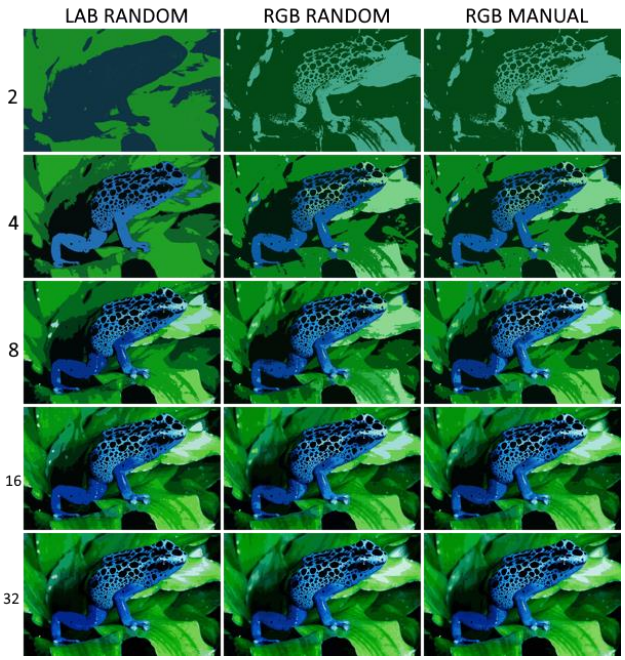
PART 1:



Methods:

pickPointsRandomly: Select random points from the image.

pickPointsManually: Select points from the image manually.

getColor: Return colors of given coordinates

calculateDistance: Calculate 3D Euclidian distance then return the number of the nearest cluster.

iterate: Return output of calculateDistance

findMean: Calculate the mean value of cluster.

createOutputImage: Create output image from mean colors, then save the image and return

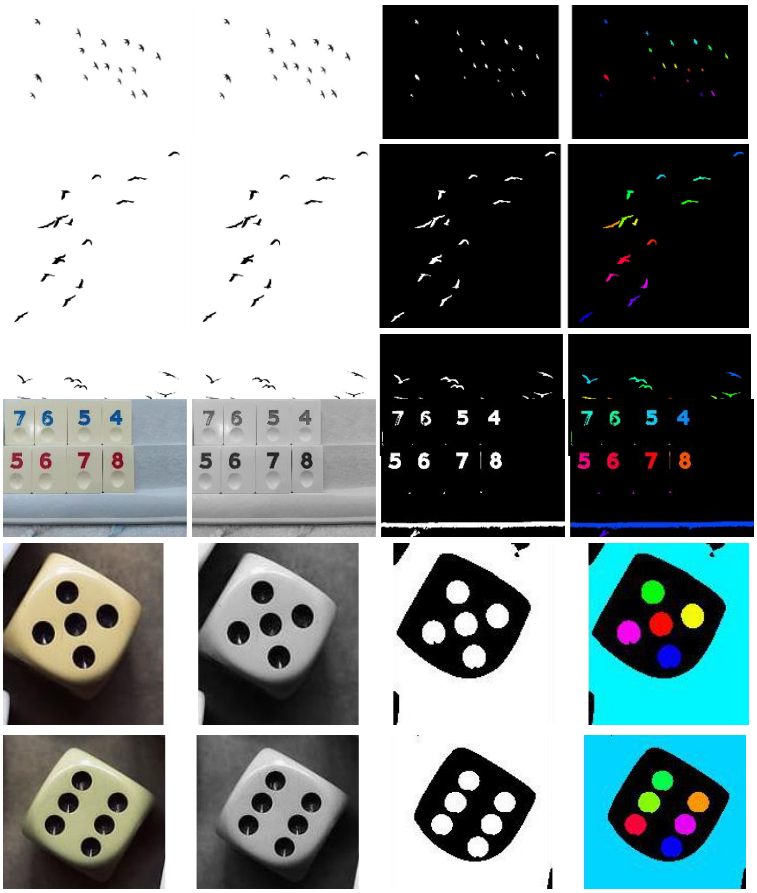
Algorithm: quantize takes number of color and a image then picks points randomly or manually. After point selection, get color values of points. Then, the last step is kMeans algorithm.

Bonus: LAB colorspace can be used by changing the mode variable as 'LAB' in the quantize function.

I found only RGB to LAB conversion on the internet. Therefore, I have to use skimage.color library for LAB to RGB

Conclusion: Quantized images become nearly similar to source images when K>=8. There is no significant difference between images except frog images. created by using all methods. (LAB Random – RGB Random – RGB Manual). Created image when we set K=2 and use LAB colorspace is different from images created using RGB.

Part 2:



(a) (b) (c) (d)

a: original image, b: grayscale image, c: binary image after thresholding, d: labeled image

First, I use the threshold function I wrote. This function use algorithm of the paper [1]. Detailed information is in the article. Then I used a median filter. However, results except for bird images are not meaningful. Therefore I decided to add another thresholding function with openCV. Both functions can be used.

Methods:

The main method is connectedComponentAnalysis. First, the image is converted to a grayscale image. Then, thresholding is operated. The output becomes an array with ones and zeros. After the array with 0's and 1's is obtained, connected component analysis is used. The number of label count is equal to the number of detected output + 1 (background). The main method returns the number of birds.

Conclusion: The function works perfectly for birds images. It returns 17, 14, 14 respectively, which are correct values. However, it returns 21 for okey board image. It was significantly high. I try to obtain the correct result but I can not be successful. It returns 6 and 7 respectively for dice images. These are one more of the correct values because there is two different background. One of them is the real background the other one is the dice surface. Therefore, the algorithm can not distinguish the one of background whether is object or not.

References:

[1] Taiyenjam, Romen & Roy, Sudipta & Imocha Singh, Oinam & Sinam, Tejmani & Singh, Khumanthem. (2012). A New Local Adaptive Thresholding Technique in Binarization. CoRR. abs/1201.5227.