Brain tumor classification

1. Dataset

The dataset contains 3064 weighted 2D, MRI type, T1 images recorded from 233 patients with 3 types of tumors:

1. meningiom 708 images
2. gliom 1426 images
3. tumor of the pituitary gland 930 images

The dataset also contains the patient's id in order to be able to divide the dataset, so that there are no images of a patient in both the training set and the test set. .
This dataset was also used in articles [1]. In this article, the authors did not use deep neural networks but obtained very good results using Fisher vectors, about 94.68% average accuracy for classification.

2. Data preprocessing

2.1 Image Resize

The images in the dataset are 512x512 pixels and because of low computing power the I resized image at 128x128 pixels, being able in this case, without exceeding the GPU memory provided by Google Colab.

2.2 Image normalization

Normalization is a very common process in the field of image processing, there are several methods by which it can be achieved:

- image standardization This operation is very common in the field of machine learning because it "centers" the data at a normal distribution of mean 0 and standard deviation 1. This avoids getting very large or very small gradients which implicitly leads to better performance of the model.

In order to use unsupervised algorithms on this data set, I extract some features.

Using „pyradiomics” library I extracted two types of feature:

1. GLCM, the grey level co-occurrence matrix which is a statistical method of finding the textures by considering the spatial connection of the pixels. This matrix contains 25 features.
2. GLRLM, the grey level run length matrix is also used for texture feature extraction using „run length” which represent the number of adjacent pixels that have the same grey intensity in a particular direction. This matrix contains 17 features.

Using resnet101 I extracted another two types of features:

1. feautures from the last layer before the fully connected from the Resnet101 pretrained on Imagenet.
2. feautures from the last layer before the fully connected from the Resnet101 pretrained on Imagenet and trained on datasets images.

3. Algorithms

For this project I choose to analyse two unsupervised algorithm for clusterization, kmeans and Agglomerative Hierarchical clustering. Two compare these algorithm, I used the hungarian algorithm to calculate the best mapping between the clusters given by algorithms, and the labels from the dataset.

I ve splitted the data into train and test subset, to see what algorithm is generalizing better.

3.1 Kmeans

3.1.1 Simple description

To process the learning data, the K-means algorithm in data mining starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids.It stops creating and optimizing clusters when either:

- The centroids have stabilized — there is no change in their values because the clustering has been successful.
- The defined number of iterations has been achieved.

I trained the Kmeans, for every type of extracted feature using the C parameter equal to number of classes.

I. GLCM and GLRLM features. I trained the Kmeans on the original features, and also on the features obtained from the PCA, on the concatenated GLCM, GLRLM feature.

To choose the best number of components, I plotted a graph to see the variance in data by number of components.

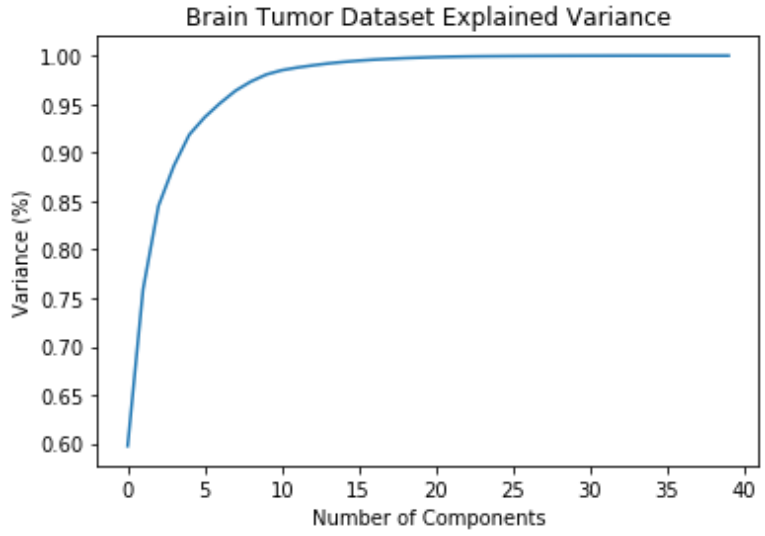


Fig. 1 Number of components to choose from features by variance on GLCM & GLRLM features.

II. Features from Resnet101 trained or not

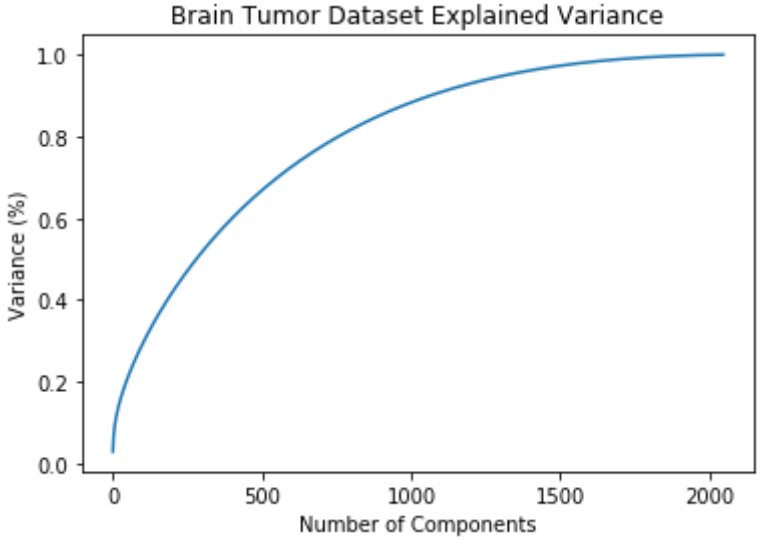


Fig. 1 Number of components to choose from features by variance on Resnet101 only pretrained on ImageNet.

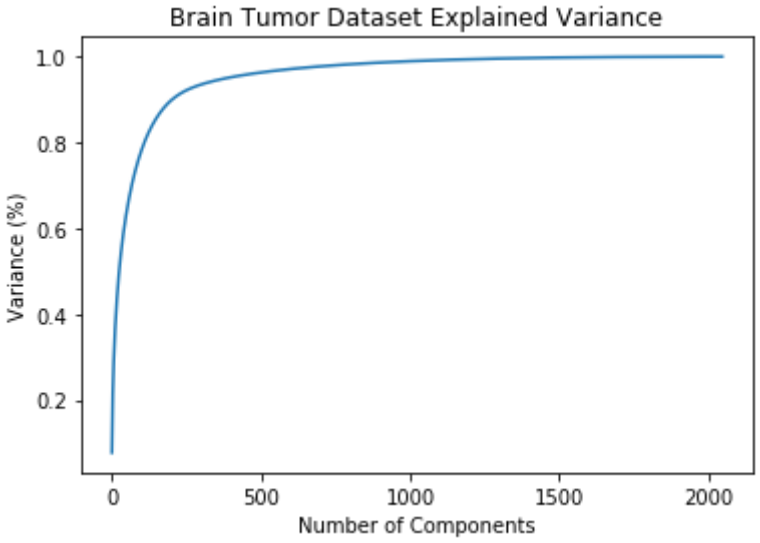


Fig. 1 Number of components to choose from features by variance on Resnet101 pretrained on ImageNet, and the trained on dataset.

3.1.2 Results Kmeans

Features	**Train Accuracy**	**Test Accuracy**
GLCM & GLRM	**0.479**	**0.365**
PCA from GLCM & GLRM	**0.480**	**0.363**
Features Resnet101 pretrained (Imagenet)	**0.446**	**0.435**
PCA from Features from Resnet101 pretrained on Imagenet	**0.447**	**0.435**
Features Resnet101 pretrained (Imagenet) and then on set	**0.922**	**0.861**
PCA Features Resnet101 pretrained (Imagenet) and then on set	**0.923**	**0861**

Table. 1 Results Kmeans

I have plotted the data to see, if there are observable cluster. Because the features were multidimensional, GLCM & GLRM have like 40 columns, and the features extracted from ResNet101 have 2048 columns, I used TSNE algorithm to reduce the data data to 3 components. Bellow are the results.

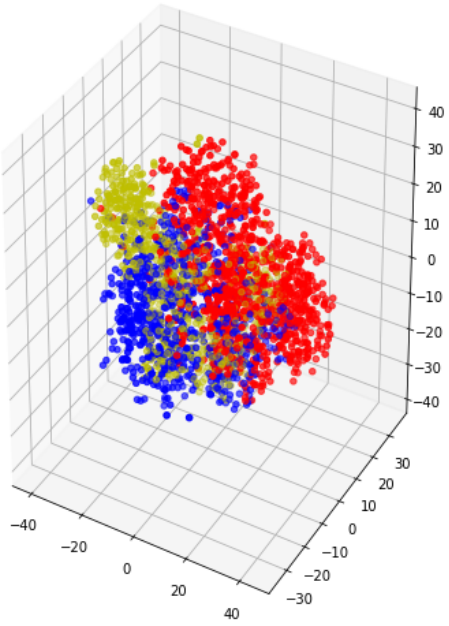


Fig. 2 Train features from trained Resnet

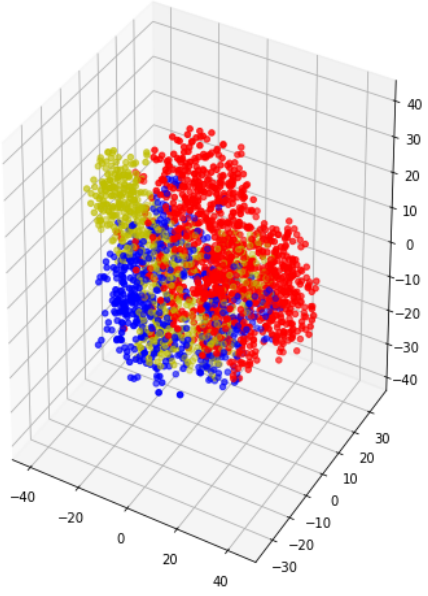
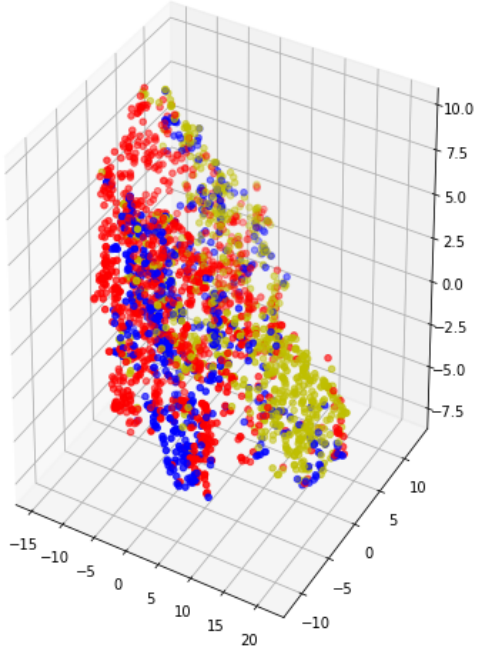


Fig. 3 Train features clustered from trained Resnet



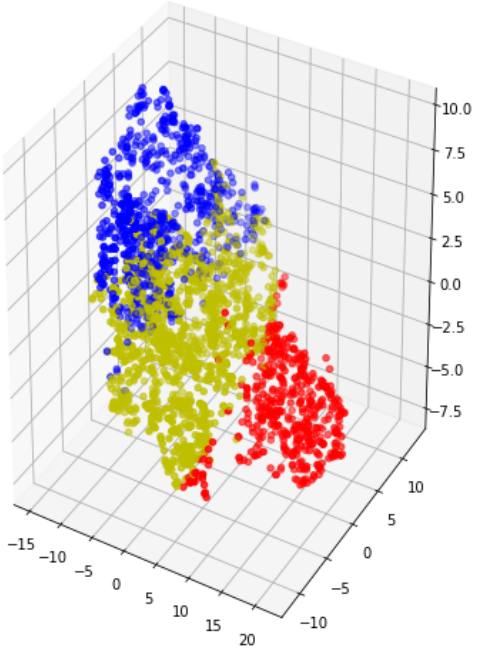


Fig. 4 Train features from GLCM & GLRLM

Fig. 4 Train features from GLCM & GLRLM

3.2 Hierarchical clustering Agglomerative

3.2.1 Description

It is a method of clustering, which consider each datapoint as an individual cluster. At each iteration similar clusters merge with other clusters until one cluster or K clusters are formed.

This algorithm doesn't have a method two predict the cluster for new points. So I used a function two calculated the centroids based on, cluster assignment given by hierarchical clustering, and then using this centroids I predict the new cluster for each point in the dataset using the Kmeans.

3.2.2 Results

Features	**Train Accuracy**	**Test Accuracy**
GLCM & GLRM	**0.558**	**0.423**
PCA from GLCM & GLRM	**0.532**	**0.421**
Features Resnet101 pretrained (Imagenet)	**0.468**	**0.451**
PCA from Features from Resnet101 pretrained on Imagenet	**0.523**	**0475**
Features Resnet101 pretrained (Imagenet) and then on set	**0.90**	**0.861**
PCA Features Resnet101 pretrained (Imagenet) and then on set	**0.90**	**0.86**

Table. 2 Result obtained with Hierarchical clustering Agglomerative

3. Conclusion

The best model is the Kmeans which seems to clusterize better the data. Also the feature extracted by “hand” using the GLCM and GLRLM algorithm, doesn’t help to much, and the accuracy for both algorithm is low.

The CNN proved again to be a very good feature extractor. Using these featuresthe cluster assignment, give an accuracy of 0.92, which is close to the accuracy of the ResNet101 supervised method.

1. Cheng, Jun, et al. „Enhanced Performance of Brain Tumor Classification via Tumor Region Augmentation and Partition.” PloS one 10.10 (2015)