Problem 3 – k-means clustering (30%)

Using the image 'soccer.jpg' included in the assignment, perform k-means clustering to vector-quantize pixels according to their RGB color.

References

- Introduction to Image Segmentation with K-Means clustering - KDnuggets
- Understanding K-means Clustering in Machine Learning | by Education Ecosystem (LEDU) | Towards Data Science
- Clustering with K-Means and reshape into a color image - Stack Overflow
- K-means clustering - Intro to Artificial Intelligence
- K-Means Clustering in Python: A Practical Guide

15% credit. Reconstruct the image using the colors in the codebook for values of k = 1,2,…,0. Generate a color JPEG image for each of the reconstructed images, and display all the images in a single 3x3 mosaic figure. Discuss which codewords emerge from the image as the codebook length increases?

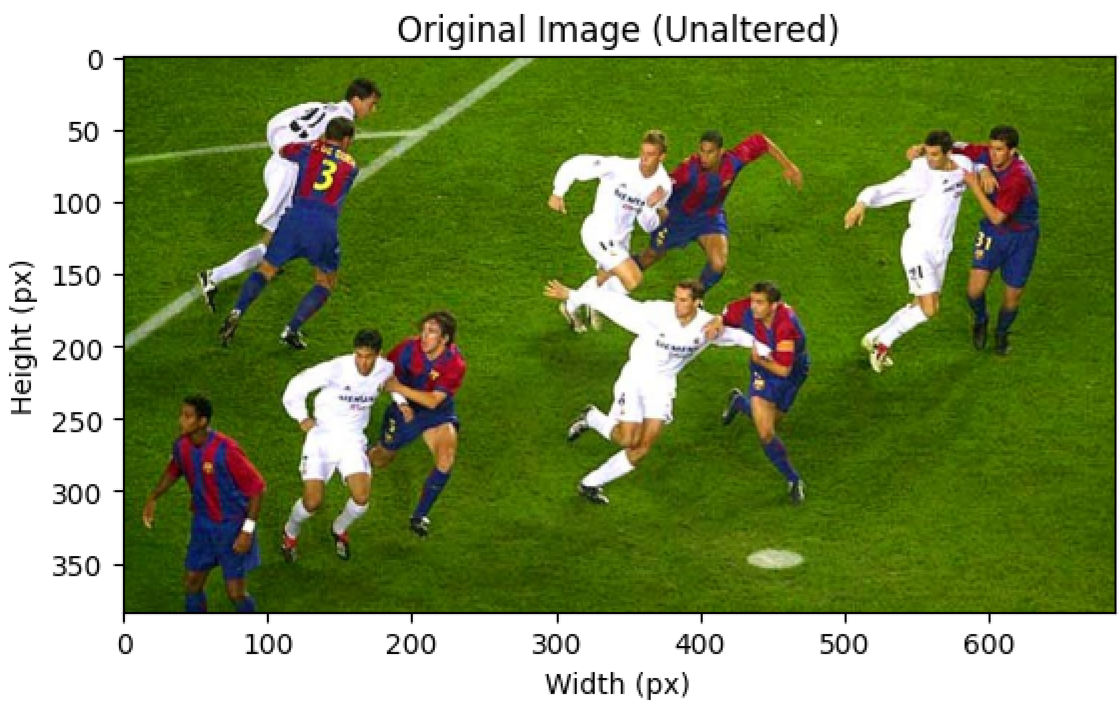

Figure 1. Unaltered image of soccer.jpg for reference

Reconstructed Images Using Different Values of K

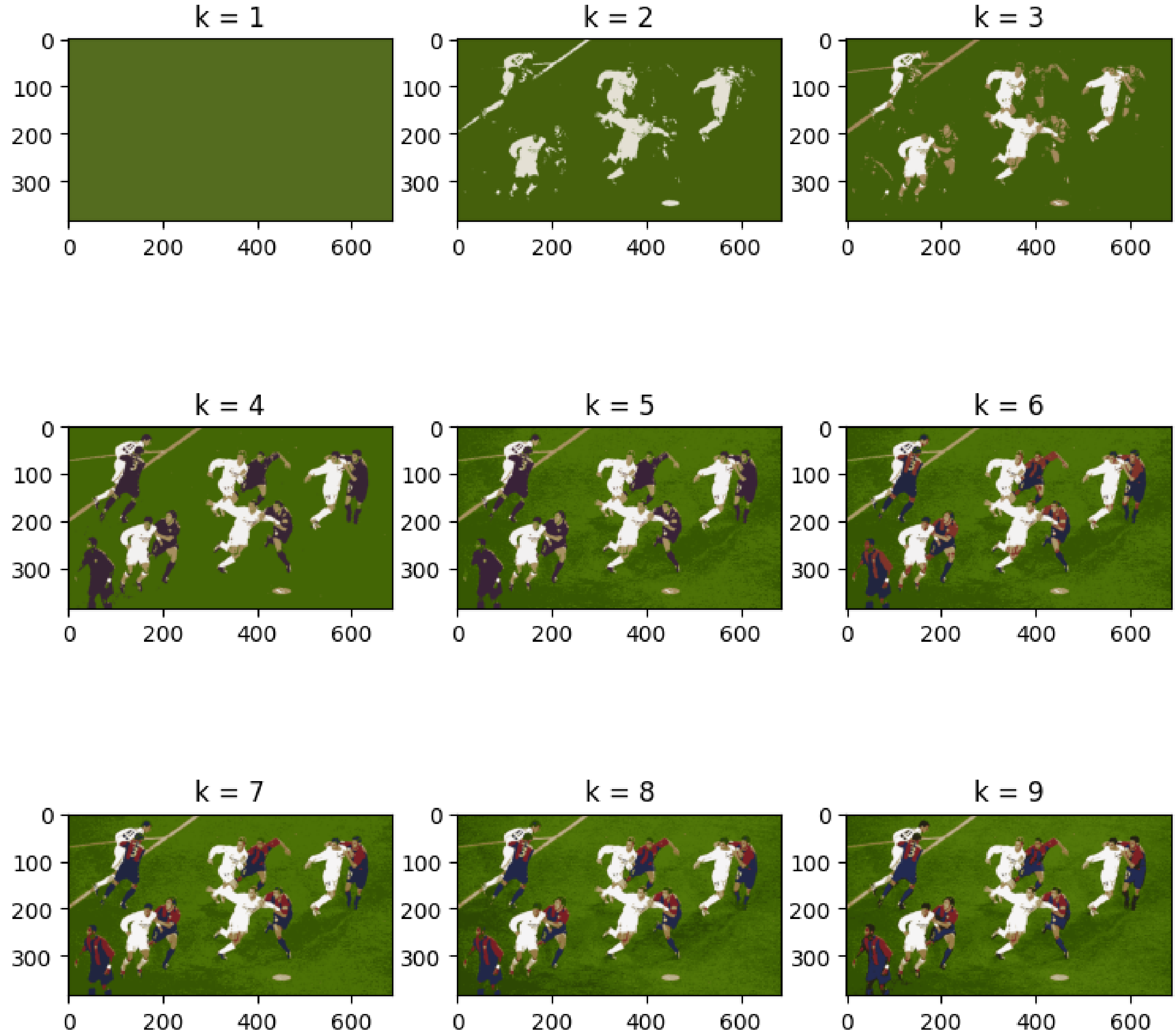

Figure 2: Reconstructed Images Following Vector Quantization Using Different Values of K

Discussion of findings:

As the codebook length increases (number of k clusters), codewords that pertain to describing the shape, position, and color of the players start to emerge. For example, when we set the codebook length to 1, we classify everything and the surrounding area as a flat green, which represents the color of the field.

The following codeword emergence is shown below (for the sake of more colloquial language, we describe the codeword emergence as the emerging features in the reconstructed image as we increase number of clusters):

- K = 2 gives us the general outline of the players in white, a bit of the players in red/blue, and the field lines as a flat white.
- K = 3 gives us a bit more of a better outline of the players–in particular, the beige tones of player skin begin to emerge.
- K = 4 gives us the uniform outline of the red/blue players as a flat color that appears to be a midway between red and blue.
- K = 5 gives us field shading, which gives us more texture and shadows cast by the players.
- K = 6 gives us more detail on the red/blue players uniform–in particular, this gives us the red coloration of the uniforms, which is similar to that of the original image, and the flat blue overlay on the red/blue players clothing becomes a more vibrant blue.
- K = 7 makes very minor shading changes–shading that was originally classified as a red color on player skin now changes to a lighter color.
- K = 8 also makes very minor changes, including coloration of player hair. Player hair no longer has the blue coloration associated with the color of red/blue player's clothing, and now appears to be a flat greenish color.
- K = 9 gives minor changes, which includes darker shadings associated with dark body shadows on player bodies as well as hair coloration.

Figure 3. Original Image (Left) Against Reconstructed Image Using K = 9 (Right)

When looking at the K = 9 reconstructed image against the Original Image, though a quick comparison between the two images shows that though the original image is much more vibrant and detailed, the reconstructed image (which uses only 9 colors due to k = 9) retains all the important features of the original image and the general outlines of notable shapes.

In short, though it is obvious that the image on the right lacks depth and is somewhat flat, it still conveys the same information from the original.

15% credit. Generate a plot that shows the sum-squared-error (SSE) between the reconstructed image and the original image as a function of k, the number of clusters. To do so, repeat part (a) several times and, for each k, record the clustering that gives the minimum SSE. Can you make sense of this plot?

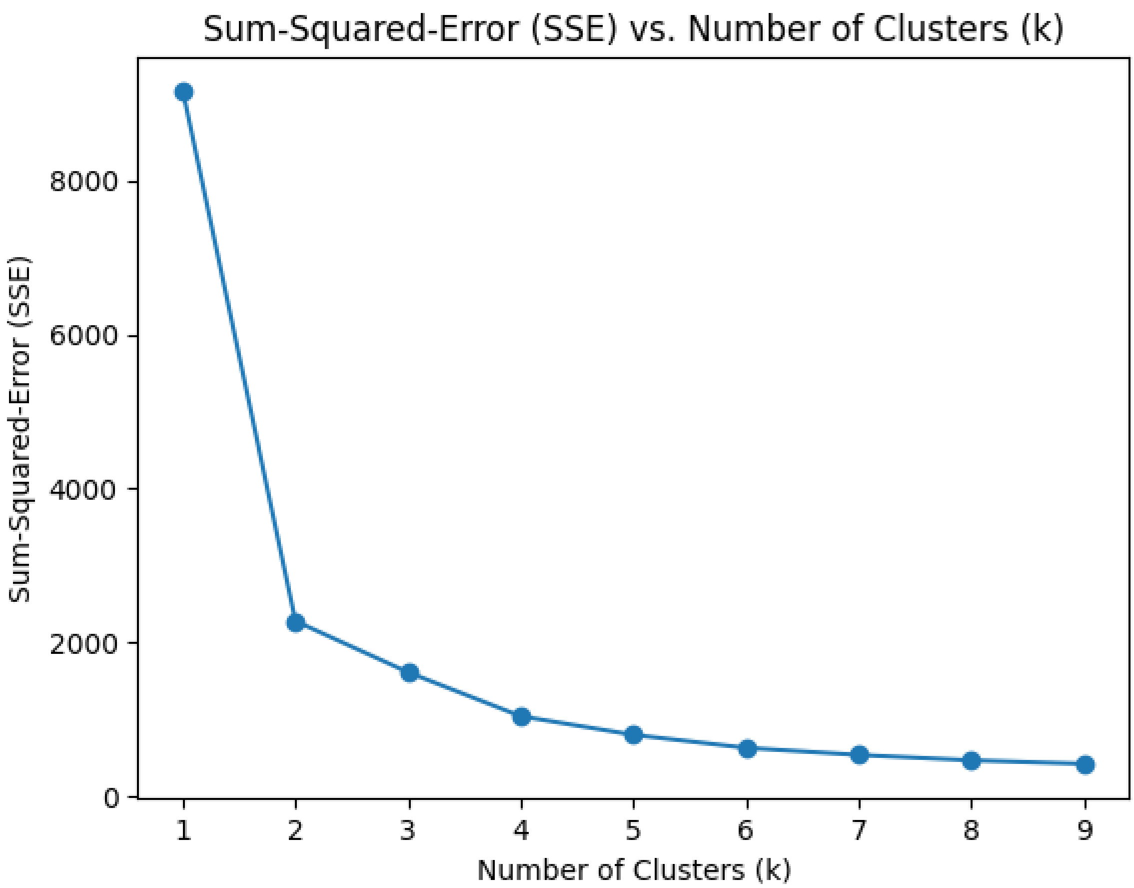

Figure 4. Sum Squared Error against Number of Clusters

Discussion of findings:

Referencing Figure 4, as we increase the number of clusters, we see that there is a trend of SSE decreasing. This makes sense, because as k increases, or we use more colors in an image, reconstructed images get closer to the display of the original image. In short, the above figure can provide information as the optimum number of clusters needed to create meaningful reconstruction of images.

The graph above can help to decide how many clusters we need to select, or the optimal value of k. Though k = 2 or k = 4 could both be arguable elbow points, as the decrease in SSE begins to become less linear past both points, because past k = 4, the model doesn't appear to make as significant contributions to lowering error, k = 4 will be considered the elbow point. When elaborating on this observation, we see that this selected k value aligns with the visuals shown in Figure 2. When k = 4, the general outlines and colors of objects of interest are laid out, and though details like uniform colors and field shading are not included, the reconstructed image contains enough information to convey to an observer that the image is of two different teams playing soccer on a field. Increasing K creates a more visually appealing image, but does not provide any additional information that could increase understanding of the image context.