In the previous video we identified clusters, or tissue substances, in a healthy brain image.

It would be really helpful if we can use these clusters to automatically detect tumors in MRI images of sick patients.

The tumor.csv file corresponds to an MRI brain image of a patient with oligodendroglioma, a tumor that commonly occurs in the front lobe of the brain.

Since brain biopsy is the only definite diagnosis of this tumor, MRI guidance is key in determining its location and geometry.

Now, make sure that the tumor.csv file is in your current directory.

Let's go to the console, and clear it, and then read the data, and save it to a data frame that we're going to call tumor, and use the read.csv function, which takes as an input the tumor dataset, and make sure to turn off the header using header equals FALSE.

And let's quickly create our tumorMatrix, using the as.matrix function over the tumor data frame, and the tumorVector, using the as.vector function over the tumorMatrix.

Now, we will not run the k-means algorithm again on the tumor vector.

Instead, we will apply the k-means clustering results that we found using the healthy brain image on the tumor vector.

In other words, we treat the healthy vector as training set and the tumor vector as a testing set.

To do this, we first need to install a new package that is called flexclust.

So let us type install.packages("flexclus").

And then the first thing R will ask us is to set the region that is closest to our geographical location.

And after that, press OK, and the installation shouldn't take more than two seconds to complete.

OK.

Now that the package is installed, let us load it by typing library(flexclust).

The flexclust package contains the object class KCCA, which stands for K-Centroids Cluster Analysis.

We need to convert the information from the clustering algorithm to an object of the class KCCA.

And this conversion is needed before we can use the predict function on the test set tumorVector.

So calling our new variable KMC.kcca and then using the as.kcca function, which takes as a first input the original KMC variable that stored all the information from the k-means clustering function, and the second input is the data that we clustered.

And in this case, it's the training set, which is the healthyVector.

And now, be aware that this data conversion will take some time to run.

Now that R finally finished creating the object KMC.kcca, we can cluster the pixels in the tumorVector using the predict function.

Let us call the cluster vector tumorClusters = predict(KMC.kcca, newdata=tumorVector).

And now, the tumorClusters is a vector that assigns a value 1 through 5 to each of the intensity values in the tumorVector, as predicted by the k-means algorithm.

To output the segmented image, we first need to convert the tumor clusters to a matrix.

So let's use the dimension function, and then the input is simply tumorClusters, and then using the c function with the first input as the number of rows in the tumorMatrix and the second input as the number of columns in the tumorMatrix.

And now, we can visualize the clusters by using the image function with the input tumorClusters matrix, and make sure to set the axes to FALSE, and let's use again these fancy rainbow colors, here.

So col=rainbow(k).

Again, k is equal to 5.

Alright.

Let's navigate to the graphics window, now, to see if we can detect the tumor.

Oh, and yes, we do!

It is this abnormal substance here that is highlighted in blue that was not present in the healthy MRI image.

So we were successfully able to identify, more or less, the geometry of the malignant structure.

We see that we did a good job capturing the major tissue substances of the brain.

The grey matter is highlighted in purple and the white matter in yellow.

For the sick patient, the substance highlighted in blue is the oligodendroglioma tumor.

Notice that we do not see substantial blue regions in the healthy brain image, apart from the region around the eyes.

Actually, looking at the eyes regions, we notice that the two images were not taken precisely at the same section of the brain.

This might explain some differences in shapes between the two images.

Let's see how the images look like originally.

We see that the tumor region has a lighter color intensity, which is very similar to the region around the eyes in the healthy brain image.

This might explain highlighting this region in blue.

Of course, we cannot claim that we did a wonderful job obtaining the exact geometries of all the tissue substances, but we are definitely on the right track.

In fact, to do so, we need to use more advanced algorithms and fine-tune our clustering technique.

MRI image segmentation is an ongoing field of research.

While k-means clustering is a good starting point, more advanced techniques have been proposed in the literature, such as the modified fuzzy k-means clustering method.

Also, if you are interested, R has packages that are specialized for analyzing medical images.

Now, if we had MRI axial images taken at different sections of the brain, we could segment each image and capture the geometries of the substances at different levels.

Then, by interpolating between the segmented images, we can estimate the missing slices, and we can then obtain a 3D reconstruction of the anatomy of the brain from 2D MRI cross-sections.

In fact, 3D reconstruction is particularly important in the medical field for diagnosis, surgical planning, and biological research purposes.

I hope that this recitation gave you a flavor of this fascinating field of image segmentation.

In our next video, we will review all the analytics tools we have covered so far in this class and discuss their uses, pros, and cons.