ESKISEHIR TECHNICAL UNIVERSITY

Engineering Faculty

Computer Engineering Department

BIM464 HUMAN COMPUTER INTERACTION FINAL PROJECT

BRAIN TUMOR CLASSIFICATION AND GRAD-CAM VISUALIZATION

29330067606 – Ayşegül TERZİ

29956154056 – Bahar GÖRGÜN

12/06/2023

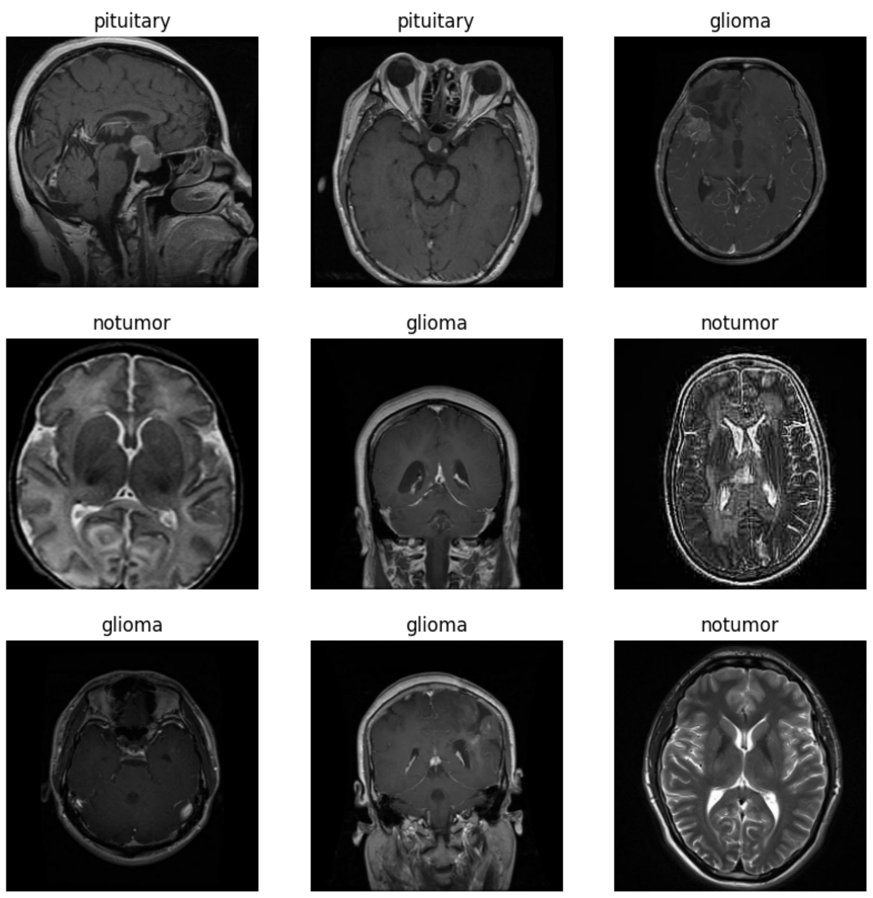
Introduction

A brain tumor is a growth of cells in the brain or near it. Brain tumors can happen in the brain tissue. Brain tumors also can happen near the brain tissue. Nearby locations include nerves, the pituitary gland, the pineal gland, and the membranes that cover the surface of the brain. Many different types of primary brain tumors exist. Some brain tumors aren't cancerous. These are called noncancerous brain tumors or benign brain tumors. Noncancerous brain tumors may grow over time and press on the brain tissue. Other brain tumors are brain cancers, also called malignant brain tumors. Brain cancers may grow quickly. The cancer cells can invade and destroy the brain tissue.

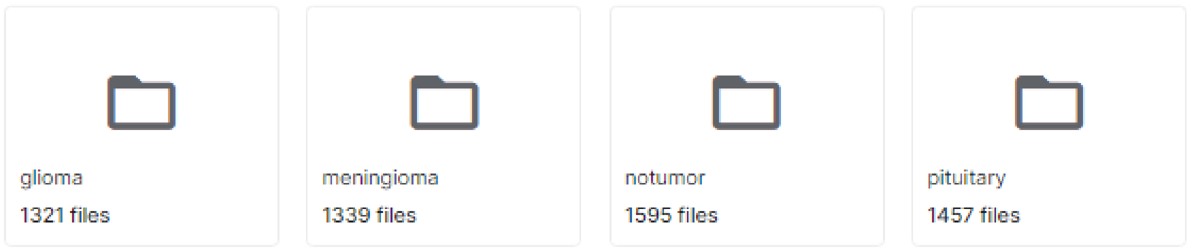
Our goal in this project is to accelerate the treatment process by determining the type of brain tumor. We used transfer learning (from imageNet) and a resnet-based CNN architecture to classify tumors. We also located the tumor using Grad-Cam visualization. Transfer learning is about leveraging feature representations from a pre-trained model, so you don't have to train a new model from scratch. The pre-trained models are usually trained on massive datasets that are a standard benchmark in the computer vision frontier. That's why we used weights from ImageNet.

Dataset

Our Brain Tumor Dataset is the sum of three datasets. This dataset contains 7023 images of human brain MRI images which are classified into 4 classes: glioma - meningioma - no tumor and pituitary, no tumor class images were taken from the Br35H dataset. Testing file includes 4 directories and 1311 images. Training file includes 4 directories 5712 images. This dataset is balanced. By balance, we mean that there is not much difference between the number of brain tumor types in the dataset. If there was too much difference, any tumor type could be weighted when training our model. This would negatively affect the performance of our model.



Training Set



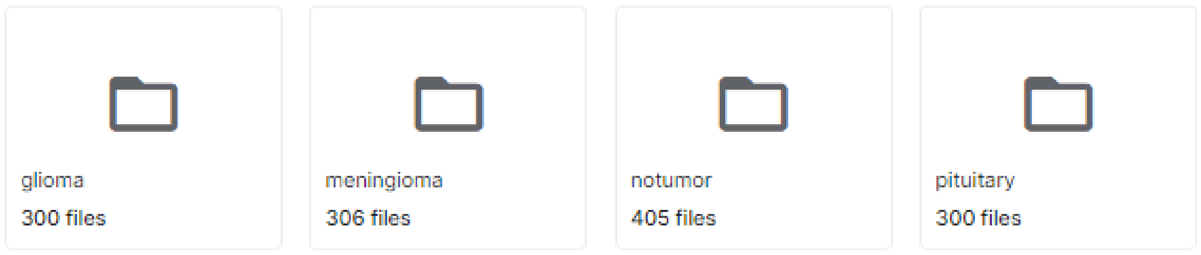
glioma
1321 files

meningioma
1339 files

notumor
1595 files

pituitary
1457 files

Test Set



glioma
300 files

meningioma
306 files

notumor
405 files

pituitary
300 files

Methods

ResNet is a deep learning model that uses residual blocks to learn complex functions from the input data. Residual blocks are subnetworks that have a skip connection that performs identity mapping, which means that the input of the block is added to the output of the block. This helps to avoid the problem of vanishing or exploding gradients that occurs when training very deep networks. ResNet architecture is inspired by VGG networks, which use 3x3 convolutional filters. ResNet can have different depths, such as 34, 50, or 152 layers. ResNet has achieved state-of-the-art results on image recognition tasks such as ImageNet. We used relu, softmax as activation functions.

ResNets are a type of deep learning architecture that have been shown to be very effective for solving a variety of problems, including clinical problems. ResNets are able to learn complex relationships between features in data, which makes them well-suited for tasks such as image classification, object detection. One of the other reasons for choosing this architecture is the usage of Grad Cam in our project. Grad Cam is a technique for visualizing the regions of an input image that are most relevant for a convolutional neural network's prediction. It stands for Gradient-weighted Class Activation Mapping, and it works by computing the gradients of the output class with respect to the feature maps of the last convolutional layer. These gradients are then used to weight the feature maps, and a linear combination of them produces a coarse localization map that highlights the important regions for the prediction.

We used Supervised Learning because this training strategy is a frequently used method in classification algorithms.

We preferred to use Adam as optimizer and SparseCategoricalCrossentropy as loss function. Computes the crossentropy loss between the labels and predictions. If your Yi''s are one-hot encoded, use categorical_crossentropy. Examples (for a 3-class classification): [1,0,0] , [0,1,0], [0,0,1] But if your Yi' 's are integers, use sparse_categorical_crossentropy. Examples for above 3-class classification problem: [1], [2], [3]

One advantage of using sparse categorical cross entropy is it saves time in memory as well as computation because it simply uses a single integer for a class, rather than a whole vector.

Adam optimizer is a powerful tool for improving the accuracy and speed of deep learning models. It helps the neural network learn faster and converge more quickly towards the optimal set of parameters that minimize the cost or loss function. Adam optimizer are generally better than every other optimization algorithm and appropriate for problems with very noisy/or sparse gradients.

Experiments

We didn’t use any preprocessing stage because our data was regular and has no missing values in it. Since our data are mostly on the same scale, we did not need normalization or standardization.

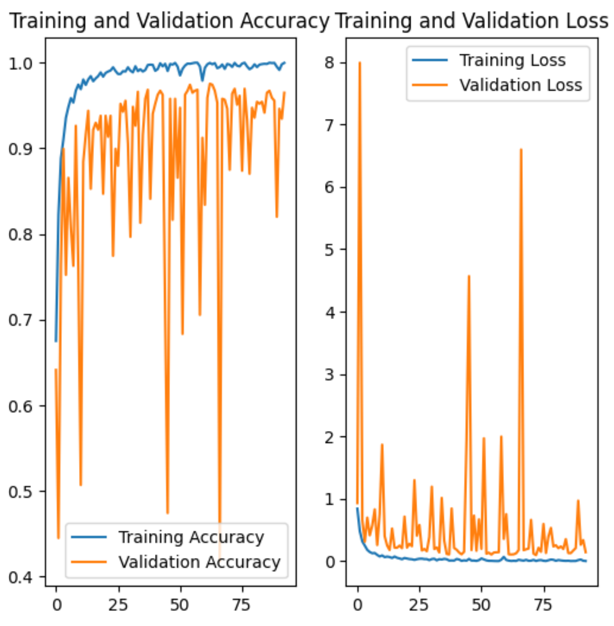
Augmentation : We used random flipping and random brightness data augmentation technique because we used MRI images. MRI images may be dark and trapezoid. We wanted to prevent that.

Dataset Usage: We splitted dataset into train and test datasets, 15% - 85%. We used the same dataset for train and validation.

Initializing Weights: We used transfer learning in this project, so we took weights from ImageNet and continued training from these weights.

Dataset Size: This dataset contains 7023 images so it was good enough to train a good model.

Learning curve analysis: Analyze the learning curve to identify overfitting or underfitting

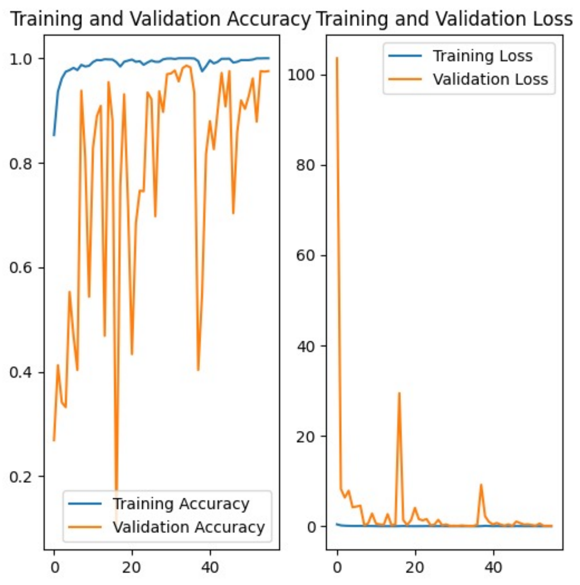


We trained this model to see the performance of the model and changing of the accuracy-loss graphs in more epochs. It is basically overfitted but that was the plan :)

Maybe, our validation set is small and not enough because this is noisy graphic.

batch size : 16 / 93 epoch / dropout 0.30

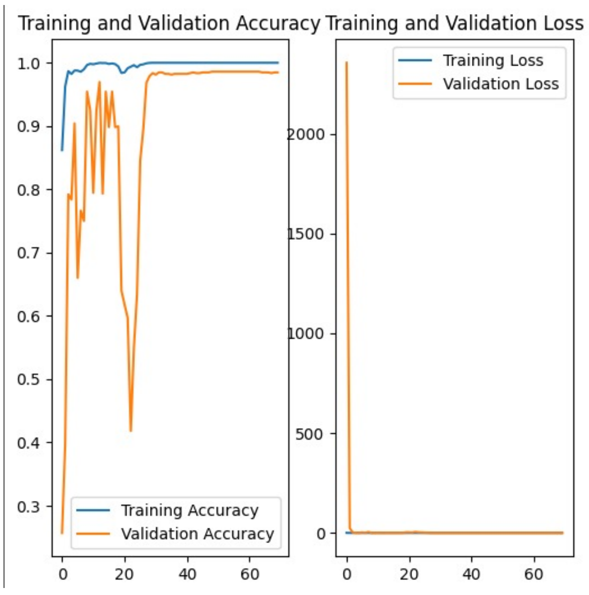
We used fully-connected layer that model (first activation :relu and second activation softmax)



This model is our ideal model. It stopped at the early stopping point with patience number=20 and its performance was the best in other images. Maybe good but close to overfitting.

batch size : 64 / epoch 52 / dropout : 0.30

We used fully connected layer (activation = softmax)



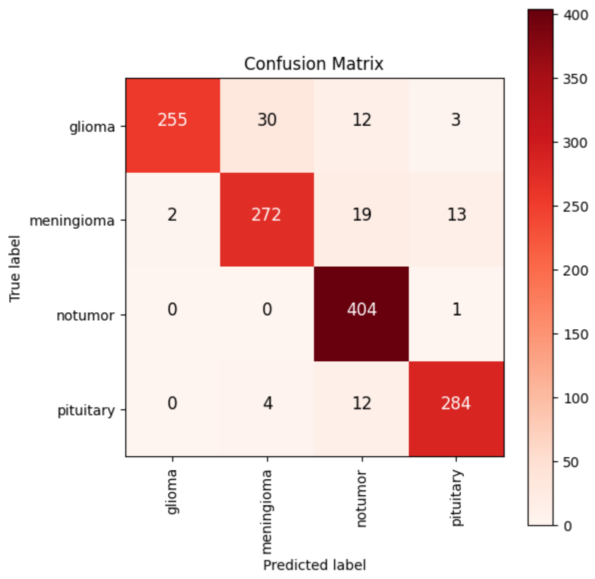
batch size 128 / dropout : 0.25 / 70 epoch

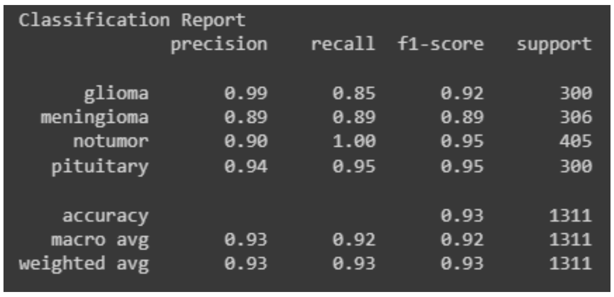
We used fully connected layer (activation = softmax)

Hyperparameters:

batch size	16
image height	224
image width	224
validation split	0.15
seed	123
dropout	0.30
activation functions	softmax, relu
patience number	20
optimizer	Adam
loss function	SparseCategoricalCrossentropy()
metrics	accuracy

Model results:





```
Classification Report
               precision    recall  f1-score   support

      glioma       0.99      0.85      0.92       300
  meningioma       0.89      0.89      0.89       306
     notumor       0.90      1.00      0.95       405
   pituitary       0.94      0.95      0.95       300

    accuracy                           0.93      1311
   macro avg       0.93      0.92      0.92      1311
weighted avg       0.93      0.93      0.93      1311
```

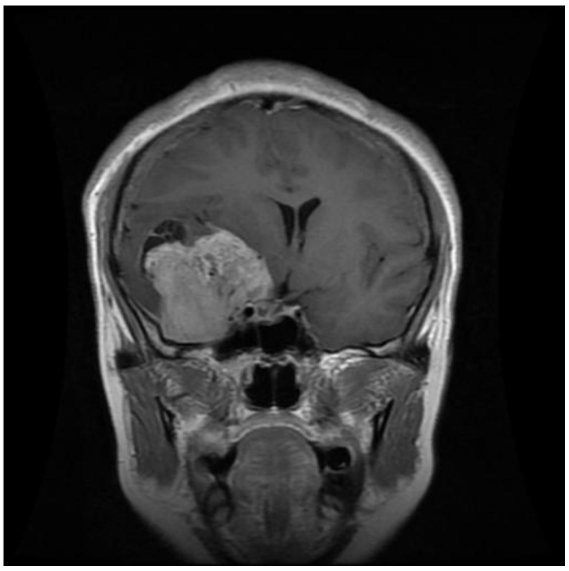
Confusion Matrix

Evaluation metrics

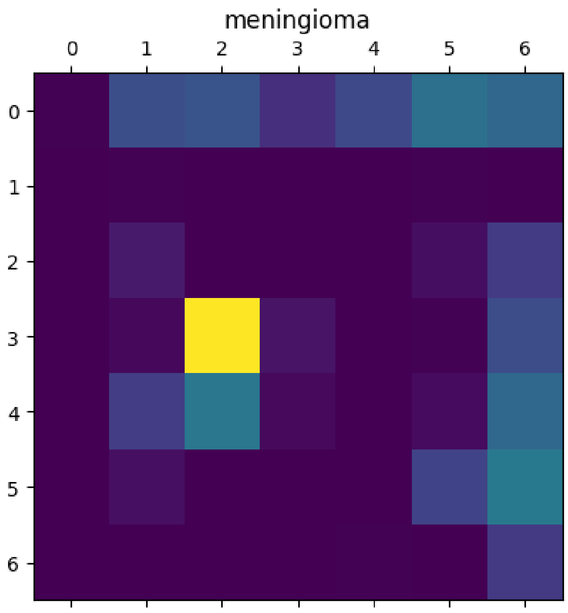
.

Conclusion and Discussion:

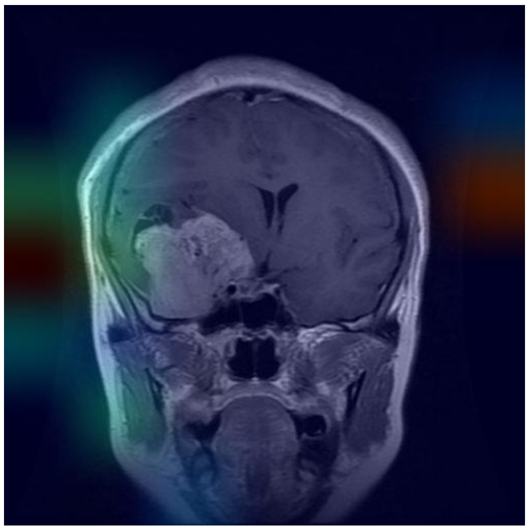
Our model created a noisy graph. This may be because our validation set is small. Our model created a noisy graph. This may be because our validation set is small. When we increased the batch size, we reached a model that is very noiseless and almost has an accuracy metric of 1. We could have separated the dataset better. It also seems that we have some mistakes when editing our architecture. We tried to change these by making different experiments. But we didn't reach the good chart we wanted.



image



heatmap



Grad Cam Visualization

Project codes

https://colab.research.google.com/drive/1b4NEoGQAusb8kpBxgGxDS9M-1BKKTDht?usp=sharing