DEEP LEARNİNG ASSİGMENT

İlhan Randa 44095242426

Ahmet Emirhan Terzi 53188550904

**Deep Learning for Brain Tumor Detection Report**

1. Problem Definition The objective of this study is to develop a binary image classification model using deep learning techniques to detect brain tumors from MRI images. Specifically, the model aims to distinguish between two types of brain tumors: meningioma and glioma. The model's task is to accurately classify MRI images into these two categories, enabling early detection and diagnosis of brain tumors.
2. Explanation of the Neural Network Model For this task, a Convolutional Neural Network (CNN) architecture is employed. CNNs are well-suited for image classification tasks as they can automatically learn and extract features from images.

Model Architecture:

* Input Layer: Accepts input images of size 256x256 pixels with 3 color channels (RGB).
* Convolutional Layers: Two convolutional layers with 32 and 64 filters respectively, each followed by a MaxPooling layer for spatial downsampling.
* Flatten Layer: Converts the 2D feature maps into a 1D vector for input to the fully connected layers.
* Fully Connected Layers: A dense layer with 256 neurons and ReLU activation function, followed by a dropout layer to prevent overfitting.
* Output Layer: Consists of a single neuron with a sigmoid activation function, producing binary classification results.

Model Compilation:

* Optimization Algorithm: Adam optimizer with a learning rate of 0.0001 is used for efficient model training.
* Loss Function: Binary crossentropy loss function is employed, suitable for binary classification tasks.
* Evaluation Metric: Model performance is evaluated based on accuracy, which measures the percentage of correctly classified images.

1. Dataset Description and Acquisition The dataset consists of MRI images categorized into two classes: meningioma and glioma. The dataset is divided into training, validation, and testing sets, each containing images of both tumor types. Data augmentation techniques such as rotation, width and height shifting, shear transformation, zooming, and horizontal flipping are applied to the training images to increase dataset diversity and prevent overfitting. The augmented images are then used for model training.
2. Results After training the model on the training and validation datasets for 40 epochs, the model's performance is evaluated using the test dataset. The following results were obtained:

Test Loss: 0.3823698163032532

Test Accuracy: 0.8126094341278076

These metrics indicate how well the model generalizes to unseen data and its ability to accurately classify MRI images.

1. Discussion In this study, a CNN model was successfully trained to detect brain tumors from MRI images with high accuracy. However, further improvements can be made by:

* Collecting more diverse and extensive datasets to enhance the model's ability to generalize to different variations.
* Fine-tuning the model architecture by adjusting the number of layers, filters, and neurons to improve feature extraction and classification performance.
* Exploring advanced regularization techniques to mitigate overfitting and improve model robustness.

Overall, this study demonstrates the effectiveness of deep learning techniques in brain tumor detection and opens avenues for future research in this field.

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

It is the output of 10 epochs after the 40 epoch.

metin, diyagram, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, ekran görüntüsü, çizgi içeren bir resim

Açıklama otomatik olarak oluşturulduGraphs

Confusion Matrix

ekran görüntüsü, metin, dikdörtgen, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

Prediction  
  
röntgen filmi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturulduHow to execute our code

Firstly install Python and Anaconda 3 then lauch the jupyternotebook after that install the requirements you will need the following python libraries:

* numpy
* tensorflow
* scikit-learn
* matplotlib
* Pillow

Next step is save the dataset in the directories we specified in the code.  
Finally you can execute the code.