
STA 160 Final Project

Elizabeth Stampher, Jianing Zhu, Sandeep Nair, Emilio Barbosa-Valdiosera

Department of Statistics

University of California, Davis

Davis, CA 95616

Abstract

This project involves analyzing Andy Warhol's five "Shot Marilyn" paintings from 1964 by comparing the colors and outlines across five different variations: Light Blue, Sage Blue, Red, Orange, and Turquoise [1]. The project utilizes computer vision techniques and image processing algorithms to quantitatively compare the colors and outlines of the paintings. Additionally, image manipulation techniques such as image compression, K-Means clustering, and hierarchical clustering are applied to explore the characteristics and variations in the artworks. The investigation's conclusions provide insight into Warhol's artistic decisions and further the study of art analysis and interpretation.

1 Introduction

In 1964, Andy Warhol, a key player in the pop art movement, created the iconic "Shot Marilyn" paintings. Monroe's death and her enormously iconic image inspired the artist to depict her utilizing the experimental repeating techniques that he had become interested in early in his career [2]. The goal of this project is to evaluate and compare the colors and outlines in five varieties of Warhol's "Shot Marilyn" paintings: Light Blue, Sage Blue, Red, Orange, and Turquoise. Computer vision techniques and image processing algorithms are used in the investigation. We investigate the differences in color palette and outlining approaches used by Warhol in the series using a combination of quantification and visualization. Scatterplots and histograms are created to compare the distributions between color channels, and picture manipulation techniques such as image compression using singular value decomposition (SVD) and clustering are used to uncover additional information. Our goal is to explore the intricacies and distinguishing qualities of each variation, shedding light on Warhol's aesthetic decisions and the relevance of these variations within his larger body of work. We provide a systematic and rigorous analysis of the paintings using algorithmic identification of regions of interest (ROIs) and image manipulation techniques such as image compression and clustering, contributing to the larger discipline of art analysis and interpretation while highlighting the intersection of technology and art.

2 Exploratory Analysis

2.1 Data Importation

We began this project by importing the data directly from the website using the 'requests' python package. After the images were imported, we created an image dictionary to easily call individual images and create loops for further exploratory data analysis. Directly importing the images from the website link resulted in us obtaining the images in a .png format, with a resolution of 960 by 960. A for loop using the image dictionary was then constructed to display each of the images, ensuring that the data was imported properly:



Figure 1: Imported Images

2.2 Image Manipulation

In order to further our understanding of the image data presented to us, we conducted various forms of exploratory data analysis. Firstly, we obtained the dimensions and sizes of each of the images, with the first two dimensions containing 960 values (the resolution of the images), and the third dimension containing four. These four values of the third dimension correspond to the Red, Blue, and Green data, as well as the transparency data. This transparency data, called the Alpha Channel, was present because we downloaded the images directly from the website in .png format, as opposed to the .jpeg format which would not contain the alpha channel. Using this knowledge, we started by splicing each of the layers of the third dimension, creating four two-dimensional images showing versions of the photo without certain colors present. For example, we spliced the third image into four matrices based on the specific color values. We then plotted the first of these matrices, which shows the red color values present in the image, along with printing the dimensions of the matrix, that being the 960 by 960 two-dimensional image:



Figure 2: Third Shot Marilyn Image spliced at the first Matrix in the third dimension

2.3 Singular Value Decomposition and Reconstruction of Images

During the Image Manipulation Process, we decided to include a form of image compression in order to explore the structure of the images, which can then be later manipulated for the implementation of clustering methods. We achieved this by creating two functions, the first being a simple image compression function using Singular Value Decomposition, which splits each of the images into three matrices following this equation: $M = USV^T$, where the U and V^T matrices are defined as real orthogonal matrices corresponding to the values in the image matrix, and the S matrix being a diagonal matrix of the singular values (which are conceptually similar to eigenvalues, however are also applicable to non-square matrices, such as the imported images, as they have a third dimension with low values). We then created a second function which splits the inputted image into the four

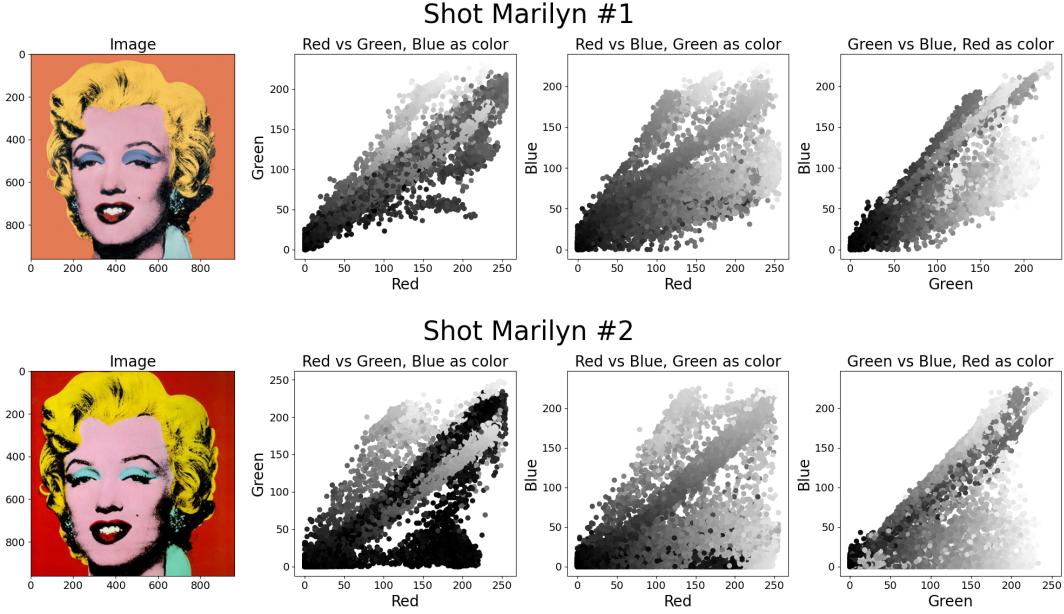
layers of the third dimension in order to create one wide tensor image of the four layers, then uses the SVD method of the first function in order to compress this tensor image. The amount of singular values which will be used in this function are determined by the user as an input of the function. Using this compressed tensor image and the defined k-singular values set by the user, the function then reconstructs the image entirely, and outputs the image, singular values used, and the reconstruction error. When experimenting with which k-singular values are possible, it can be seen that lower singular values used in the image reconstruction directly correlates with a higher reconstruction error, and subsequently, a poorly reconstructed image.



Figure 3: Third Shot Marilyn Image reconstructed with high k-singular values vs low

2.4 Color distributions

To investigate the color distributions of the images, we plot the red, green, and blue values of each pixel in a 2D projected scatterplot, where the third color dimension has been represented by the color of the data points, ranging from white to black with increasing value. The size of the images means that each image contains 921,600 pixels, so to avoid obscuring underlying patterns with overlapping data points, we randomly sample 40,000 pixels to plot instead:



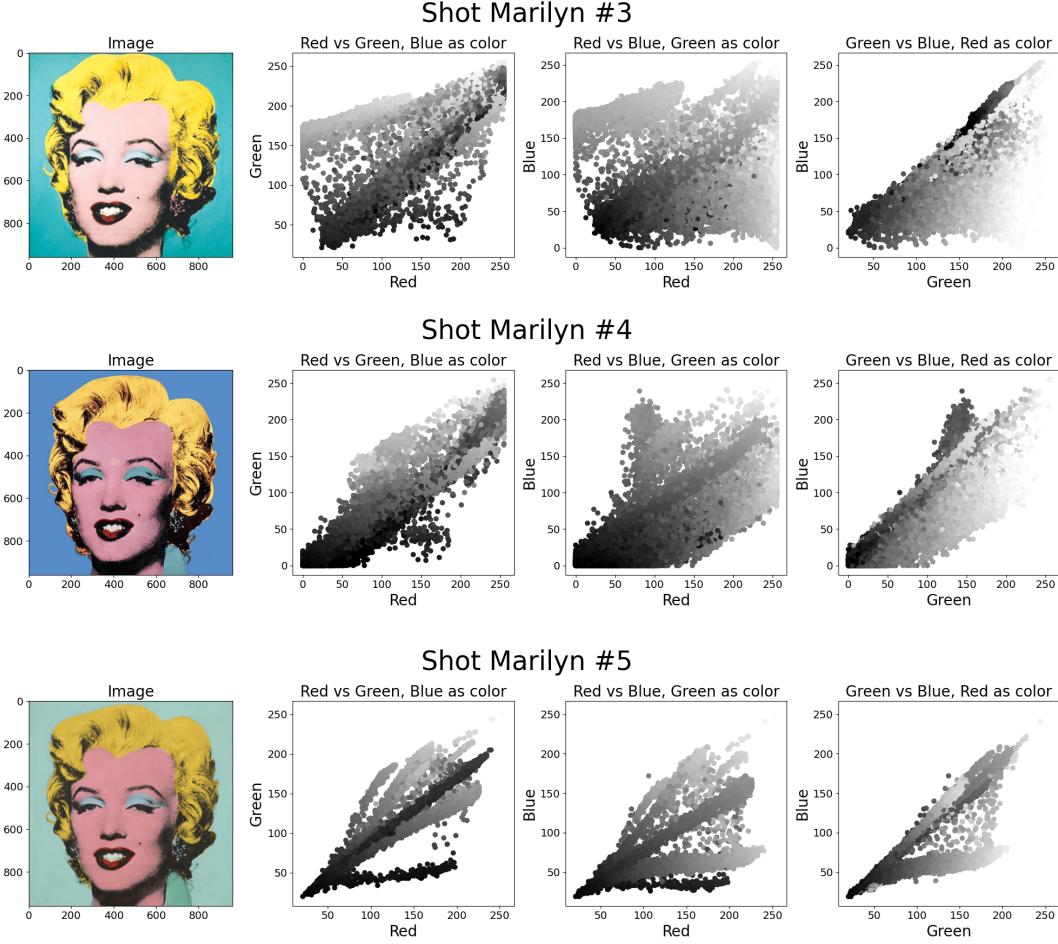
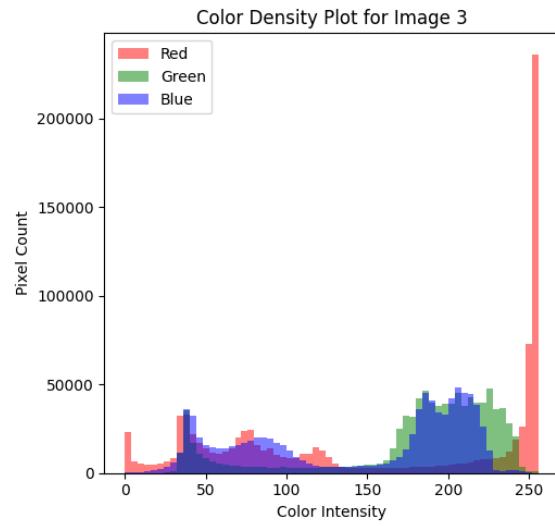
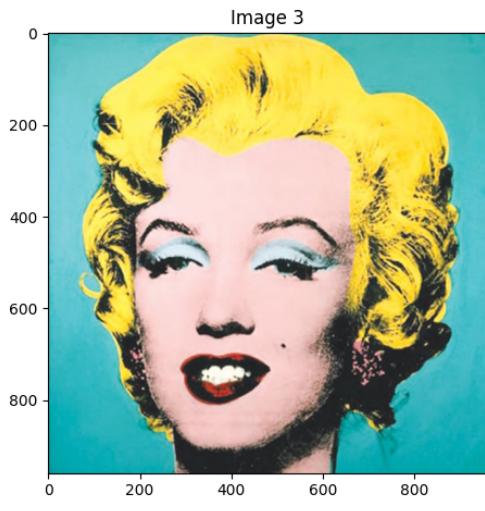
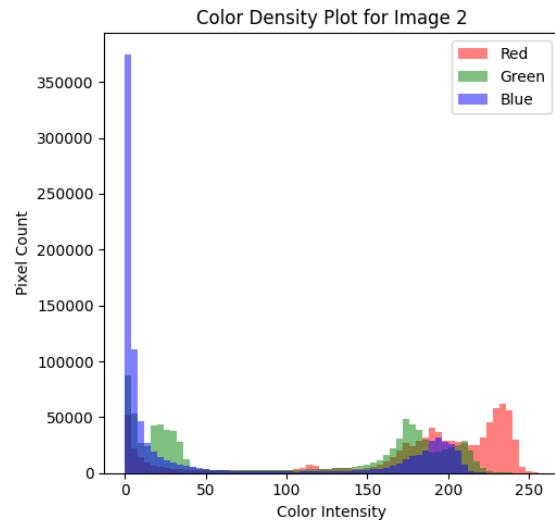
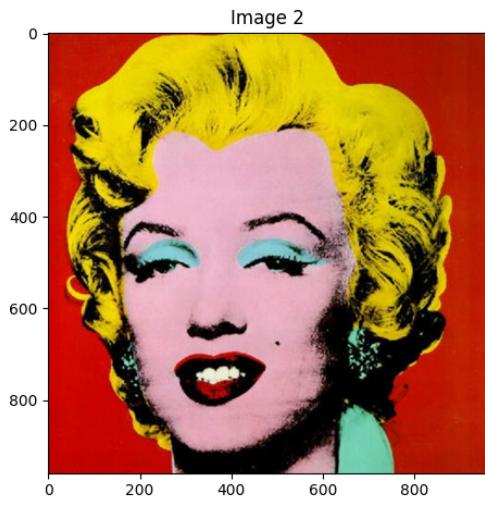
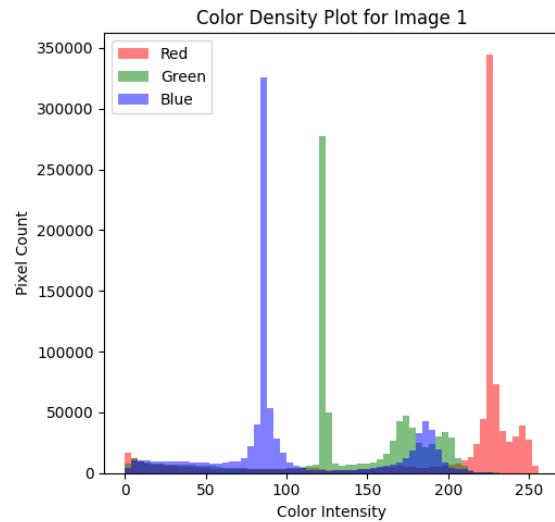
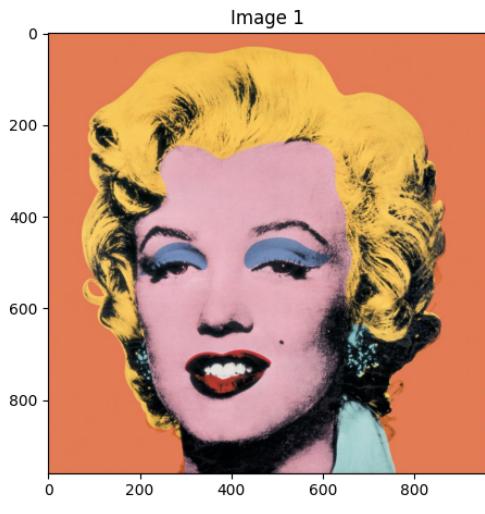


Figure 4: Shot Marilyns #1-5 and their distributions in RGB space, from 40,000 randomly sampled pixels.

Across all five Marilyn images, a strong central diagonal seems present in all RGB axis combinations. This seems to imply that all RGB values are highly present in every image, which might be in virtue of the repeated contrast in color choices between Marilyn's skin, hair, makeup, and background. However, each image also contains a flaring of values in favor of a particular color axis, which tracks most closely to the color of the background. Marilyn #1 features values biased toward its red and green axes, likely due to the presence of red, orange, and yellow hues. Marilyn #2 also features a heavy bias toward red and green, #3 features an increased presence of green, #4 an increased presence of blue, and #5 an increased presence of red. Interestingly, Marilyn #5's pixels don't seem to reach as far into the RGB space as the other images, which may be due to its relatively less saturated colors.

We now turn our attention to plots of the color density distributions in the images. We plot histograms of the frequencies that specific RGB values occur across all pixels:



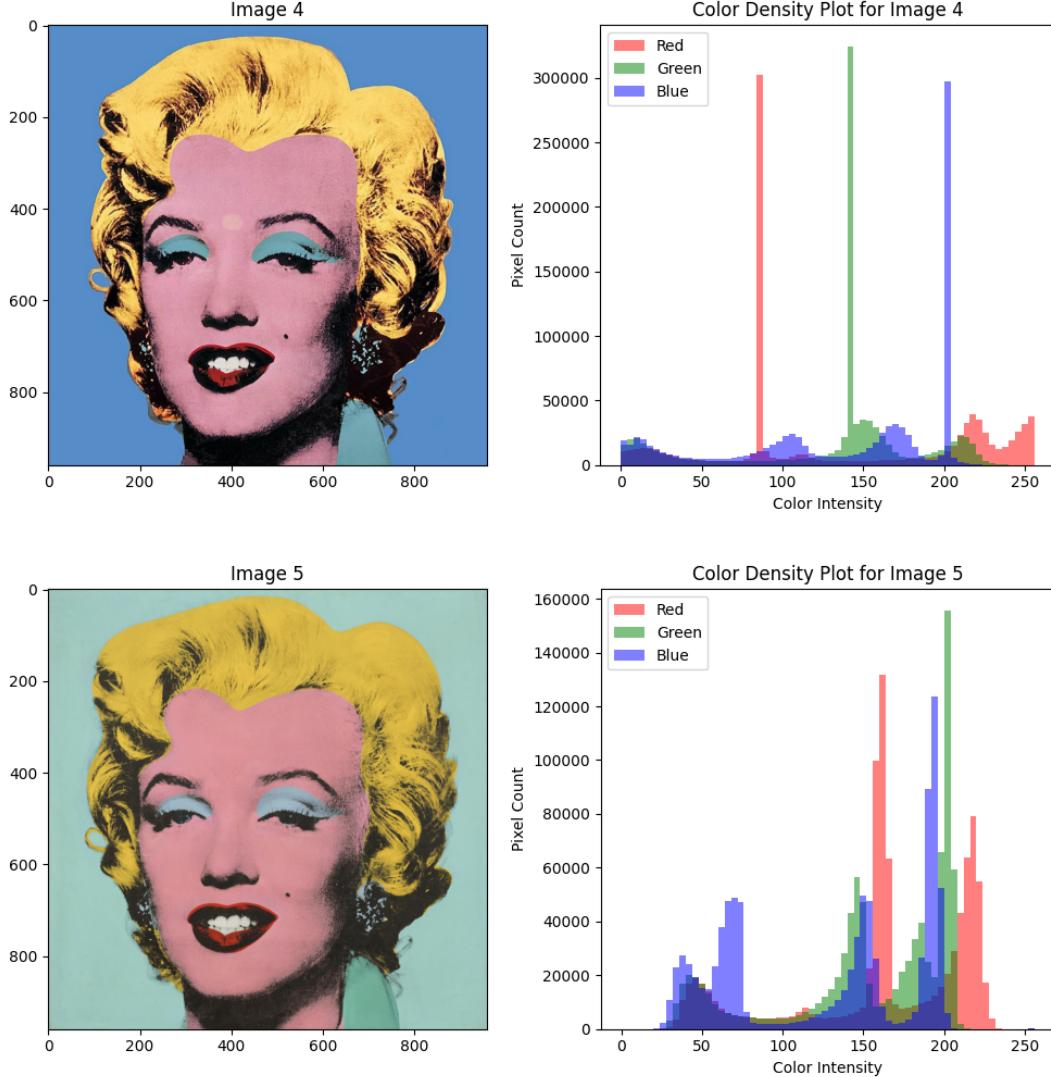


Figure 5: Pixel color histograms for Marilyns #1-5.

These histograms offer interesting findings. They are varied and multimodal, and of particular note are the large spikes in specific red, green, and blue values in image 1 and 4. It's unclear what is causing this, perhaps some sort of compression involved in the images as they were uploaded to the Internet resulting in a concentration in those values. Another unexpected finding is the high frequency of strong red values in Image 3, despite the image looking fairly blue. This is especially interesting when contrasted with Image 2 sitting right above it. It may be that the stark absence of blue values in Image 2 carry the red dominance in the picture even more strongly than the large presence of red values in Image 3, as it could mean that there are fewer green and blue values to counterbalance the presence of red. Otherwise, these histograms comport with the appearance of the images, and we see once again the reduced range of values in Image 5 which is most likely due to its relative desaturation.

We now move to the task of using these colors to process and analyze the images further.

3 K-Means

Image segmentation is a fundamental task in computer vision that aims to partition an image into distinct regions or objects. It plays a pivotal role in numerous applications, including object recognition, image retrieval, medical imaging, and autonomous driving. Among the plethora of segmentation algorithms available, K-means clustering has gained popularity due to its simplicity and efficiency. In this section, we explore the application of K-means clustering for image segmentation and analyze its effectiveness in capturing the inherent structures and patterns within images. We discuss the underlying principles of K-means clustering, its algorithmic workflow, and the considerations specific to its application in image segmentation.

In order to perform K means on images, We need to first select a appropriate value for K. We chose k=15 since this is the minimum number of clusters which clearly separates our regions of interest.

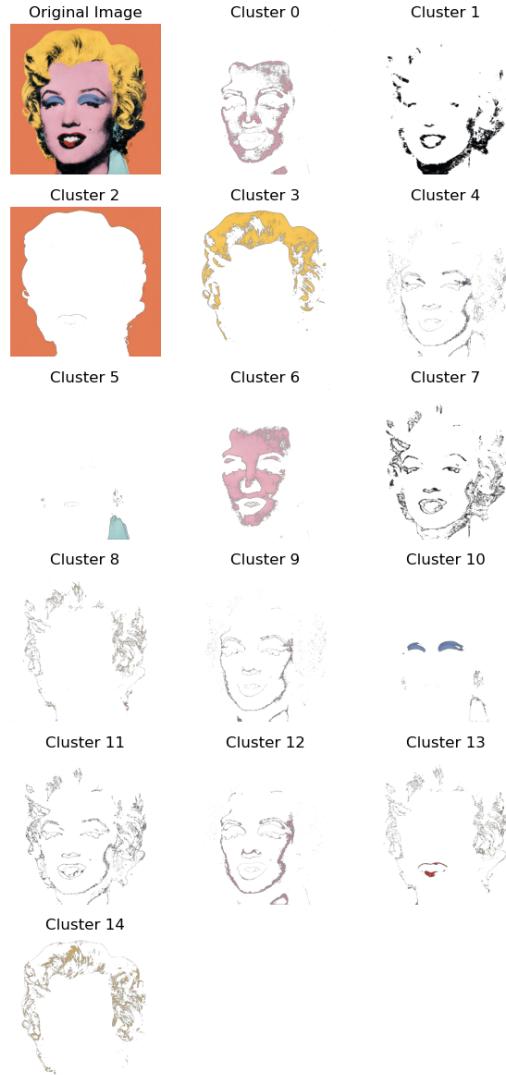


Figure 6: K-means segmentation using k=15

From figure 6, we can clear see that the background (cluster 2), skin (cluster 0,6 and 12), shirt color (cluster 5) and hair (clusters 0,6,12) are separated from the rest of the features. The rest of the clusters correspond to the lips, eye shadow and regions which are shaded. Features like hair and skin are spread across multiple clusters. We can now manually join these clusters to clearly differentiate these regions of interest. However, the eye shadow and lips are in isolated clusters. They are in

clusters with dark shady lines or regions. We need a different procedure to isolate these two clusters individually



Figure 7: Segmentation after grouping clusters together

From figure 7, we can conclude that that skin, background and shirt have been segmented properly. The lips and eye shadow are not in their isolated clusters. To solve this, we have to subset or crop the image. The lip and eye shadow areas have intense shading making it hard to isolate using k means. Cropping the image should provide better results.

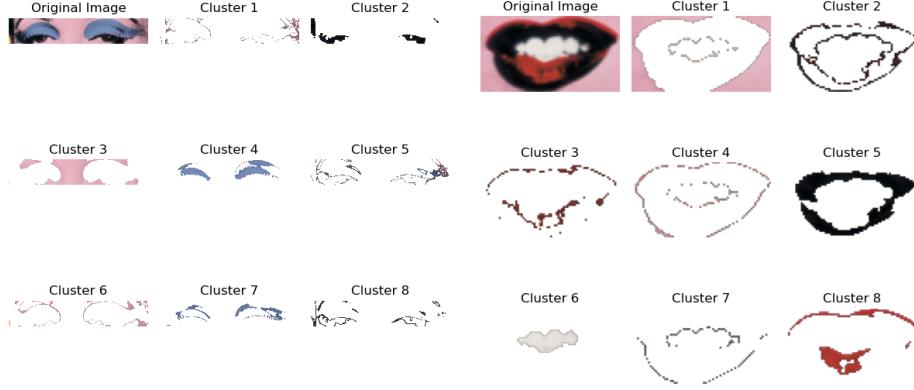


Figure 8: K-means segmentation on eye and lip areas with $k=8$

Now, we can hand pick the clusters which best represent the region of interest and combine them together like we did before with the skin and hair clusters. Figure 8 shows the segmented images and their spatial location with relation to the original image. The segmentation is now precise. Skin, cloth, background, hair, lips and eye color have been isolated with high resolution.

Limitations of K-means followed by hand picking clusters While this method is precise, the procedure requires human inputs at multiple points. The clusters which are similar and pixel ranges of cropped images need to be set by human inputs. This makes the process like automated and more tedious. Initially we were of the impression that all 5 shot Marilyn had the same spatial positioning of features. For instance, we expected the hair to encompass the same region across all images, with the only difference being the color distribution. However, this assumption was wrong. As seen in figure 9, Shot Marilyn's have slight differences in spatial positioning. The edges of each cluster bleed into one another. We can transfer labels from one image to the next. This means that for a precise segmentation for all 5 shot Marilyn's, we need to set labels unique to each image. Segmenting using k-means and hand picking clusters is already tedious, and doing it four more times is not time-efficient. Instead we use Hierarchical Clustering on k-means labels to segment the image.

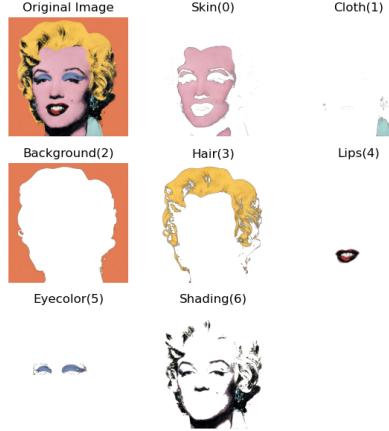


Figure 9: Final segmentation for orange Marilyn

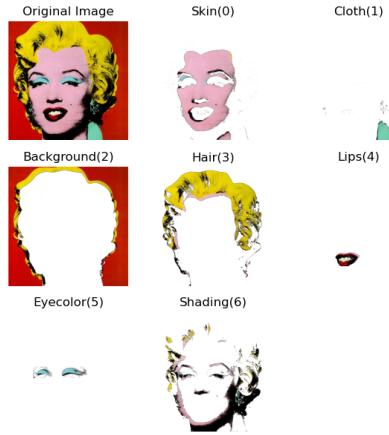


Figure 10: Transferred Labels on Red Marilyn

4 Hierarchical Clustering

Hierarchical clustering (HRC) can be better suited for data sets which have clusters of differing shapes, sizes, and variability. As we have seen in section 2.4, the distributions of color channel values per image are multimodal and otherwise not clearly normally distributed, so we employ the use of HRC as an alternative end point for identifying regions of interest in the images. The algorithm works by first setting every individual data point as its own cluster, and iteratively merging each point into larger clusters until a single final cluster is reached. There are different methods of cluster calculation used in HRC, and the method we use is the Ward calculation. These iterations are represented as a tree dendrogram, where each step of the clustering process is shown proceeding from the bottom of the tree to the top. A major limitation to the use of HRC with the present data sets is the computational intensity demanded by clustering almost a million data points in this way; Indeed, a raw feeding of the image data directly into the algorithm demands over 3 Terabytes of memory space, well beyond the capacity of available computational resources. To ameliorate this, we reference our previous usage of K-Means clustering as an intermediary step, reducing the total number of unique color values, and use these values for further processing by HRC. With K-Means used as an intermediary step, we keep its cluster number high, at $k=150$, for HRC to cluster further.

4.1 ROIs and color replacement

Once we have clustered the K-means color output with HRC, we extract the HRC groupings from the dendrogram and verify that these correspond to meaningful regions of interest (ROIs) in the images by replacing the pixel values corresponding to each group assigned to them by HRC with a pure

green color ($R=0$, $G=255$, $B=0$). In some cases, HRC pixel groups are manually combined together into a single group if the HRC segmentation appears too granular.

The results of the entire process, from K-Means input to ROI color replacement are presented for each Shot Marilyn painting below:

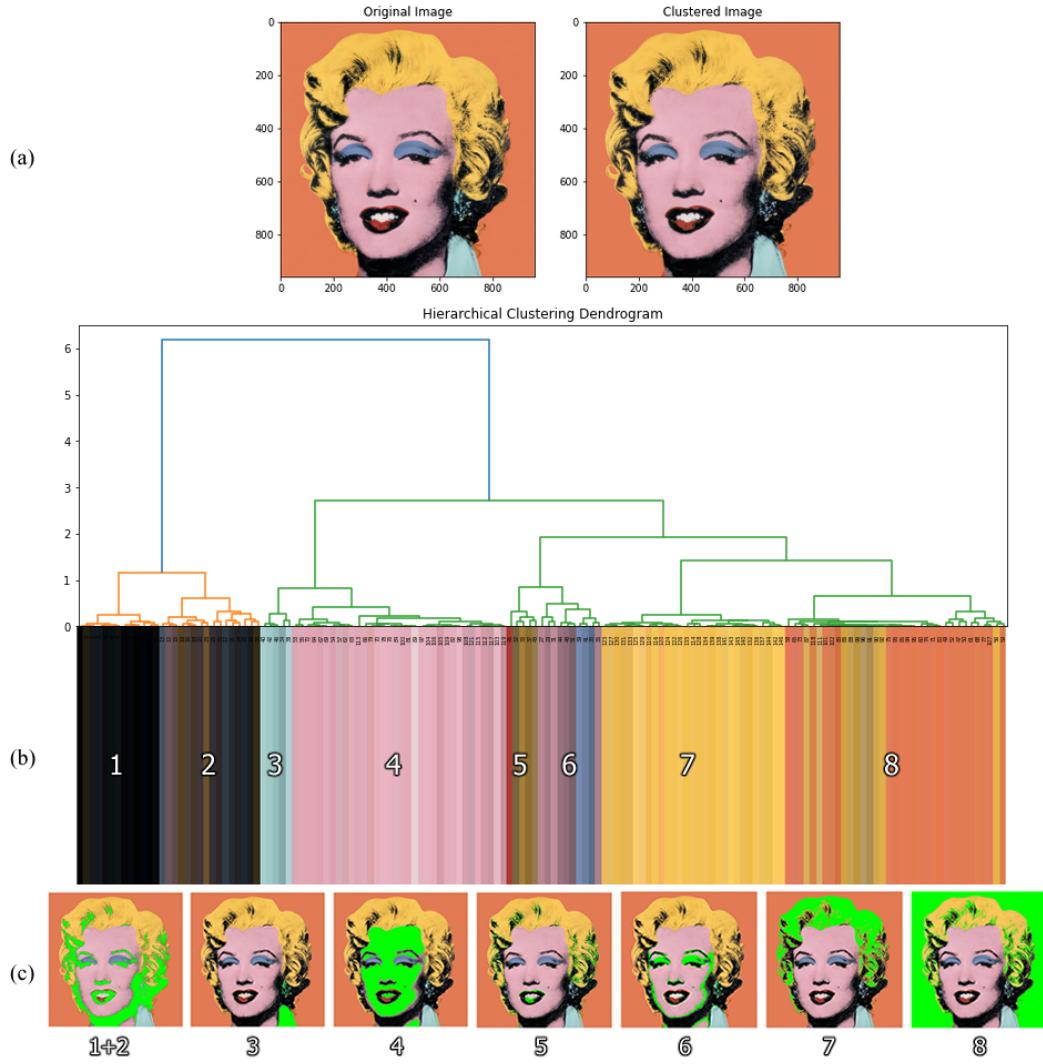


Figure 11: HRC process for Marilyn #1. (a) shows the initial K-Means clustering, which is fed into HRC in (b), and a dendrogram is produced. In (c), ROIs are highlighted in green in accordance with their HRC group assignments.

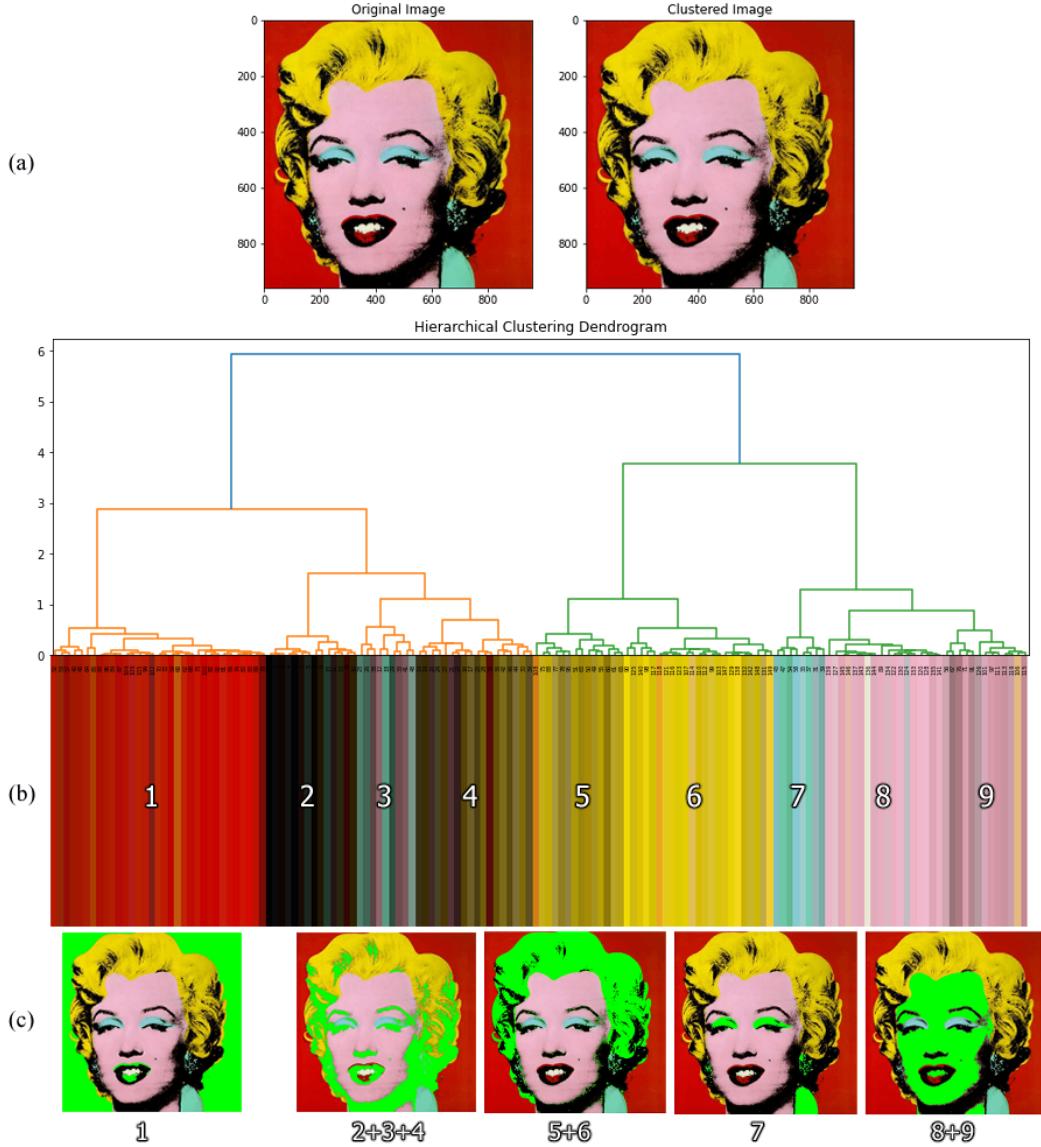


Figure 12: HRC process for Marilyn #2. (a) shows the initial K-Means clustering, which is fed into HRC in (b), and a dendrogram is produced. In (c), ROIs are highlighted in green in accordance with their HRC group assignments.

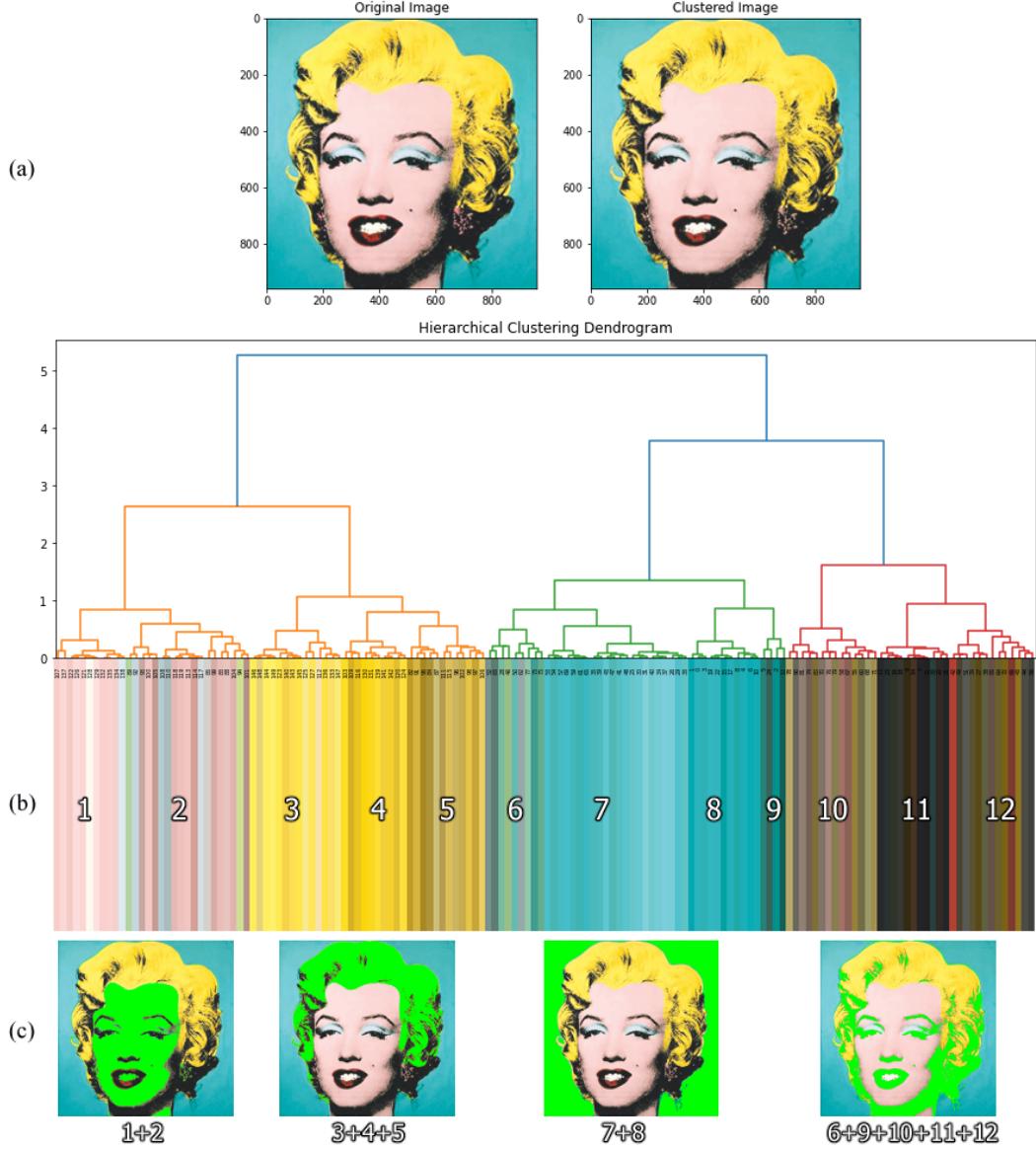


Figure 13: HRC process for Marilyn #3. (a) shows the initial K-Means clustering, which is fed into HRC in (b), and a dendrogram is produced. In (c), ROIs are highlighted in green in accordance with their HRC group assignments.

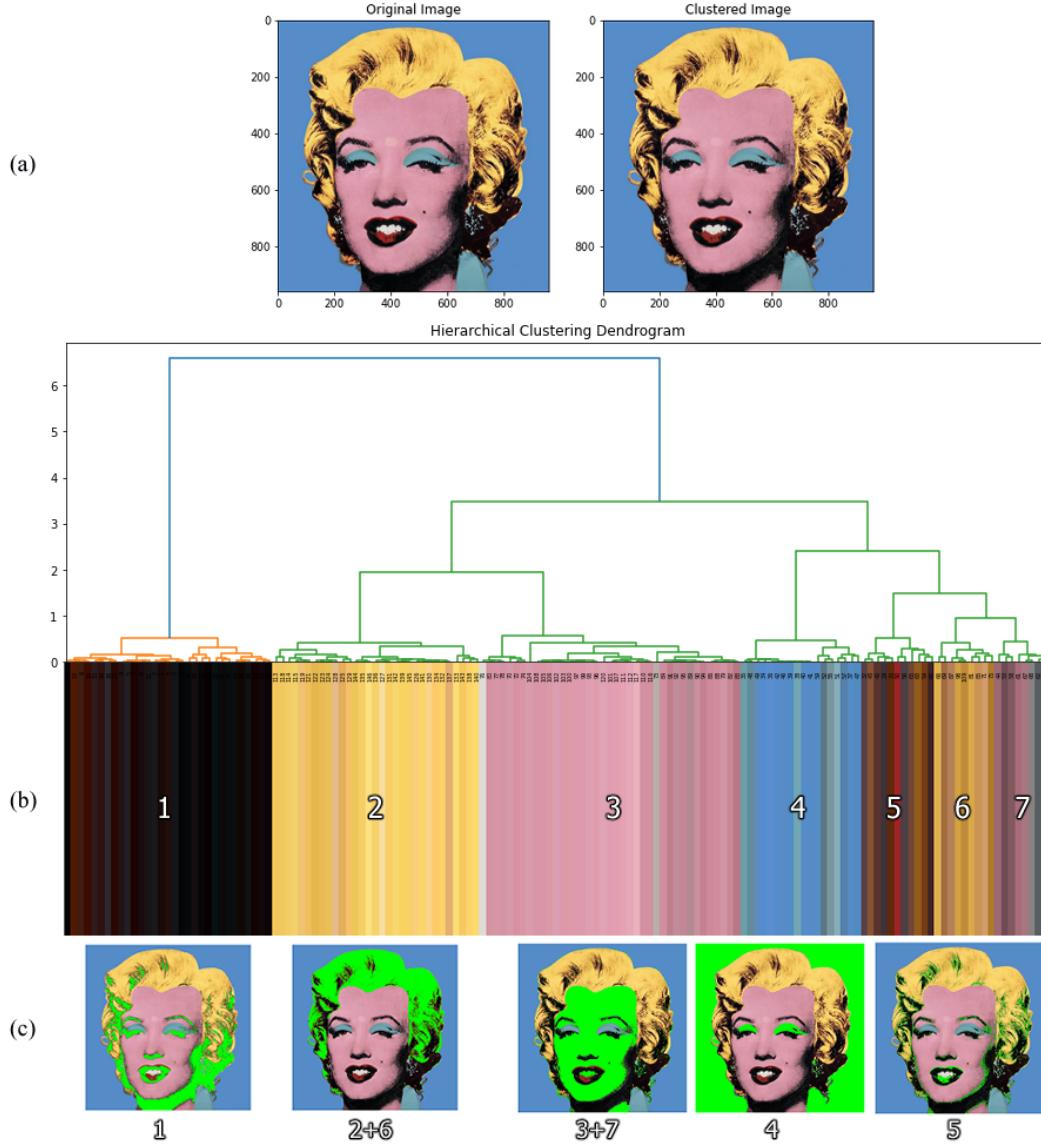


Figure 14: HRC process for Marilyn #4. (a) shows the initial K-Means clustering, which is fed into HRC in (b), and a dendrogram is produced. In (c), ROIs are highlighted in green in accordance with their HRC group assignments.

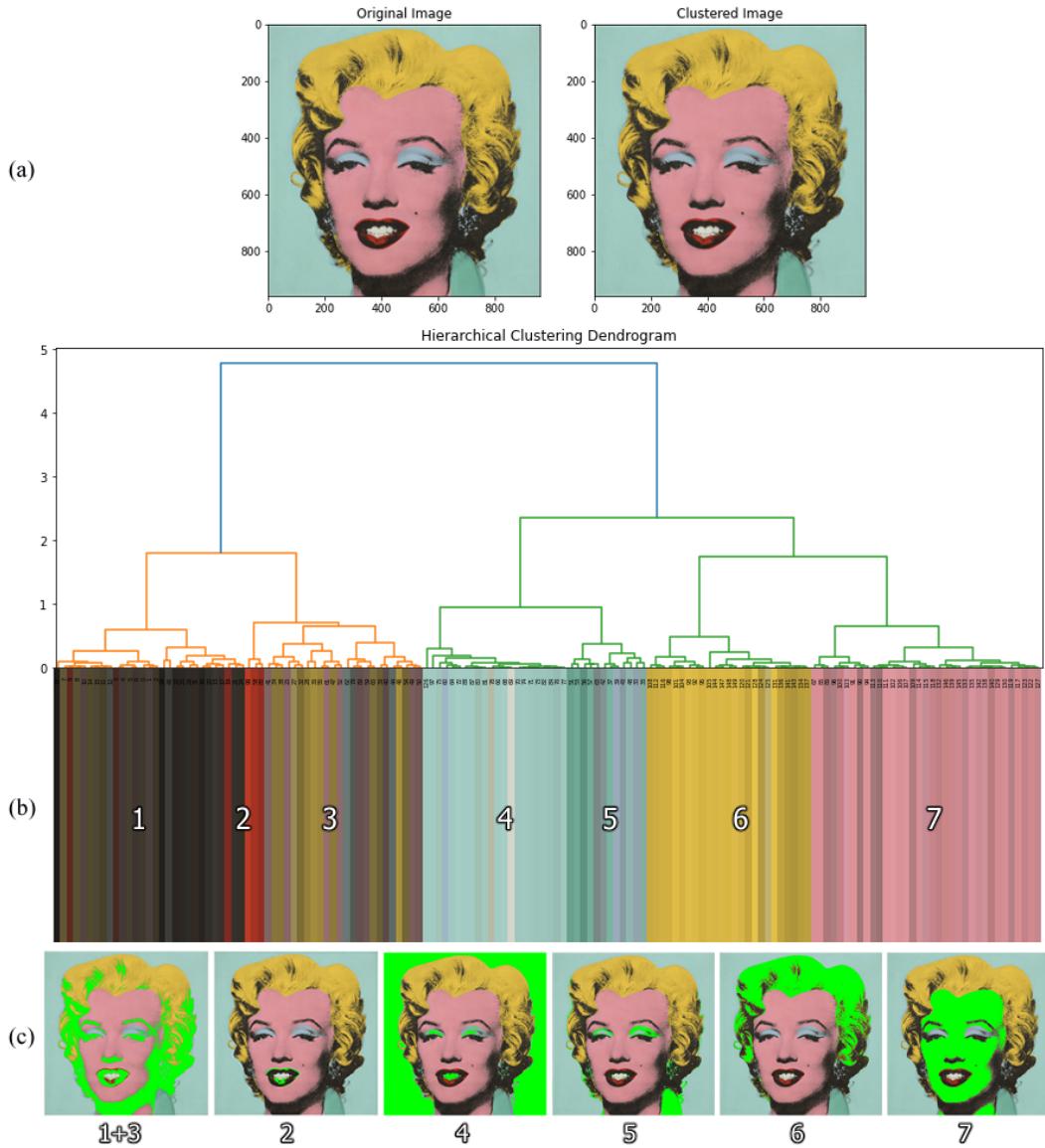


Figure 15: HRC process for Marilyn #5. (a) shows the initial K-Means clustering, which is fed into HRC in (b), and a dendrogram is produced. In (c), ROIs are highlighted in green in accordance with their HRC group assignments.

Overall, this method of HRC augmenting K-Means appears to work rather well. The algorithm consistently identifies Marilyn's skin, hair, outlines, and the background across all images, with some manual combination of HRC groups required in some cases. However, the algorithm does struggle to identify finer ROIs such as Marilyn's lipstick, eye shadow, teeth, and shirt collar. It performs best at identifying her lipstick in Fig.12 (c), group 1, and Fig.15 (c), group 2. It identifies her eye shadow best in in Fig.12 (c) group 7, and Fig.14 (c) group 4. It identifies her shirt collar best in Fig. 11 (c), group 3, and fails to identify her teeth in all images. This may be because the color corresponding to her teeth only takes up a single value across the 150 colors outputted by K-Means, and is almost always grouped in with her skin colors. Additionally, successful identification of finer ROIs tends to occur in images where that ROI shares similar colors with another larger region in the image, such as the highlighting of the lipstick in Fig. 12 (c) group 1, which matches the strong red background color, as well as the eye shadow and shirt collar which are successfully identified together in group 7.

5 Conclusion

Color and image processing poses a unique and interesting challenge for data analysis. For the problem of identifying regions of interest within these paintings, modified K-Means, while precise, requires a high degree of human intervention to meaningfully extract ROIs for even a single image. An augmentation of K-Means with hierarchical clustering substantially reduces the degree of necessary human intervention, but still fails to identify much smaller regions. While Warhol's bold and simple color choices make some ROI identifications easier, the instances where he chooses very similar colors for different regions is an aspect of these paintings which makes color-based identification of ROIs very challenging and perhaps unsuitable on its own.

From the HRC plots, we can observe that the shot Marilyn's with brighter colors like 2 and 3 are accompanied with brighter shades for hair, skin and eye shadow. Marilyn's 1, 4 and 5 have relatively neutral backgrounds. In these images, the shades of hair skin and eye shadow are comparatively neutral. Another interesting observation is that the shade of shirt color at the bottom right of the image stays the same irrespective of the background.

References

- [1] "Shot Marilyns." 2020. Wikipedia. March 16, 2020. https://en.wikipedia.org/wiki/Shot_Marilyns.
- [2] Revolver. 2022. "Andy Warhol, the Shot Marilyns, and His Early Silkscreens." Revolver Gallery. April 27, 2022. <https://revolverwarholgallery.com/andy-warhol-the-shot-marilyns-and-his-early-silkscreens/:text=Monroe>.
- [3] "A Visual Critique of Warhol's Shot Sage Blue Marilyn, 1964." n.d. The Interior Review. <https://www.theinteriorreview.com/story/2022/5/10/critically-assessing-warhols-shot-sage-blue-marilyn>.