating palettes, a common source of such inspiration are images. In this project, we use designer generated palettes as input data along with the images that inspired these palettes. We also generate palettes using different strategies, from the image itself. We compare these palettes (we assume the designer palettes are good) and analyze a few palette extraction strategies. We then generate scores for these palettes, comparing them to the designer palettes. This gives us some insight into which strategy may be used to generate palettes that are closer to designer palettes.

In the second phase, we extract certain features from the image. We use these features to train a model using lasso regression to predict the score. This gives us some insight into the features that are useful for determining the distance between two color sets.

Palette Extraction:

We use a set of approximately 4000-4500 palettes designed by artists along with the set of images that inspired these palettes. Using different strategies enumerated below we generate more palettes per image.

1) K-means clustering: This is a common technique used for clustering data. We use the k here as the number of colors that are present in the designer palettes.
2) Mini Batch K-means: This has been proposed as an alternative to the K-means algorithm for clustering massive datasets. The advantage of this algorithm is to reduce the computational cost by not using all the dataset each iteration, but a subsample of a fixed size. [1] Here again, the number of colors are the number of palettes in the artist image.
3) Random Sampling: Here we sample the pixels of the image randomly. using a normal distribution.

Once we algorithmically generate palettes we score these palettes by comparing them to the palettes generated by the artist for the same image. For scoring we used bipartite matching to map the closest colors in a palette and we find the distance for each palette from the corresponding artist palette.[2] We then calculate the score as:

$$\text{Score} = 1 - (\text{Normalized distance from artist Palette})$$

Note that this score indicates how close the algorithmically generated palettes are to the designer palettes and so gives an indication of how good they are.

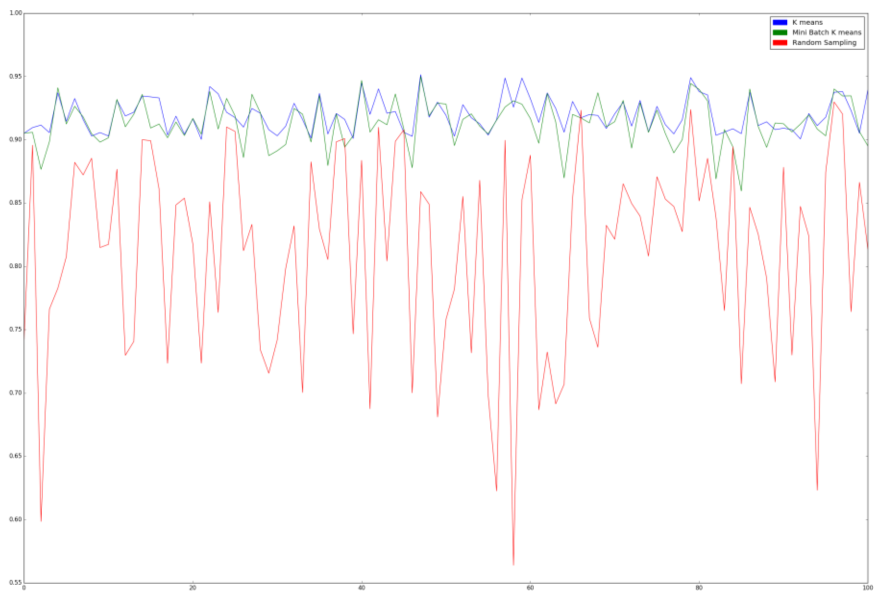


Figure 1 Comparision of Algorithimic Techniques for Palette Generation

Observation: Mini k-means generates scores that are very close to k-means generated scores. Even after reducing the computational costs we do not see a proportional reduction in quality. Hence, in situations where computational costs are expensive a cheaper costing algorithm may suffice.

We also observed that scores generated by these methods are clearly better than using random sampling.

Features

We decided to generate 6 features for each designer/algorithmically generated palette pair. These features are divided into two categories: pixel coverage features and color diversity features.

Pixel Coverage: Pixel coverage features are features that are used to determine how well a set of colors covers another set of colors[5]. Pixel coverage features include soft recoloring error, lightness coverage, and saturation coverage[5].

Soft Recoloring Error: We define recoloring error is defined as the total error when recoloring a designer palette by using the colors in algorithmically generated palette for a certain image.

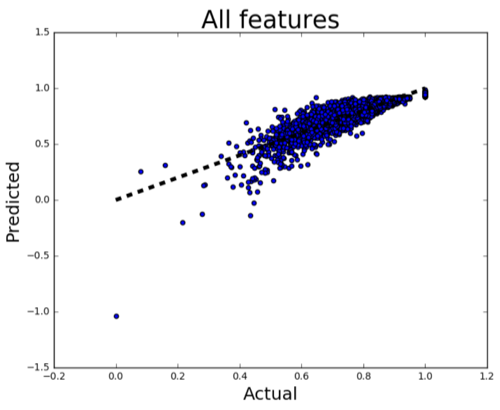
Lightness Coverage: Lightness coverage is the quotient of the difference between the maximum lightness and minimum lightness of an algorithmically generated palette and the difference between the maximum and minimum lightness of the designer palette corresponding to the same image as the algorithmically designed palette[5]. To obtain the lightness value of each color, we convert colors from RGB space to CIELAB space and get the L value that stands for lightness of a color.

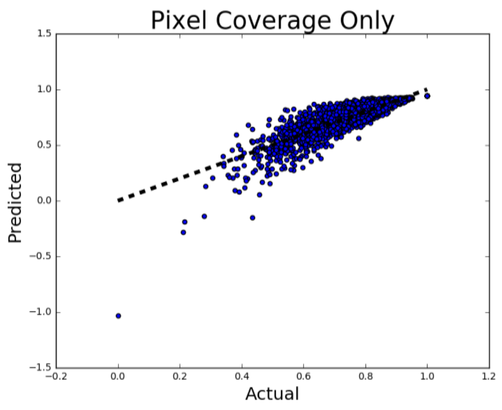
Saturation coverage: Saturation coverage is defined similarly as lightness coverage but instead of using maximum and minimum lightness we use the saturation obtained from transforming these colors to HSV space[5].

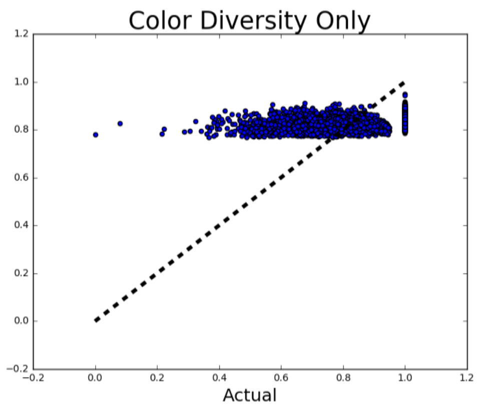
Color Diversity: The color diversity features determine how different are the colors in one palette[5]. These features include the min, max and mean of distances between colors in the algorithmically generated palette.

Lasso Regression Model Evaluation

We randomly divided our samples into two groups: 60% of samples for training our model and 40% for testing. We trained our model using 3 different set of features. The first set contains all the features, the second set contains only pixel coverage features and the third set contains only color diversity features. This was done to determine which features were the most relevant when comparing two palettes







We learned that pixel coverage features are good indicators of palette similarity while color diversity features do not give a good discrimination of the scores. When comparing the graphs for these sets, we observed that there is no much difference between the predictions made by the model using all features compared to the predictions of the model based on pixel coverage features alone. We evaluated our models each time using the coefficient of determination. The coefficient of determination is defined as

$$(1 - u/v)$$

where u is the regression sum of squares u = ((y_true - y_pred) ** 2).sum()

and v is the residual sum of squares

v = ((y_true - y_true.mean()) ** 2).sum()[4].

In our case, the coefficient of determination determines the proportion of the variance in the scores predicted by the model using the features[3]. The best possible value for the coefficient of determination would be 1 and 0 would be the worst. For the model using all features, we obtained a mean of the coefficients of determination of 0.80. For the model using pixel coverage features we obtained a mean of the coefficients of determination of 0.81 and for the color diversity features we obtained 0.05. This indicates that color diversity features alone are not good indicators of palette which in fact is correct since most of these features only determine the difference between colors in one palette.

We then tested the model by testing a good and two bad palettes (as provided by a designer)

The model predicted the results as follows:

	Good	Bad 1	Bad 2
Predicted:	0.67172722	0.25711189	-0.50468479

References:

[1] D. Sculley. Web-scale k-means clustering. In Proceedings of the 19th international conference on World wide web, pages 1177–1178. ACM, 2010.

[2] Paulina Volkova, Automatic Web Page Coloring

[3] http://stattrek.com/statistics/dictionary.aspx?definition=coefficient_of_determination

[4] Scikit http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

[5] Sharon Lin, Pat Hanrahan, Modeling How People Extract Color Themes from Images