CMPE 537 – Assignment 1

Oğuzhan Sevim

November 25, 2020

Part 1: Color Quantization

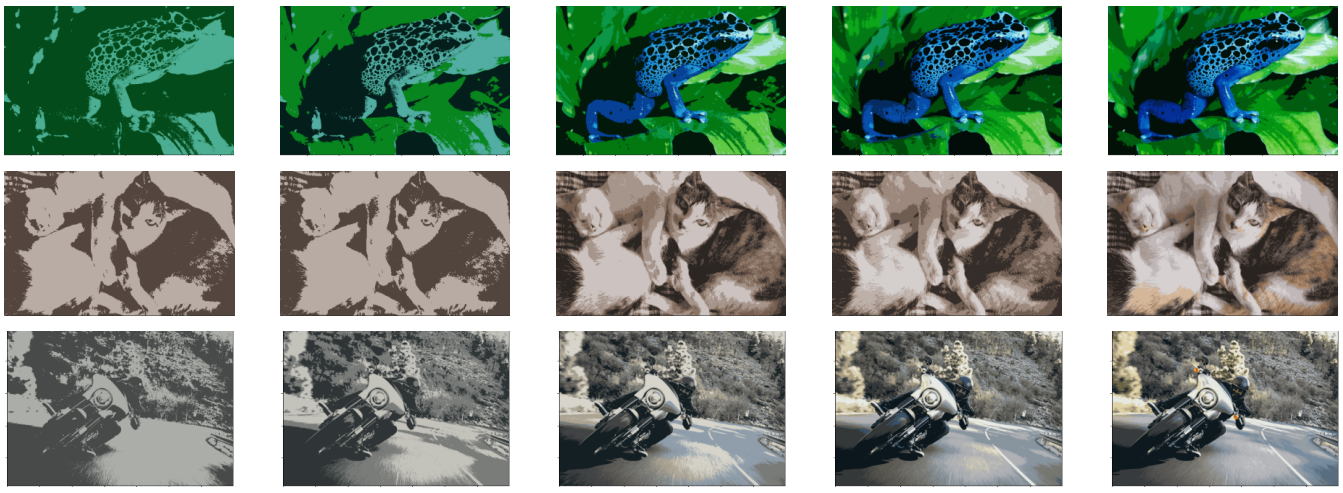
Color quantization is a process to compress a given image such that the resulting image remains visually similar to the input image while the number of bits required to represent it decreases. In a usual RGB image, each channel of a pixel is represented by an $8-bit$ number. By quantizing this image using 2^n colors, we can represent each channel of a pixel by using an $n-bits$.

K-means clustering algorithm is used for quantization. For a given image and a predefined number of quantization levels, each pixel of the given image is mapped to one of the quantization levels which is found by K-means algorithm. If we consider each pixel as a sample, a given RGB image with size (height$\times$width) can be considered as the collection of height*width points in $3D$ space of RGB values. By using a K-means clustering algorithm, we can assign each of the pixels into one of the predefined K categories.

Random and Manual Initialization

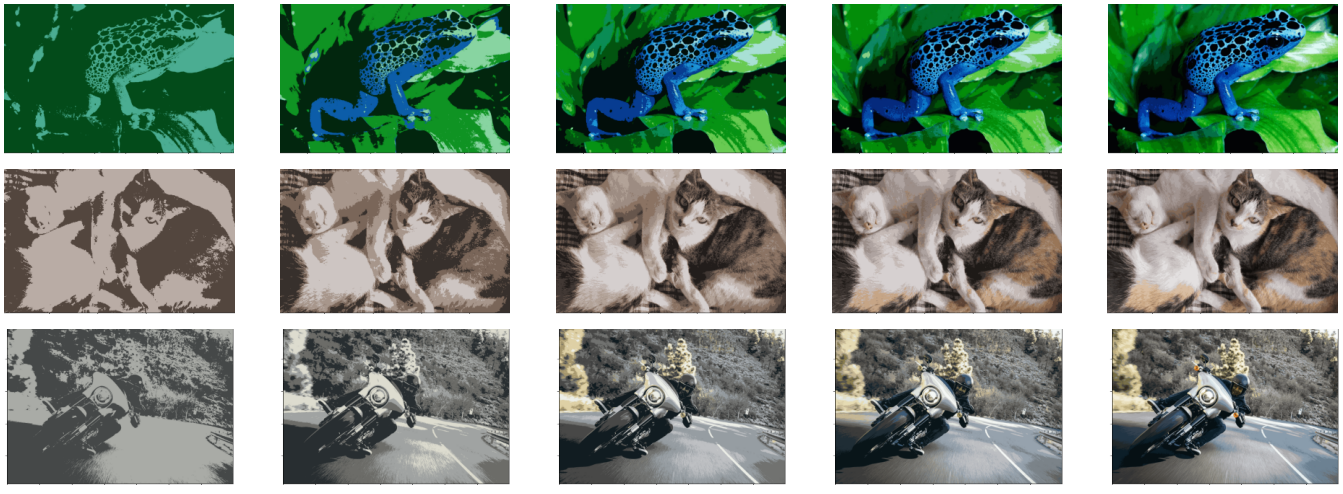
In this section, we present the results of different quantizations. For each of 3 different images, the quantization is implemented by using $2, 4, 8, 16$, and 32 colors levels. The number of iterations for K-means algorithm fixed to 5.

When the initial centroids of K-means algorithm are chosen randomly from a uniform distribution between $0-255$, the quantization results for each input image are illustrated in Figure 1. When the initialization is done manually (the centroids are tried to be chosen as distant as possible), quantization results for each input image are shown in Figure 2. For the low number of quantization levels, we observe that manual selection of initial centroids results in more distinctive (or close to original) results.



(k) 2 color quantized (l) 4 color quantized (m) 8 color quantized (n) 16 color quantized (o) 32 color quantized

Figure 1: Quantization results when the initial K-means centroids are chosen randomly.



(k) 2 color quantized (l) 4 color quantized (m) 8 color quantized (n) 16 color quantized (o) 32 color quantized

Figure 2: Quantization results when the initial K-means centroids are chosen manually.

Part 2: Connected Component Analysis

In this section, we implement connected component analysis (CCA) to label and count connected components in given images. The results of birds and dice images are given in Figures 3 and 4, respectively.

General CCA pipeline is as follows: First, we read the image in RGB format. If the image is in RGBA (the case in dice images) turn them into $0-255$ RGB images (this converter function is implemented in the script). Then, a threshold function is applied such that for an RGB image input we have a binary image where a pixel is 1 if all RGB channels of corresponding input pixel is larger than the predetermined threshold. Thresholds of 230 and 120 are used for birds and dice images respectively. After threshold block, the salt and pepper noise of the images are filtered by using a 3×3 median filter (this was the only major outsourced function). Finally, CCA is applied on the noise free images in order to get labeled images. The number of connected components are simply calculated by using a max operator on labeled image. Furthermore, a *labeled2RGB* function written to plot the labeled images with distinct colors for each component.

The code implemented finds the number of connected components correctly on all images. However, without filtering the noise, it was not possible. There were random black (or white) pixels in the images. I am not sure if it was allowable to use median filter, but it was quite easy to implement and gave very good results with 3×3 size on all the images.

Figure 3: Evolution of CCA pipeline on 3 birds images. From left to right, first column is the original images. Second column shows images after thresholding. Third is median filtered images. Finally, each labelled components are illustrated in last column.

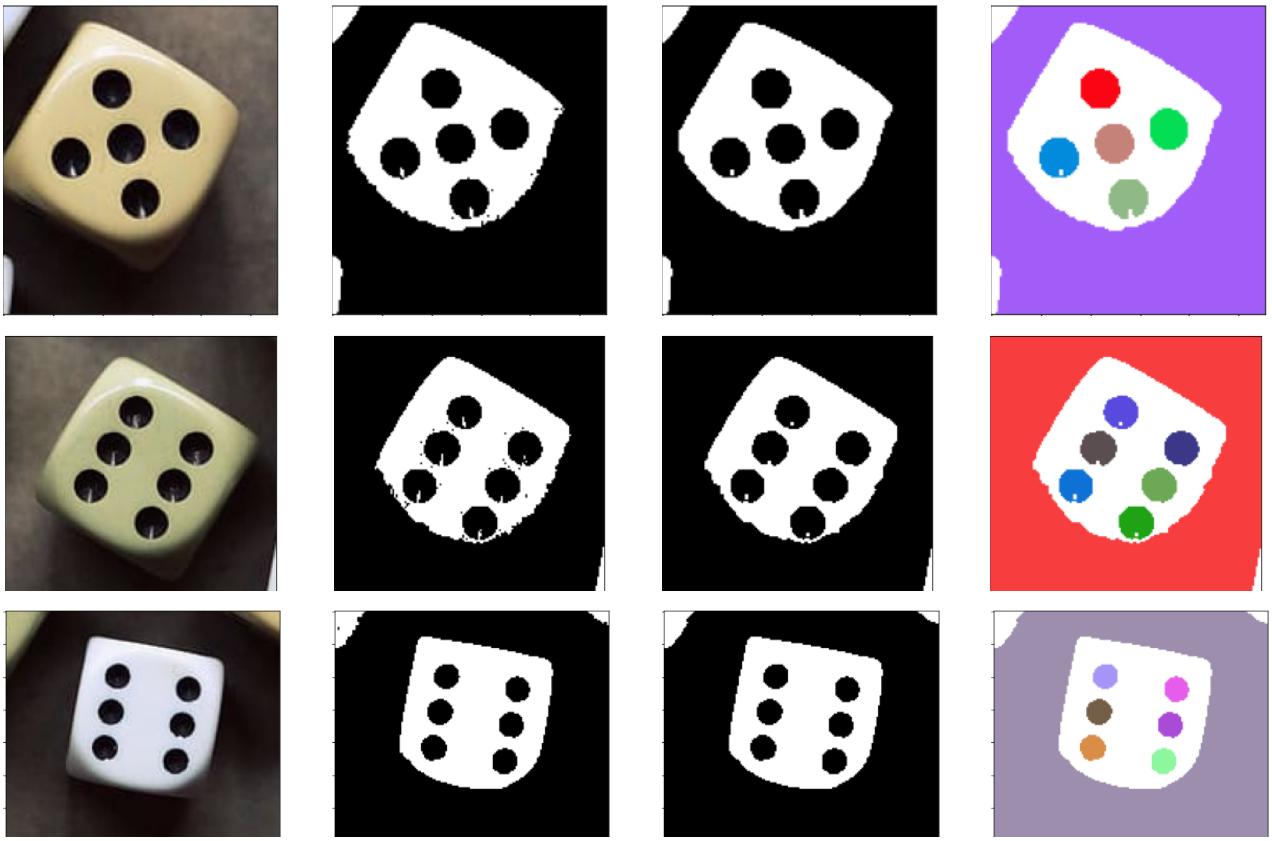


Figure 4: Evolution of CCA pipeline on 3 dice images. From left to right, first column is the original images. Second column shows images after thresholding. Third is median filtered images. Finally, each labelled components are illustrated in last column.