UE : BIOMED

Digital pathology

Yuxuan XI
Mojan OMIDI

SCIENCES SORBONNE UNIVERSITÉ

20 February 2024

Contents

1 Color Quantization:

- Extract the cancerous cell region in yellow of the Lung.png image

Here, we use the color quantization to extract the yellow part in the image.

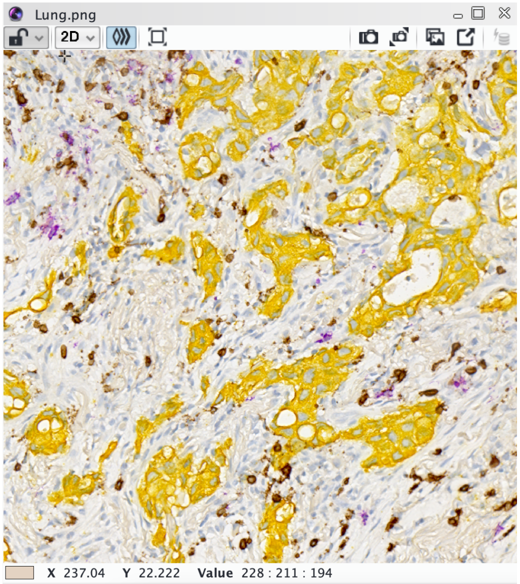


Figure 1: Lung image

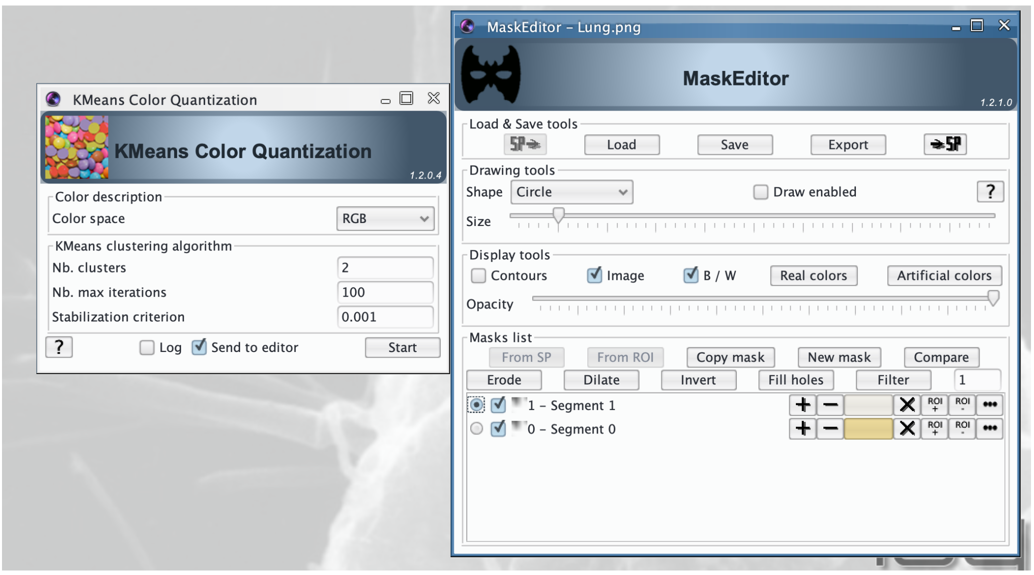


Figure 2: Kmeans Color Quantization

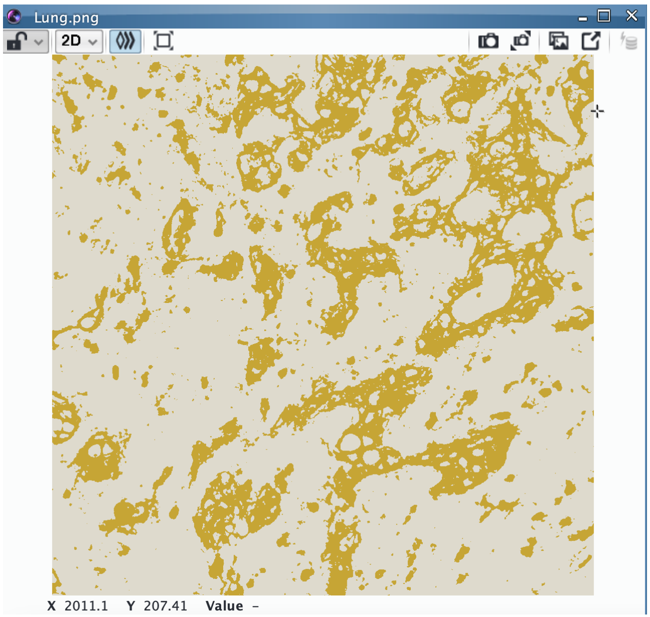


Figure 3: Yellow part of the image

2 Color Separation

- Separate the dye staining of the following images: he.png hDAB.png in order to count the nuclei

We want to extract the nuclei part from the image and count it. So we want to use "Color Separation", "Color Deconvolution" or "Stain Separation" to separate the dye staining of the image. However, we didn't find any related plugin in ICY software.

Q1: Count the brown spot. Explain your approach

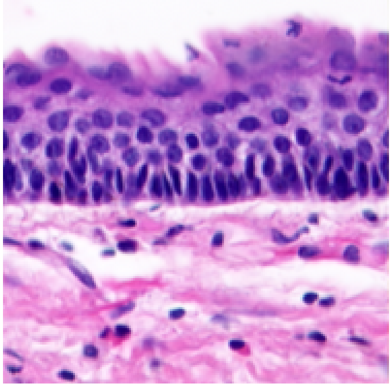
Color quantization is a process of reducing the number of distinct colors in an image while preserving its overall visual appearance. It's often used to reduce the memory usage and computational complexity of image processing algorithms, as well as for reducing the size of images for web or mobile applications.

KMeans clustering is a popular unsupervised machine learning algorithm used for clustering data points into groups based on similarity. In the context of color quantization, KMeans clustering can be applied to group similar colors together and reduce the number of distinct colors in an image while preserving its visual quality.

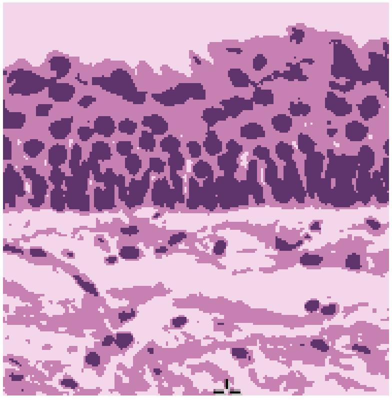
In the KMeans algorithm, the number of clusters defines how many groups the algorithm will try to create. In the context of color quantization, the number of clusters determines how many distinct colors will remain in the quantized image. Choosing the optimal number of clusters often involves experimentation and evaluation of visual quality.

We apply KMeans Color Quantization with a specified number of clusters set to 3. The choice of 3 clusters is because we believe it yields the best results in terms of reducing the number of colors in the image while still maintaining its visual quality.

Below we can see the results:



(a) Original Image



(b) Image 2

Figure 4: Color Clustered Image

Now the aim is to segment regions of interest within an image. To achieve this, the default colors assigned to different segments need to be reconfigured. This typically involves adjusting the color thresholds or criteria used to define the boundaries between different regions.

Once the default colors are reconfigured, the result is a binary mask.

Reconfiguring the default colors for segments is often done through techniques such as thresholding, where pixels are classified as belonging to a particular segment based on their color values falling within a certain range or meeting specific criteria. This process helps distinguish between different regions or objects in the image and facilitates further analysis or processing.

Below we can see the resulted binary mask:

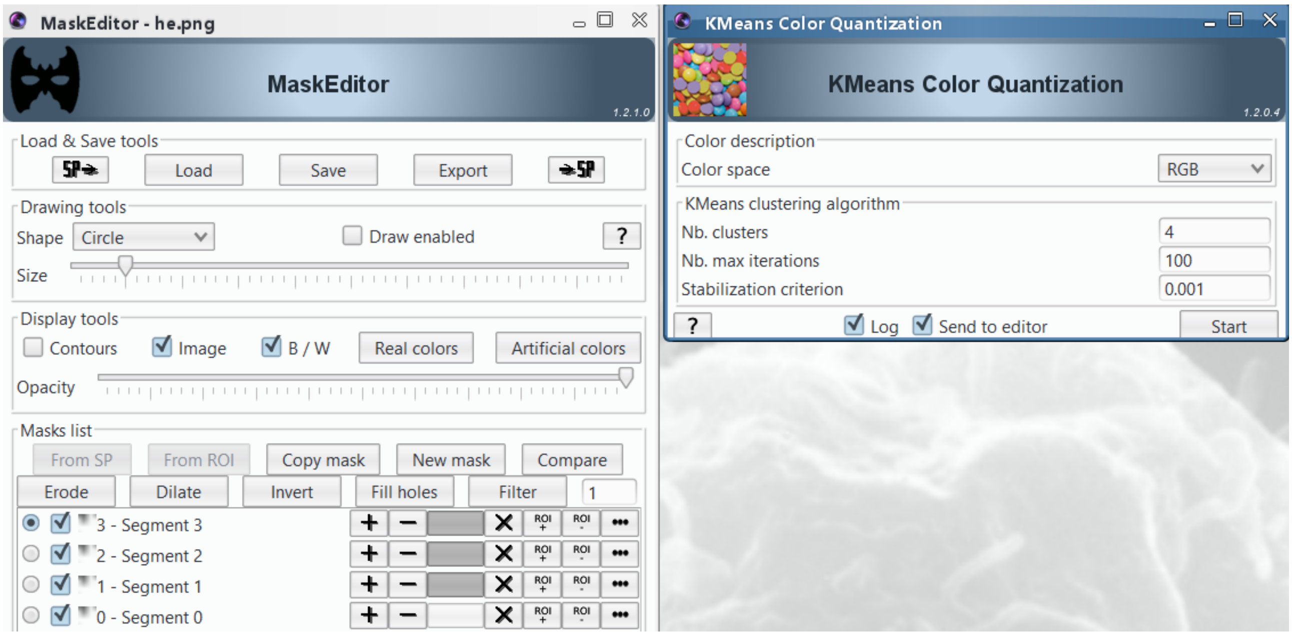


Figure 5: Reconfiguration

The result is shown below:

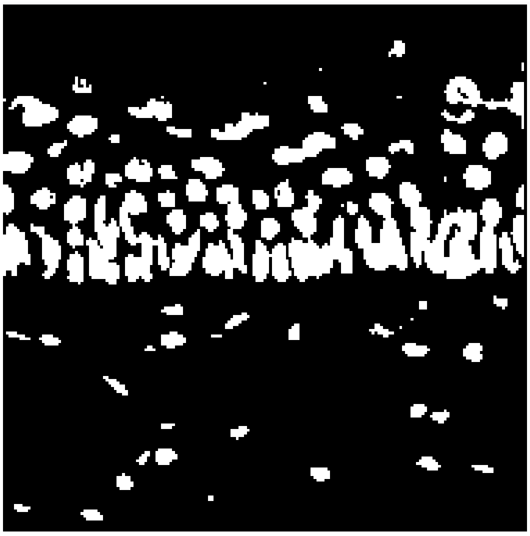


Figure 6: Reconfiguration

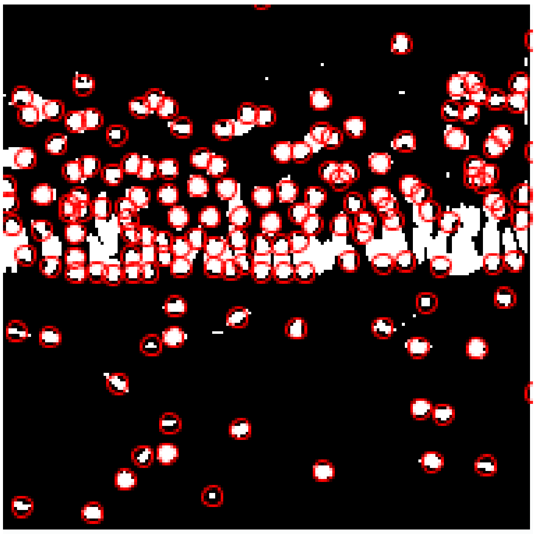


Figure 7: Detected Spots

Q2: segment the image test_001.tif with DeepIcy (installation: Gson library Snake yaml, Msgpack core

To use deep learning for segmentation we install the DeepClas4Bio API.