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 the case of feed forward and CNN networks, I normalized the images with the standard and standard deviation of the training set, and in the case of the SVM I scaled the features obtained using the MinMax scaler.

3. Algorithms

3.1 Support Vector Machine

3.1.1 Simple description

In a few words, a Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. More explicit, given labeled training data (*supervised learning*), the algorithm outputs an optimal hyperplane which categorizes new examples. In this problem I have three classes, and becauses SVM builds is a binary classifier, the best practice is to build |C| (number of classes) „one versus rest classifier”. This method is already implemented in the library I used, sklearn.

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.

I trained the 3 SVMs, one for GLCM features, one for GLRLM and another on concatenated features of this matrixes.

Using GRID search implemented in sklearn I’ve trained the SVM on following tuned_params

```
[{'kernel': ['rbf'], 'gamma': [1e-3, 1e 4], 'C': [1, 10, 100, 1000]},

 {'kernel': ['linear'], 'C': [1, 10, 100, 1000]}].
```

The best parameters selected by grid search:

- C=1000
- kernel='linear'

3.1.2 Results SVM

GLCM

CLASS	PRECISION	RECALL	F1-Score	Examples
1	0.76	0.63	0.69	165
2	0.83	0.88	0.85	275
3	0.66	0.71	0.68	190
Accuracy				0.76

GLRLM

CLASS	PRECISION	RECALL	F1-Score	Examples
1	0.78	0.53	0.63	165
2	0.77	0.87	0.82	275
3	0.66	0.73	0.69	190
Accuracy				0.73

GLCM + GLRLM

CLASS	PRECISION	RECALL	F1-Score	Examples
1	0.81	0.68	0.74	165
2	0.85	0.91	0.88	275
3	0.70	0.73	0.72	190
Accuracy				0.80

The best classifier was trained on concatenated features of GLCM and GLRLM.

3.2 FeedForward net

3.2.1 Description

I used a feedforward net with only 1 hidden layer of size 100. Because this dataset has a small number of examples, I used cross-validation and early-stopping. Also as optimizer for backpropagation I used SGD, with a scheduler for the learning rate. The scheduler is deacreased with a factor of 0.75 after every 3 epochs in which the test loss doesn't decrease.

The loss function used for the gradient descent is crossentropy.

3.2.2 Results

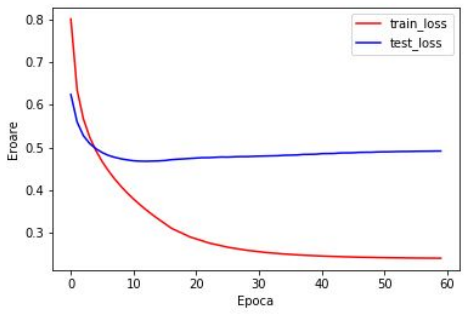
I have used 5-fold, and also I augmented the data with rotations, vertical and horizontal flips.

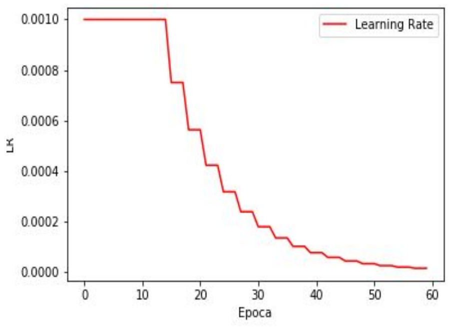
To measure the model I also used the recall and precision metrics.

NET	**Accuracy**	**Recall**	**Precision**	**F1 score**
Feed_1	0.77	0.74	0.75	0.73
Feed_2	0.81	0.79	0.78	0.78
Feed_3	0.69	0.69	0.67	0.67
Feed_4	0.75	0.68	0.67	0.65
Feed_5	0.72	0.66	0.68	0.66

The accuracy of 0,74 is the mean accuracy over the 5 models.

I have plotted the loss , learning_rate, accuracy and for the best model.





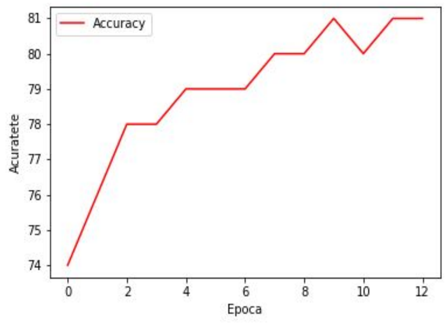


Fig.1 Loss Fig.2 Learning rate Fig. 3 Accuracy

3.3 CNN

3.3.1 Description

Convolutional Neural Network is a class of deep neural network that is used for Computer Vision or analyzing visual imagery. This type of network is somehow similar with multilayer perceptron, but uses convolution process for extracting the feautures from images.

This network is made up of several types of layers:

- **convolution layer**: This layer performs a series of operations to extract features from the input.
- **the nonlinearity layer** : I used the ReLU activation function, as it is very computationally efficient and also solves the problem of vanishing gradient.
- **the pooling layer** that helps reduce the convolution layer dimension

- **the dropout layer**, which ignores some connections between different neighboring neurons, from one iteration to another so that they do not learn according to the neighbors. This can get rid of overfit.

I used crossentropy loss function for classification of the tumors.

For this task of classification, I used the „transfer learning” method. I choose the ResNet101 pretrained on ImageNet. I trained the net for 80 epochs with „frozen” layers of convolution, the only learnable parameters are in the Linear layer, which realize the classification.

After 80 epochs, I trained all layers of the network for another 40 epochs and the accuracy increased. I have used the same optimiser, and learning rate with scheduler to compare the results with the feed forward.

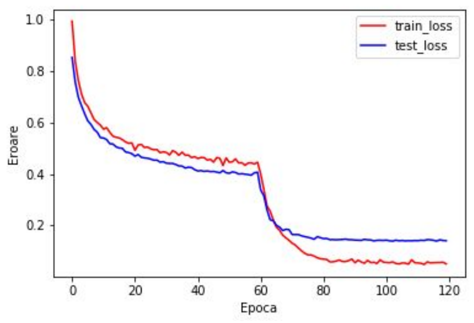
3.3.2 Results

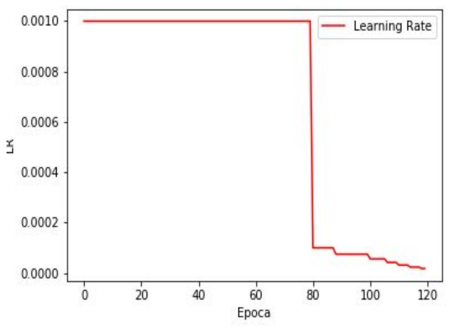
I have used 5-fold, and also I augmented the data with rotations, vertical and horizontal flips.

NET	**Accuracy**	**Recall**	**Precision**	**F1 score**
Resnet_1	0.89	0.87	0.88	0.97
Resnet_2	0.95	0.94	0.88	0.88
Resnet_3	0.89	0.89	0.88	0.88
Resnet_4	0.89	0.88	0.90	0.89
Resnet_5	0.94	0.94	0.92	0.93

The accuracy of 0,912 is the mean accuracy over the 5 models.

I have plotted the loss , learning_rate, accuracy and for the best model.





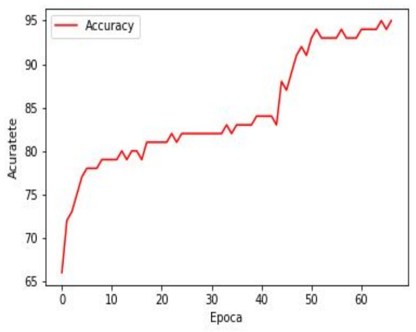


Fig.1 Loss Fig.2 Learning rate Fig. 3 Accuracy

3. Conclusion

The best model is the Resnet101 which classify the tumors with an accuracy of 0.912. The convolutional neural network is better than SVM and feedforward because this type of network are greater for capturing local information. I believe that the CNN did better due to the technique of transfer learning, because the model came pre-trained on a very large data set, and thus has well-initialized weights.

Also the SVM proved to be better than the feed forward network because it was trained directly on the statistical feature extracted from GLCM and GLRLM matrix. Another reason why the SVM has performed well would be that the number of features is much smaller than the number of examples.

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