Advanced Topics in Computer Vision - Lab 2

Leonardo Palacios Fidalgo   Ce Xu Zheng

April 20, 2022

Color Practice: Image Segmentation

The K-means algorithm aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. **The exercise demands that you implement your own algorithm for k-means clustering algorithm.**

1. Use the k-means algorithm to segment the 'papillary-carciroma.png' image (fig. 1) in the RGB color space. The idea is to locate the bluer areas corresponding to cells reacting to the Alcian blue colorant. Ideally the bluer areas must be segmented in separated compact regions (use morphology after pixel segmentation).

The original papillary carcinoma color image is shown in figure 1. As mentioned in the prompt, the goal is to segment the image by color in order to locate the bluer areas corresponding to cell reacting to the Alcian blue colorant. This segmentation can be achieved in a myriad of ways, one common approach is the K-means algorithm.

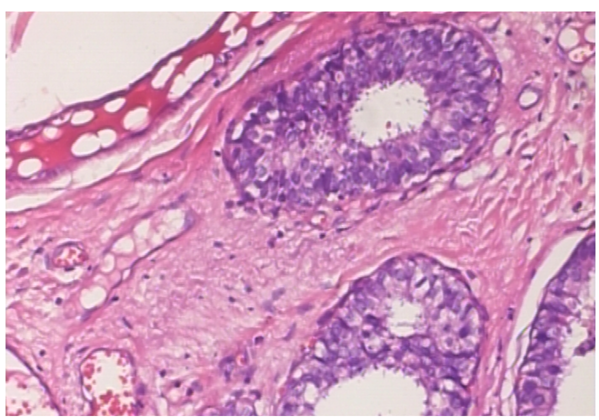

Figure 1: Papillary carciroma original color image.

The k-means algorithm has some design considerations worth mentioning: firstly, we must define the distance function used by the algorithm to determine the distance from a given pixel to a cluster distance. This distance function can be the euclidean distance between points, the mahalanobis distance between the points and a given color distribution, the city-block distance, chess distance, etc.

The utility of the k-means algorithm is dependent on the color space in which we apply the algorithm. For cyclic values such as angles, special attention must be done when implementing this algorithm, like we will see later in this practice. For the RGB color representation of a pixel, we can use the euclidean distance without an issue as the RGB values are not cyclic and the distance between two colors will be the euclidean distance between points inside the RGB color representation cube. But it is important to mention that computing the distance of colors does not make sense in most cases. Often the results of such a distance measure will be counter-intuitive. Colors that are "similar" according to your distance measure can appear "very different" to a viewer, and sometimes other colors, that have a large "distance" can be very close to our eyes. For this reason, we will study the results obtained when using different color spaces, distances measures and clusters in our algorithms.

First, lets begin by examining our image in the RGB color space. In figure 2 we can see the distribution of pixel colors in the RGB space for the image in 1. As we can see, colors are very close to the grey line and agglomerated together.

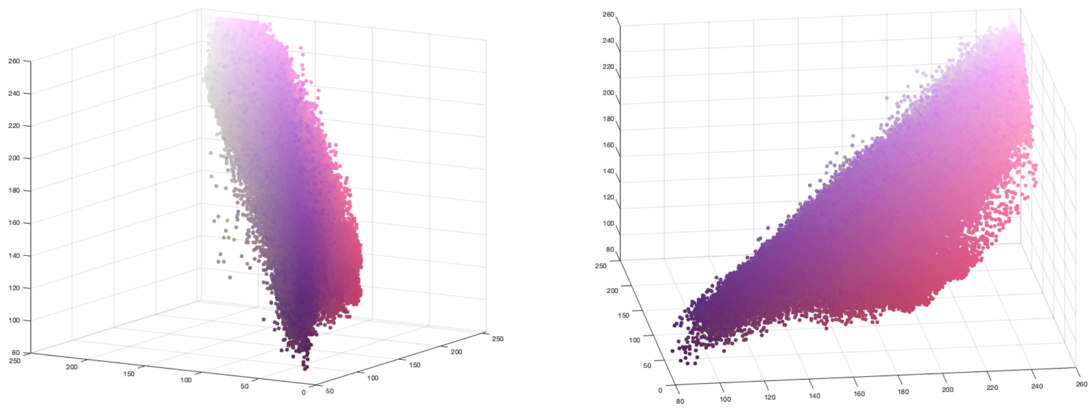

Figure 2: Pixel color distribution in RGB space.

The k-means algorithm was implemented on MATLAB, code can be found along with this document, and the following results were obtained using the RGB color space for k (number of clusters) equal to 3,4,5 and using the euclidean distance and the city block distance:

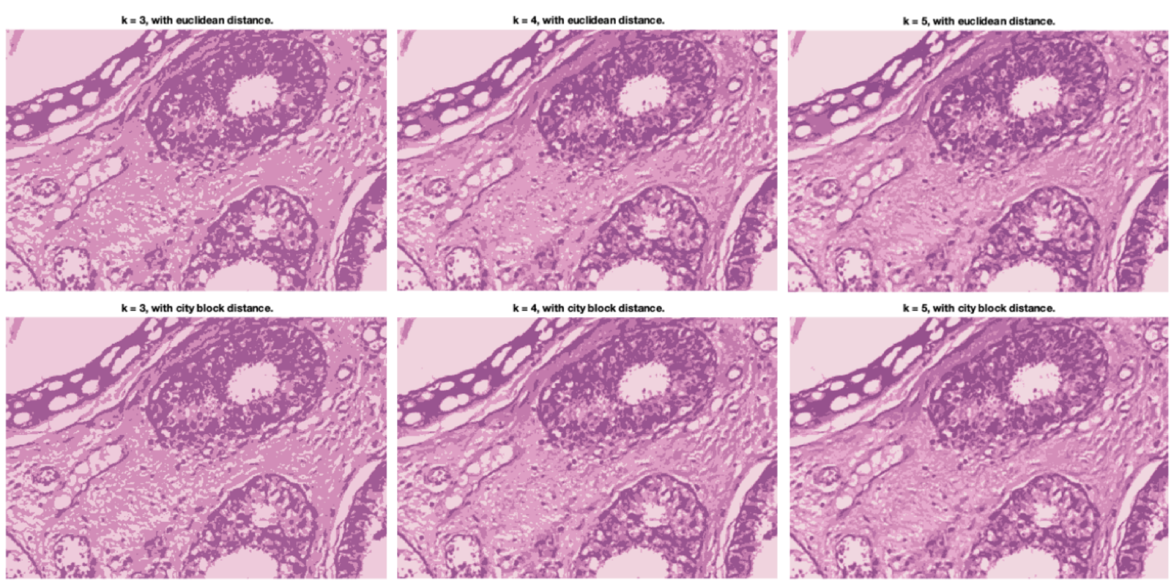

Figure 3: Papillary carciroma original color image.

As it can be seen, the city-block distance clustering failed at properly separating the redish colors with the purple/blue color we are trying to segmentate. Moreover, the best results were obtained for a k=5 clustering with euclidean distance. Note that we could determine the optimal number of clusters by evaluating for which number of clusters the interclass distance is maximized and the intraclass distance minimized, for this an information criterion could be explores, but for the scope of this analysis we will simply select 5 clusters.

Now, let's take a look at the segmentation achieved thus far:

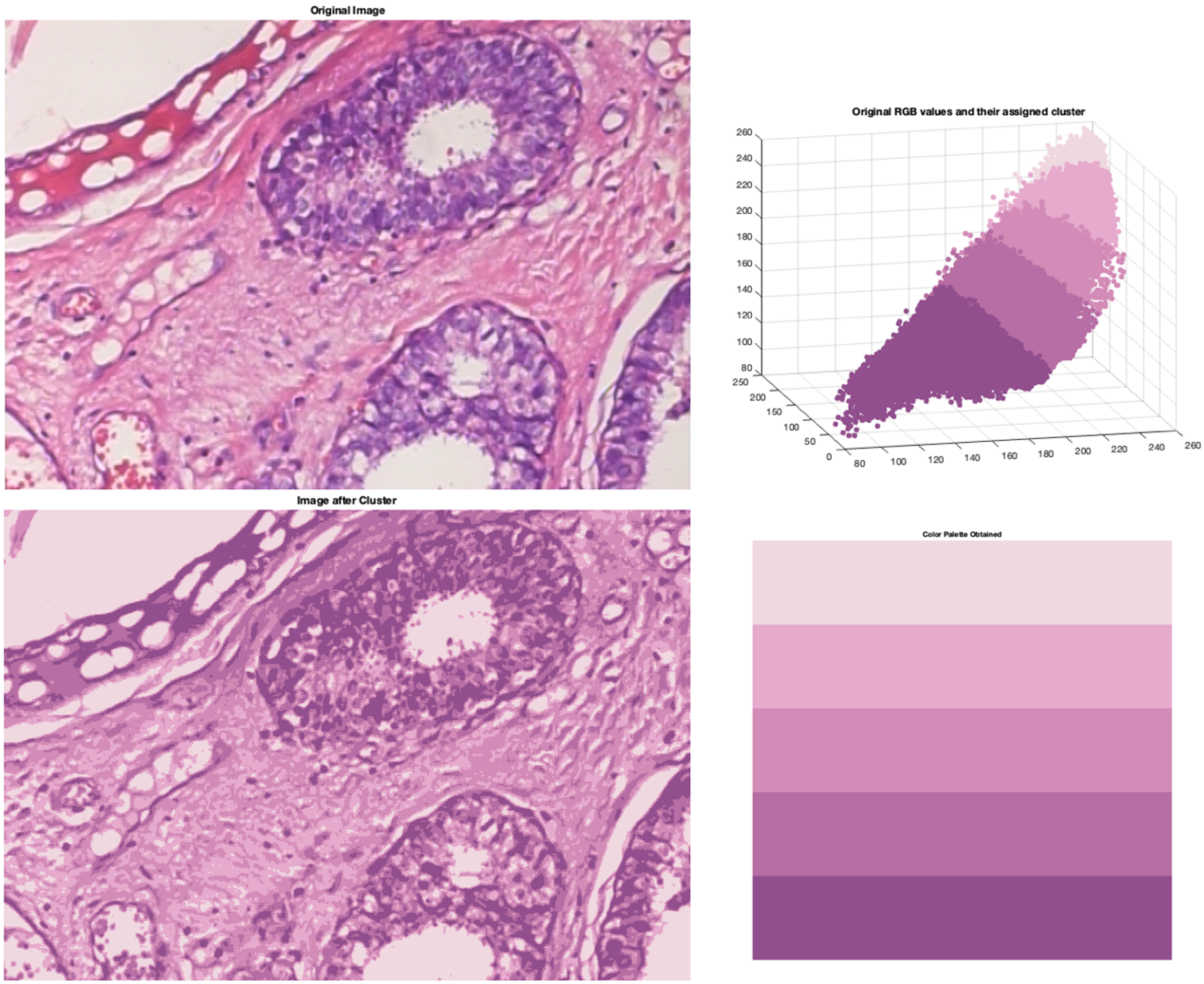

Figure 4: Clustering Results.

From these results, we can conclude that given the color distribution in the original image, clustering in the RGB space with the euclidean distance does not yield satisfactory results. Another distance measure should be devised to further divide colors in the purple/blue region, and maybe agglomerate colors that are not of interest into a larger cluster.

From these results, we can now proceed to select the cluster pertaining to the blue dye. The selected color will be the darkest purple shown in the color palette in figure 4. Note that this color is the result of averaging the color of the pixels that belong to this cluster. After determining the relevant cluster, we can proceed to binarize the image and apply morphological operators to obtain an image in which bluer areas are segmented in separated compact regions.

This method can be automated by defining the the *Alcian blue* color in the RGB space and compute the index of the color center with lower distance. Once we have chosen the cluster that contains the wanted color, we have to make some morphological transformations. First we will need to get rid of the smaller particles considered as noise or wrong detection with an opening of the image. After that we will make a few consecutive dilates of the remaining detection and finally a image filling.

Figure 5: From left to right, we have the initial pixels of the cluster, in the second image we can see the detection after the morphological operations and in the last image we can see the detection over the original image.

As we can see in Fig 5, we had a spare detection of the cluster and it also merged the cellular membrane, so we could not make an aggressive opening to get rid of the smaller blue spots in the plasma.

2. Try to segment the same image in the HSL and HSV color spaces, using k-means too. Comment how have you treated those pixels near white and black colors. Also those with a very low saturation. Comment how have you dealt with the cyclicity of the hue component.

Next, we will proceed with a similar color segmentation procedure using the implemented k-means algorithm. This time we will explore the results obtained when clustering in different color spaces, namely the Hue, Saturation and Value (HSV) and the Hue, Saturation and Lightness (HSL) color models.

HSL and HSV are alternative representations of the RGB color model (shown in figure 6), designed in the 1970s by computer graphics researchers to more closely align with the way human vision perceives color-making attributes. In these models, colors of each hue are arranged in a radial slice, around a central axis of neutral colors which ranges from black at the bottom to white at the top. For this reason, the cyclical nature of the Hue value will represent a challenging obstacle to overcome in the clustering procedure.

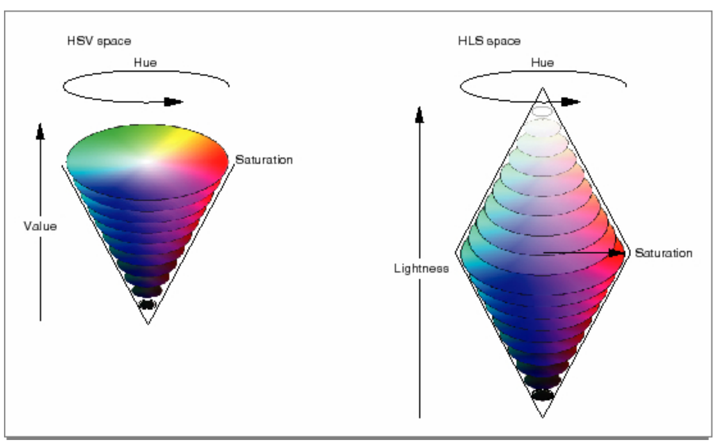

Figure 6: HSL and HSV double cone and cone model representations.

Given the nature of the values used to represent saturation in most common representations, a more realistic depiction of these color models is a cylindircal model and shown in figure 7.

Notice that even though these models were created to more closely align with the way human vision perceives color-making attributes, the nature of these model's definitions represent a challenging obstacle for determining the distance measures used for clustering in the k-means algorithm. For

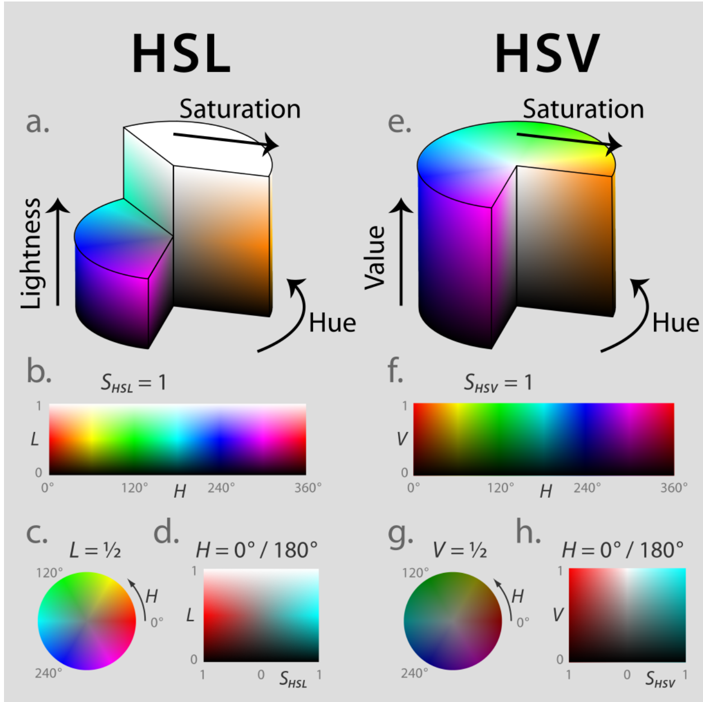

Figure 7: HSL and HSV cylindrical representations.

example, notice that when lightness in the HSL model is either very close to 1 (white) or 0 (black), the distance between colors can differ greatly even when the two colors will be very close to white/black and hence similar to our eyes. In other words, they should be clustered together but their distance might make it harder to do so. A similar situation occurs with the 0 Value in the HSV model, where all color will be black regardless of Hue and Saturation. Moreover, notice that for both models the grey line lies when saturation equals to zero. When this is the case, the Hue value can change greatly and the color will be the same, a challenging design consideration for the k-means algorithm. Finally, notice that a value of Hue = 1 is the same as Hue = 0, due to the cyclic nature of the angular value. Hence, special care must be done to cluster this values together, and when determining the average hue of a cluster find a way to take this property into account.

To mitigate the aforementioned issues, for the HSL color model we will use the following distance measure, as introduced by Patrascu, Vasile in [1]:

$$D_P^2\left(Q_1, Q_2\right) = \sqrt{S_1 \cdot S_2} \sin^2\left(\frac{H_1 - H_2}{2}\right) + \sqrt{(1 - S_1) \cdot (1 - S_2)} \cdot (L_1 - L_2)^2 + (S_1 - S_2)^2 \tag{1}$$

The equation in 1 is derived [1] from the idea that the colour distance should have three terms: the distance between hues, the distance between saturations and the distance between luminosities, and that these should have different weights depending on where on the cylinder we are at. The distance between hues is multiplied by a factor that depends on the colour saturations. This factor has a multiplicative structure. Thus, when the saturation values increase, the hues distance influence increases in framework of distance DE. When the saturation values decrease, the hue distance influence decreases. From here, the idea to multiply the luminosity distance with a similar factor. This factor will have the following behaviour: when the saturation values increase, the luminosity distance influence decreases and when the saturation values decrease, the luminosity distance influence increases.

Finally, for the HSL color model, the mean function for the algorithm for determining new cluster center will be the mean of the the values of Saturation and Luminosity, and for the Hue values the atan2 function is calculated over the mean cosines and mean sines of the hue values of the cluster points.

For the HSV model, a similar approach is taken, the following distance formula is implemented:

$$
\begin{aligned}
dh &= min(abs(h1 - h0), 360 - abs(h1 - h0))/180.0 \\
ds &= abs(s1 - s0) \\
dv &= abs(v1 - v0)/255.0
\end{aligned} \tag{2}
$$

$$distance = sqrt(dh * dh + ds * ds + dv * dv)$$

Finally, for the HSV color model, the mean function for the algorithm for determining new cluster center will also be the mean of the the values of Saturation and Luminosity, and for the Hue values the atan2 function is calculated over the mean cosines and mean sines of the hue values of the cluster points.

The proposed distance and mean functions manage to deal with pixels near white and black colors in these color model spaces. Moreover, the also eliminate the problems that arise from the cyclicity of the hue component.

The proposed solutions where tested on the image in 1 for k (number of cluster) equal to 3, 4, and 5. The results are shown below:

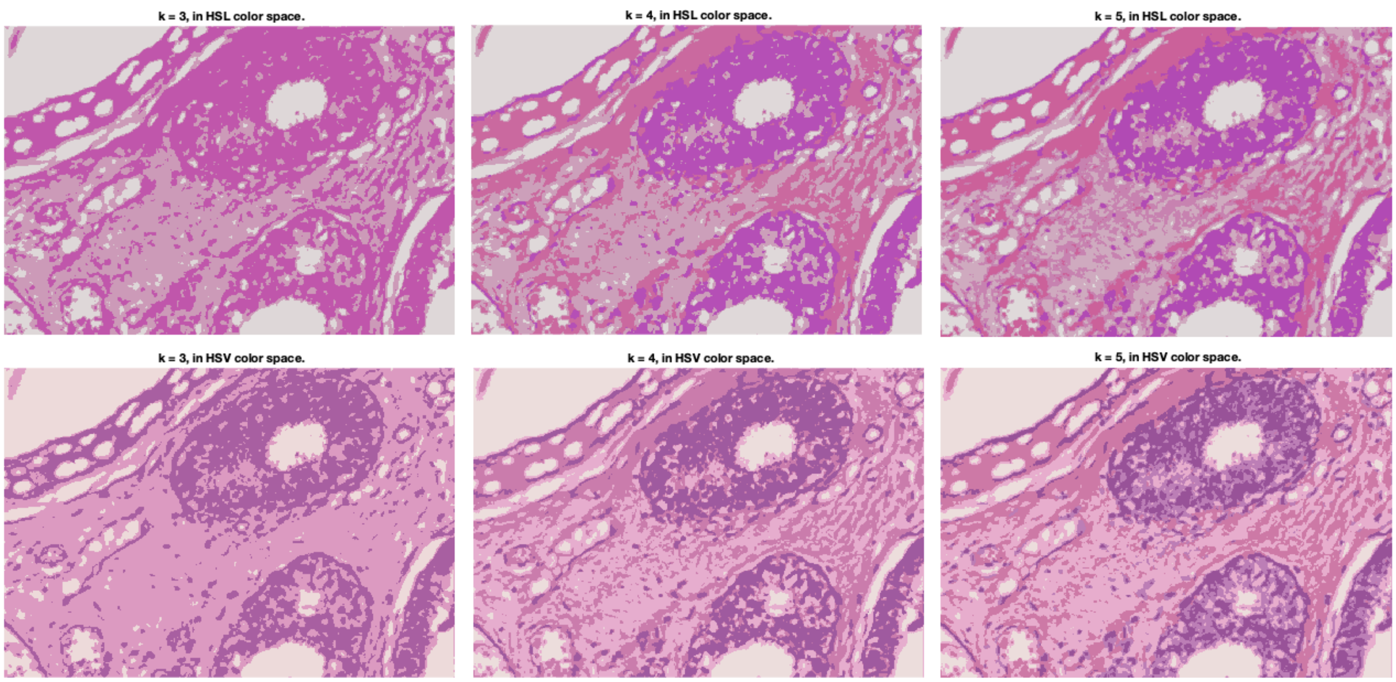

Figure 8: HSL-HSV clustering results for different ks.

As it can be seen, the color segmentation achieved is slighlty different for both models. Both color spaces achieve acceptable results with k=4,5. Though k=4 seems to work sufficiently well without any further color segmentation, we will continue with k=5 for a fairer comparison with the RGB model obtained. Moreover, we see that the distance functions and mean function used by the algorithm to deal with pixels close to white and black, and the problem with the hue cyclicity functions properly.

Having selected the desired number of clusters, we now present in further detail the results obtained with the k-means algorithm working on the HSL space:

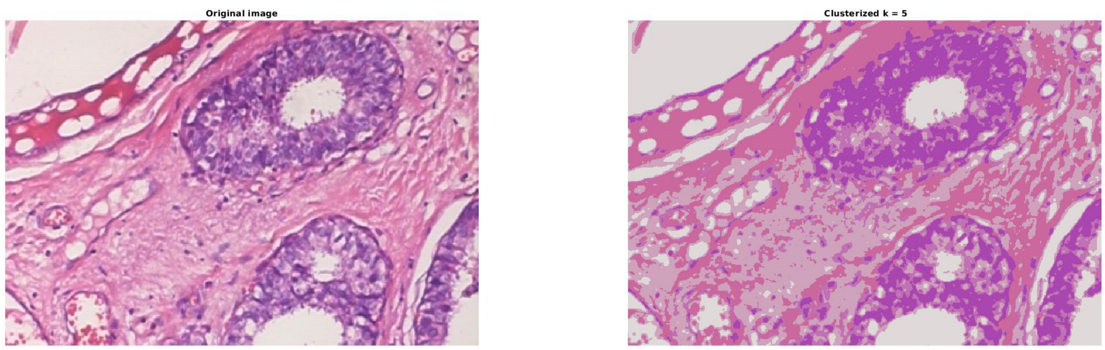

Figure 9: HSL Clustering Results.

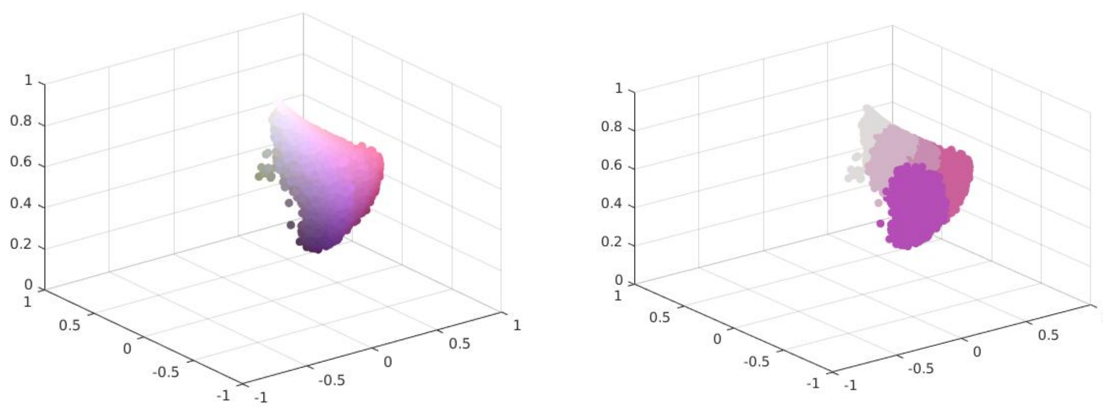

Figure 10: Pixel color distribution of the HSL results in RGB space.

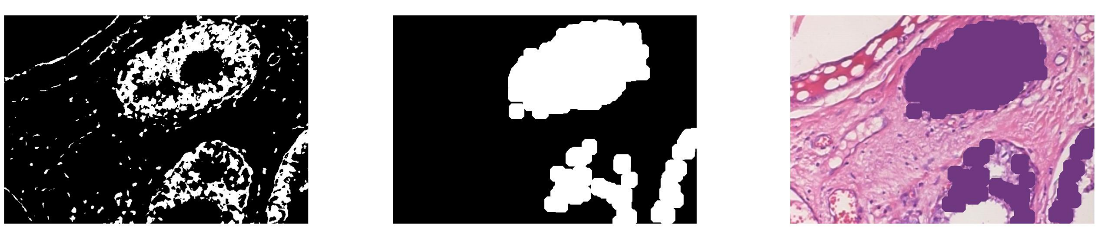

Figure 11: From left to right, we have the initial pixels of the cluster, in the second image we can see the detection after the morphological operations and in the last image we can see the detection over the original image.

Next, we present in further detail the results obtained with the k-means algorithm working on the HSL space:

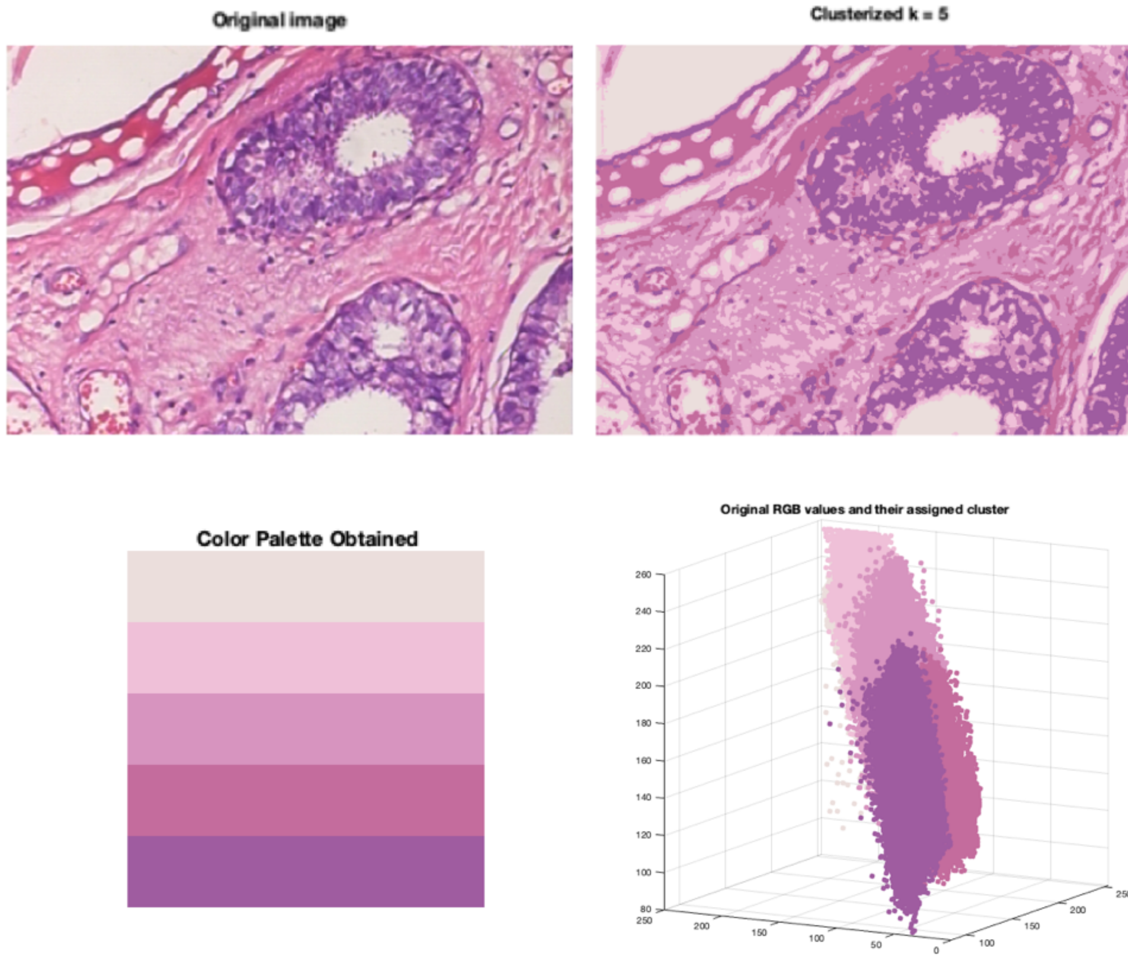

Figure 12: HSV color segmentation results.

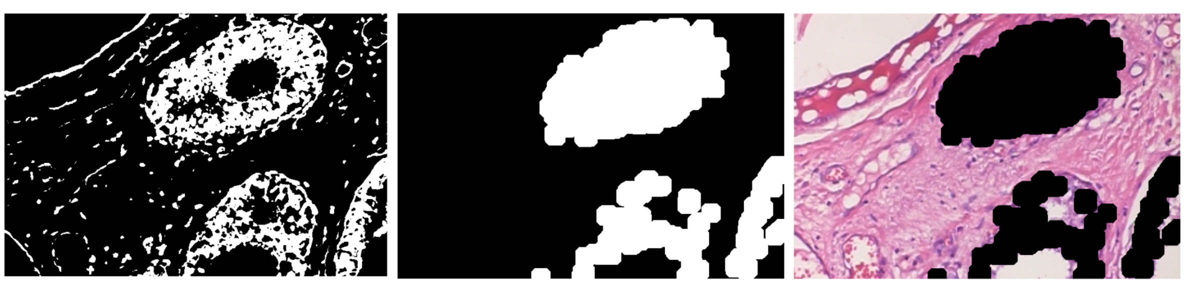

Figure 13: From left to right, we have the initial pixels of the cluster, in the second image we can see the detection after the morphological operations and in the last image we can see the detection over the original image.

3. Finally use the Lab color space. Does this space help to deal with the issues of HSL/HSV color spaces?

The issues of HSL/HSV color spaces inspired the International Commission on Illuminationto to design a color space that is also closely aligned with how human vision perceives color-making attributes but solves the aforementioned issues with HSL/HSV color spaces.

CIELAB is designed to approximate human vision. The L* component closely matches human perception of lightness. CIELAB is less uniform in the color axes, but is useful for predicting small differences in color. The a* and b* components relate to the four unique colors of human vision: red, green, blue, and yellow. They can be positive or negative.

Since the CIELAB color space was created as a 3D matrix of color points. We can use the traditional euclidean distance, moreover we can also determine the mean of a set of values in the traditional sense while avoiding the problems presented by the HSV,HSL color spaces. In recent years, more accurate distance formulas have been proposed as it can be found on [2], for our purposes we will use the euclidean distance between the colors space coordinates. The CIELAB color space model is shown in figure 14.

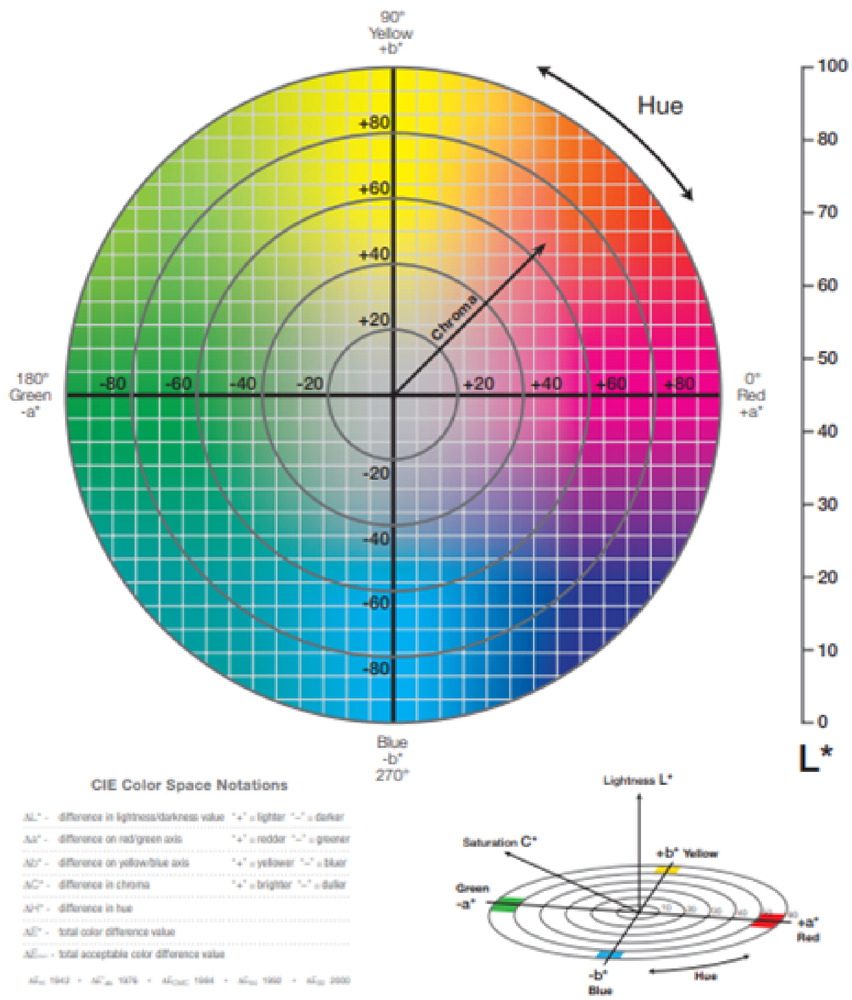

Figure 14: CIE L*A*B* color space.

The k-means algorithm was implemented using the CIE LAB color spaces with k (number of clusters) = 3,4,5 and the results presented below:

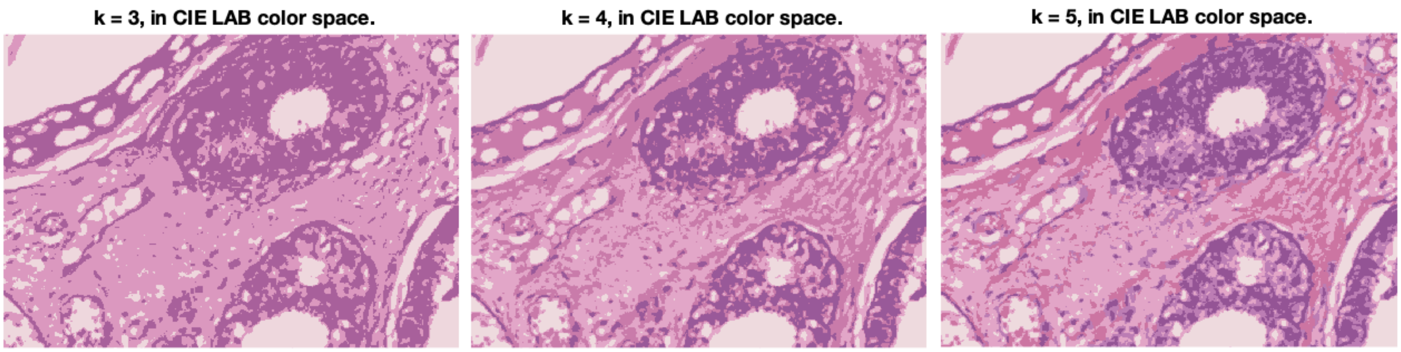

Figure 15: CIE-LAB clustering results for different ks.

As it can be seen, the LAB color space achieves acceptable results with k=4,5, even with a simple distance and mean functions. Though k=4 seems to work sufficiently well without any further color segmentation, we will continue with k=5 for a fairer comparison with the previous color models.

Having selected the desired number of clusters, we now present in further detail the results obtained with the k-means algorithm working on the HSL space:

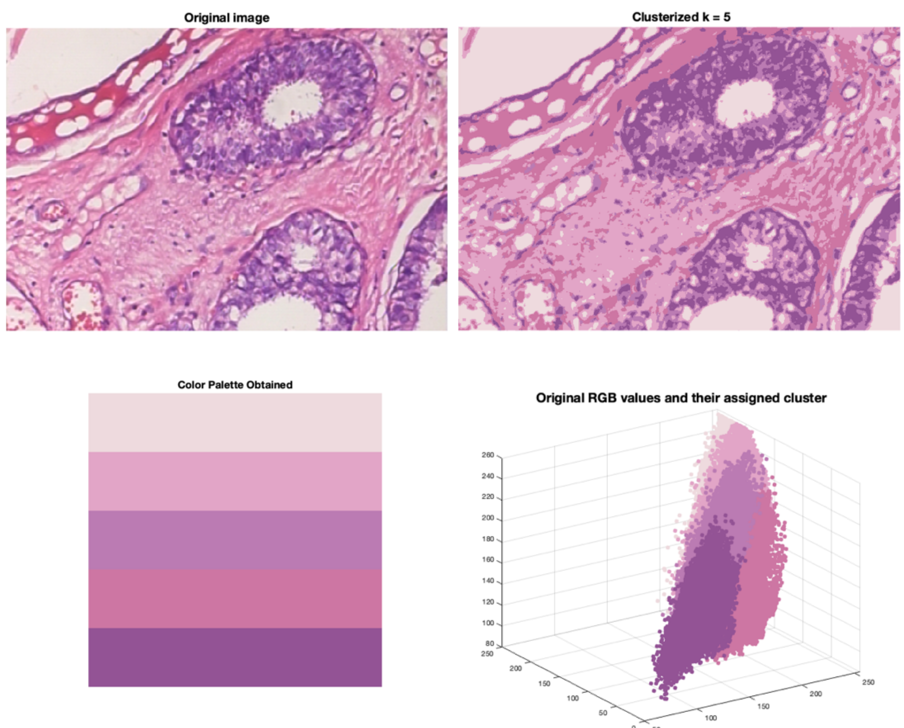

Figure 16: CIE L*A*B* color space clustering results.

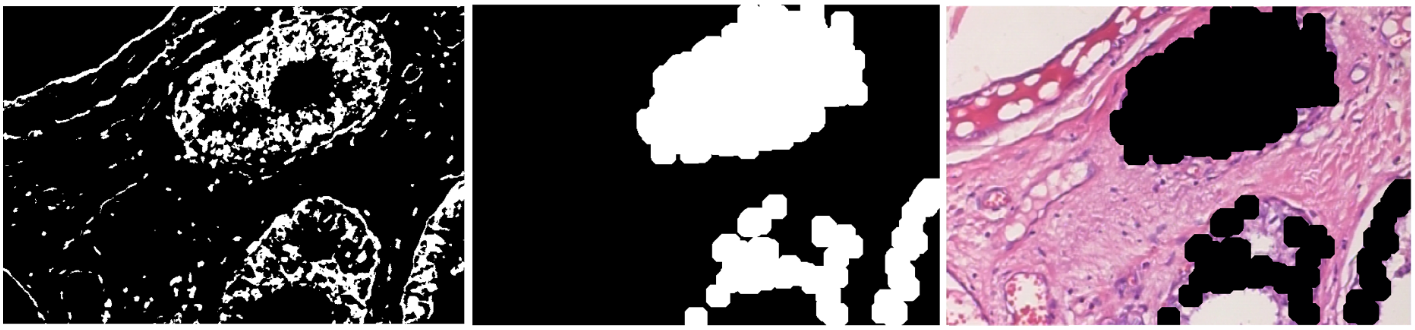

Figure 17: From left to right, we have the initial pixels of the cluster, in the second image we can see the detection after the morphological operations and in the last image we can see the detection over the original image.

As we can see from figure 16 and figure 17, the LAB color space successfully segments the image and after some morphological operations we can extract the location of the cells reacting to the dye. Similar to the HSV and HSL implementations, the color segmentation around the dye color is done more accurately than the RGB implementation. In the next section we will compare the results across the different color spaces.

4. Compare the results of the segmentation in the different color spaces. Try different number of clusters. Test your results using the different color spaces in a new set of images: tucan1, peppers, flowers, ... (without trying to segment specific compact areas).

In this section we will compare our implementations for each color space for the original image, and for a different set of images. First, let's take a look at how these color spaces compare to each other when segmenting the papillary carcinoma image for 5 clusters, as it was selected before:

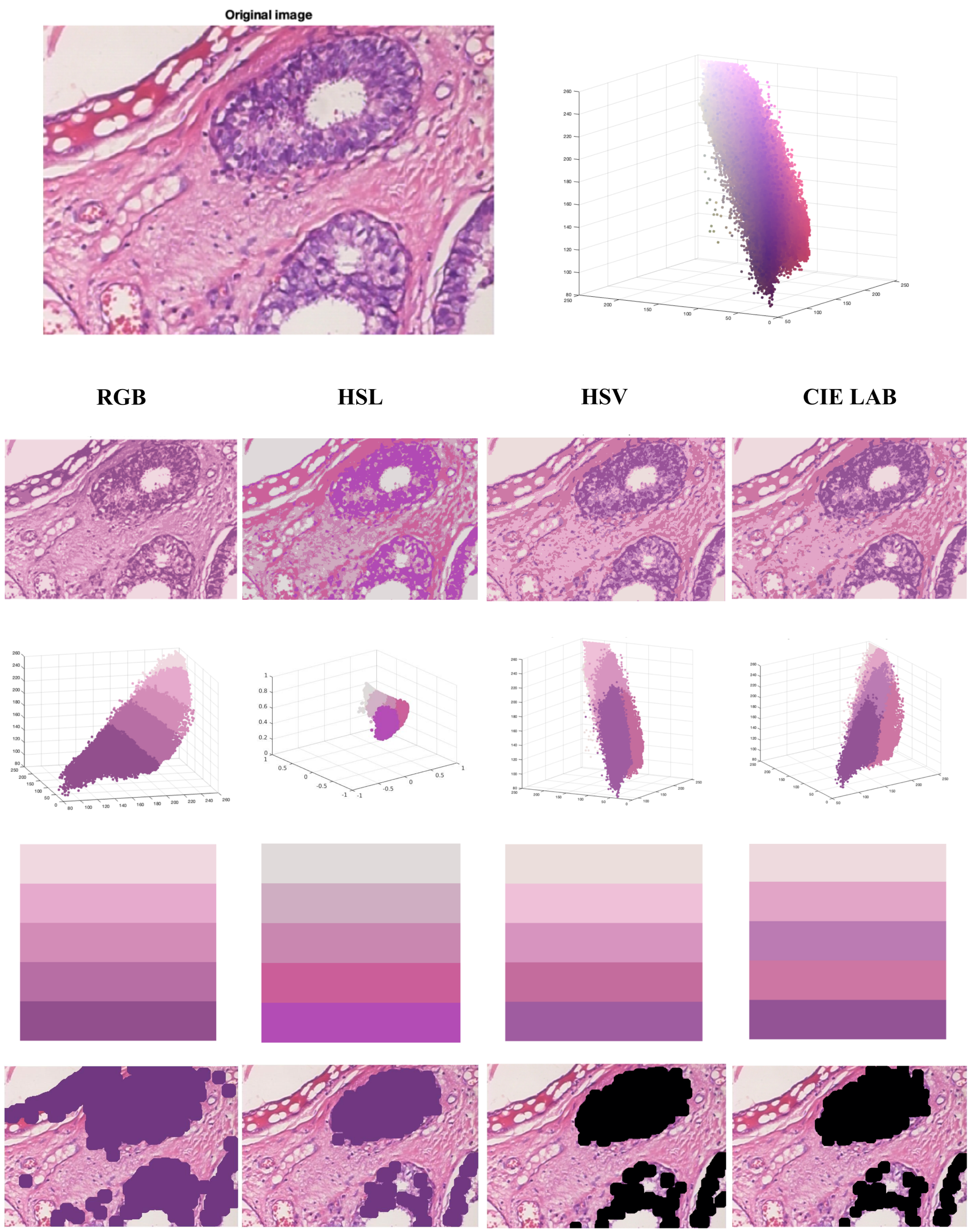

Figure 18: Segmentation results.

As we can see from figure 18, all color space implementations achieve relatively successful results. We can see the HSL color model performs better at clustering points close to the red color. The RGB color space struggles with the red color given the close distribution of colors in the image, hence it fails the most during the cell segmentation phase, even though it managed to close the section on one of the cells. For our purpose, the CIE Lab yields the best results, albeit morphological operations must be improved.

For this exercise we will make four clusters for each image in the dataset and each one of the color spaces.

Figure 19: Tucan original image.

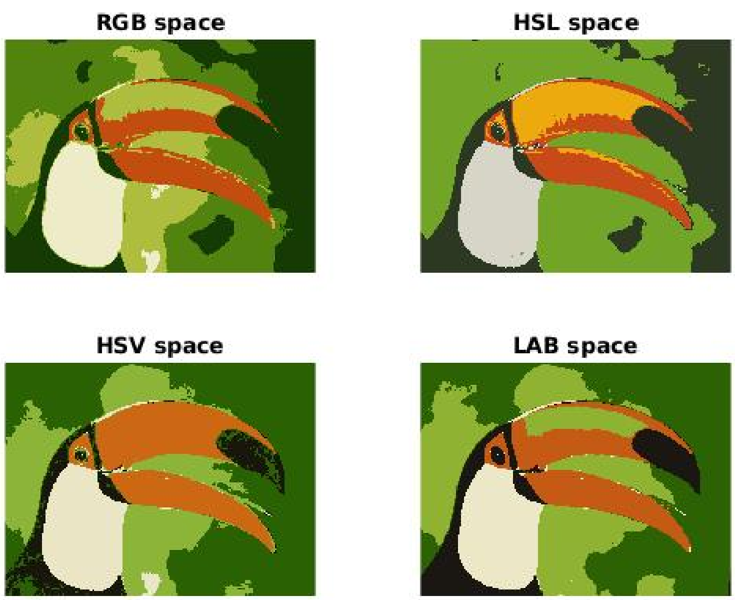

Figure 20: Tucan color segmentation with 5 clusters.

In the Tucan image we can see that the green background, the black feathers, the white feathers and the orange beak. However, if we only choose four clusters, the beak and part of the background clear green is merged in the HSV and HSL spaces.

Figure 21: Flowers original image.

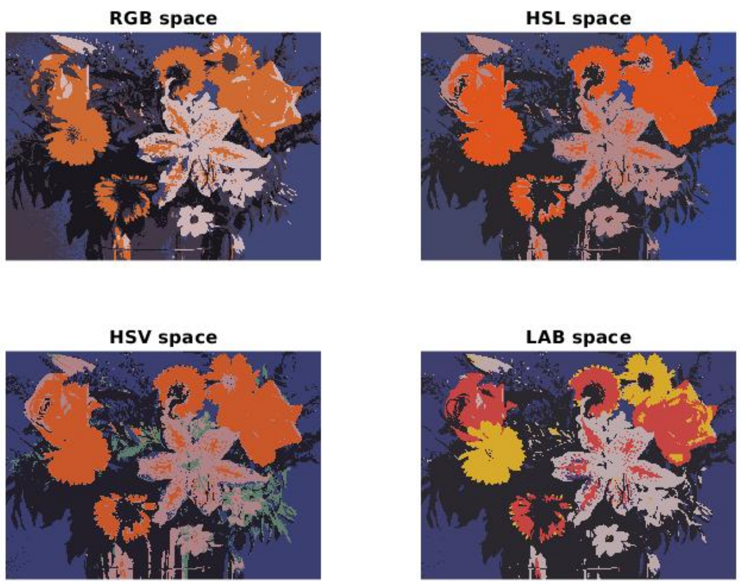

Figure 22: Flowers color segmentation with 5 clusters.

In the flowers picture, we can see that the colours that are more sensible for the human eye like the difference between the orange and the yellow or the relevance of the white flower is represented in the CIE LAB space.

Figure 23: Peppers original image.

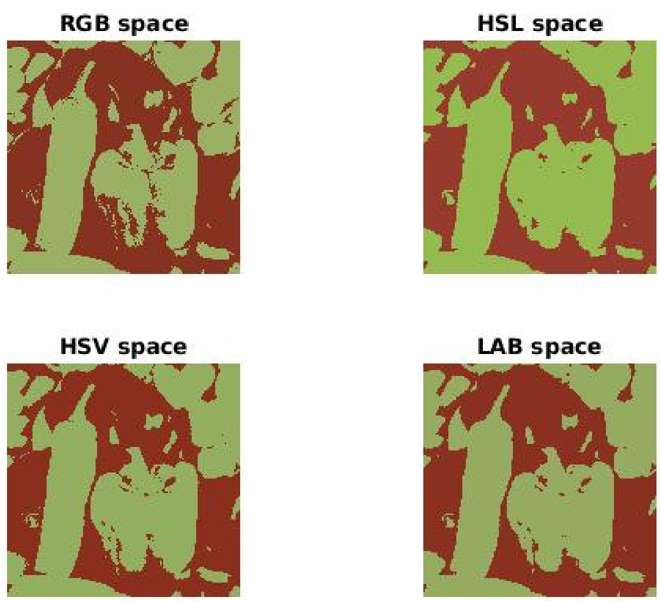

Figure 24: Peppers color segmentation with 2 clusters.

In this last case, we have chosen only two clusters as we previously know that there are two main types of peppers. In this case the segmentation is pretty goo defined except for the shadows.

As we can see, all the color clusterization methods works pretty good at simplifying the number of needed colors, and as we increase the number of clusters, the color representation is also more accurate.

References

[1] Patrascu, Vasile. (2013). New HSL Distance Based Colour Clustering Algorithm. 10.13140/2.1.4990.8007.

[2] Delta E 101.(2016). http://zschuessler.github.io/DeltaE/learn/