Using the K-Means Algorithm to Recolor Pagan Women Paintings

Jeff Stevens, Dr. Matthew Kretchmar, CS 339

17 March 2021

1 Introduction

The K-Means algorithm is popular and effective algorithm used for the purpose of finding clusters of points in a data set. In this project, we propose using the K-Means algorithm on an image for the purpose of vector quantization and image reconstruction. Essentially, we intend to find groups of pixels in the image which are similar colors, find the mean color of these groups, and then recolor each pixel to the mean color of their group. The result will be an image reconstructed with different colors than the original image which will still tend to appear very similar to the original. Our particular experiment with this vector quantization process involves sampling the mean color data from one image and using it to recolor the pixels in a different image.

The images that we recolor are the Baroness paintings of John Dyer Baizley. These paintings are all fairly similar in that they contain pagan women with flowers and animals, but they differ in that each painting each is themed around a specific color. There are six paintings aptly titled Red, Blue, Yellow, Green, Purple, and Gold & Grey. Our goal is to use vector quantization to apply the color scheme of one painting upon another, as to see the colors of skin, flowers, and hair in one painting be transferred to the characters and scene of another painting,

(From left to right, top row to bottom row)
Red, Blue, Yellow, Green, Purple, Gold & Grey

For a brief introduction to steps the K-Means algorithm, the algorithm is written in pseudocode below:

Algorithm 1 K-Means

```
Select value for K, the number of centers
Initialize centers by giving each of them a random starting position
loop
  #Assignment Phase
  for each data point d do
    Find the distance from d to each of the K centers
    Assign d to the closest center
  end for
  #Adjustment Phase
  for for each center c do
    Compute the mean of all data points assigned to c
    Change the position of c to the computed mean
  end for
end loop
when no centers move
```

To begin this algorithm, we create K objects called centers. Centers are objects that contain the same number of dimensions as any data point d in our data set, and each dimension of each center created will begin with random values.

Next, for each data point d, we find the Euclidean distance between d and each of our centers. We find the center d is closest to, and then assign d to that center's data point group.

Continuing, for each center c, we find the mean of all data points in c's data point group. This calculated mean now becomes the new value for c.

After these steps, we check to see if the values of c are unchanged from when we began the algorithm. If so, we repeat instructions starting from the second paragraph on this page. Otherwise, we halt the algorithm. The result obtained from this algorithm is a cluster of data points that are reasonably similar to one another, given the value for K.

The vector quantization technique is also found below:

Algorithm 2 Vector Quantization

```
for each center c do
  for each data point d assigned to center c do
    Assign the value of c to d
  end for
end for
```

Essentially, when we vector quantize our data points, we make each data point's value equal to the value of the center it is assigned to from the output of the K-Means algorithm.

2 Methodology

To undertake this project, we use Python 3 in the Jupyter Notebook interface. To interact and manipulate the images, the Pillow image library and Numpy library are used. The images of the Baroness paintings are 545x545 pixels in size and constitute arrays of 297,025 pixels each once they are converted to a Numpy array format. Each pixel in the Numpy array of an image represents a 3-index sub-array containing the red, green, and blue values [R,G,B] for the pixel. Each pixel in this scenario represents a single data point with three dimensions which our K-Means algorithm will attempt to find clusters of in 3D space.

The K-Means algorithm is written from scratch and places a variable K cluster centers randomly between 0 and 255 for each of its R, G, and B values. The algorithm runs until the locations of all cluster centers remain the same after two iterations. The K-Means algorithm returns a variety of information: the cluster centers, the number of iterations to convergence, the final error function value, the cluster grouping of pixels, and the initially randomized cluster center positions.

The last important function is the vector quantize and recolor function. The center positions of a color-sampled image are assigned to cluster groupings of an original image based on what center positions in the original image they are closest to in Euclidean distance. This process essentially changes the color of groups of pixels in original image to produce a unique, recolored image. the The intention here to take similarly colored features of one image, like skin and hair, and impose them onto another image.

3 Results

The given methodology is implemented here to recolor the Baroness paintings. The following are each of the images recolored using the K-Means algorithm and vector quantization to the mean colors in each other image. The value for K used to run the K-Means algorithm on each image for color clustering is 4.

Original Blue

Vector quantized reconstructed-recolors

We are successful with simple recolors for Blue. It appears that the mean colors extracted from Yellow do not produce as clean of a reconstruction as all of the other colors. We must note that the hair and skin of other images seem to be recoloring well to present in a similar way they do in the color-sampled images.

Original Red

Vector quantized reconstructed-recolors

Red also seems to recolor well regarding the skin and hair of the women. We do also notice though that the flowers of the Gold and Grey recolor do not become the strikingly bright yellows and oranges found in the original image, nor do we recolor to the bold magenta seen in the flower of the original Purple. So far we appear to be lacking in matching details. These reconstructions seem to have been "grayscaled" to different colors.

Original Yellow

Vector quantized reconstructed-recolors

Yellow's recolors continue to net similar results to our previous experiments. At this point, we decide to increase our number of colors K in our future experiments to try to obtain more colors for detail.

In an attempt to pursue our original goal of imposing the colors of certain details onto the same details of another image, we use varying values of K to recolor the original Purple using the color data sampled from the original Gold & Grey.

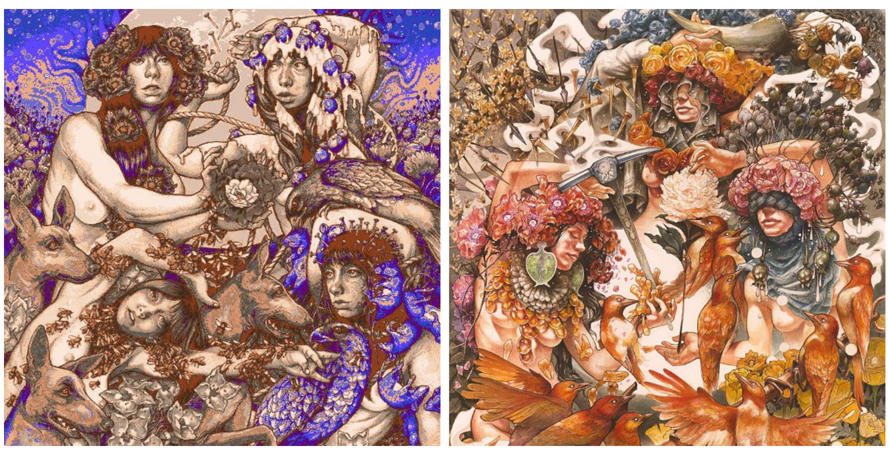
K = 4

K = 10

K = 16

K = 32

K = 50 reconstruction vs. original "Purple"



K = 50 reconstruction vs. original "Purple"

4 Discussion

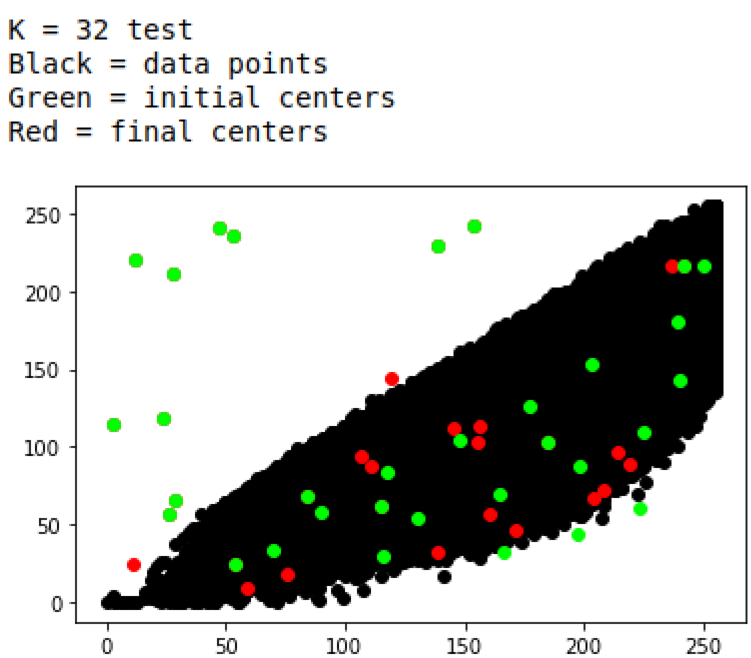
It appears that a Baroness painting recolors decently well using the mean colors from its sister paintings. However, we have been unable to capture desired features of images and impose them onto other desired parts, especially when there is a vast color difference between the features.

Our method of assigning colors from one image to another is picking the color that matches the color in the original image the most closely. The effect of this can be seen in the K = 50 reconstruction of "Purple". As we increase the value for K, we actually grow more closely toward replicating the original "Purple".

We do actually extract the vibrant oranges and yellows from Gold and Grey for our reconstruction, but they are only sparsely visible on the bees covering the woman in the lower left-hand corner of our K = 50 reconstruction. We will need to design a new technique to assign colors if we want to see more of Gold and Grey's details come through our future reconstructions.

Another fascinating occurrence we should discuss includes the presence of blues and purples in our reconstructions. Since we are taking the color data from Gold and Grey, an image with seemingly no presence of blues or purples. Let us examine the K-Means clustering data of our K = 32 reconstruction of purple, as it seems particularly faithful to the original.

The following is the cluster data for Gold and Grey for running the K-Means algorithm at K = 32. On this graph, our pixels' red values are on the X-axis and their blue values are on the Y-axis.



We find that the final locations of the centers after the K-Means algorithm runs, the green points, do match the general spread of the data. However, we must notice the green points that find themselves ending outside of the trend on this red vs. blue value graph. This probably due to the centers gravitating towards the position of some points' green values, but we still see the phenomena of our K-Means algorithm assigning colors that do not seem to be within our original data spread to the image we are reconstructing. A scenario to consider may be a single center wanders into a position that creates a purple that is useful for reconstructing our image, and therefore we may see a lot of that purple throughout our reconstructed image. This informs us for future attempts at our color assignment method, so that we might be more inclined to assign large clusters of pixels to similarly sized clusters for recoloring.

5 Conclusion

Our attempts at recoloring Baroness paintings using the K-Means clustering algorithm have yielded some interesting visual results, but remain unfinished with respect to requiring more experiments to create unique recolored images instead of reconstructions of original images. In future iterations of this project, we expect to add efficiency to our algorithm to boost testing speed, to record more data in graphical form for presentation purposes, and to develop a robust color assignment method.