

Brain Tumor Detection CNN

By Miłosz Tkacz

Based on this sources of data:

<https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection> - Train/Test

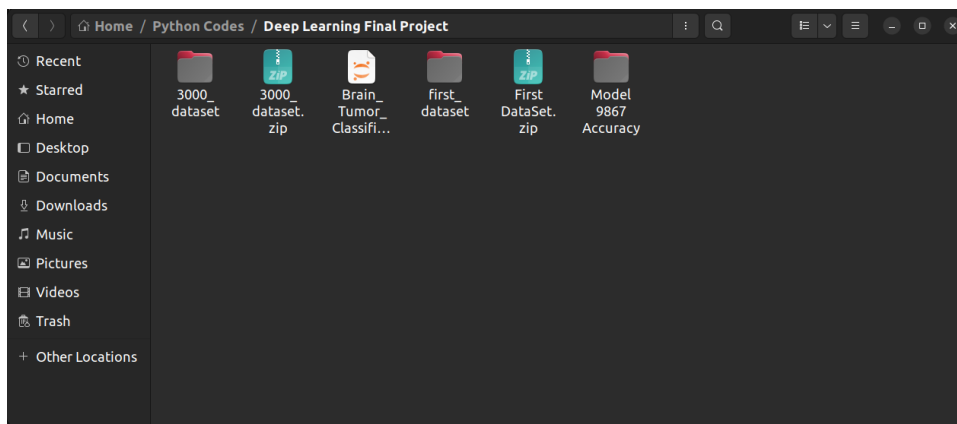
<https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection> - Test 2

The goal of this project was to leverage deep learning techniques to accurately classify brain MRI images into two categories: tumor-present and tumor-absent. The CNN model was designed, trained, and evaluated using a dataset consisting of labeled brain MRI scans.

Installation:

To install the model tensorflow, matplotlib and all their dependencies have to be installed in python (3.11.2) environments.

1. Put all the files supplied inside a single folder
2. Download the datasets from their original directories
3. Rename the zip files, the bigger file should be named "3000_dataset" and the smaller "first_dataset", optionally the code inside of the .ipynb file may be changed to fit different directory names.
4. Unpack the zip files into two separate folders for each dataset
5. Delete the unnecessary folders inside of the dataset folders so that only "yes" and "no" folders are present.
6. Unpack the "Model 9867 Accuracy.zip" into a folder with the same name.
7. Run all cells in .ipynb file.



Example of the well-structured folder.

Training the model

The model was trained on 2400 samples of data and tested on 853 samples of data.

The images were turned into grayscale and scaled to 64x64 pixels.

```

# Creating the model
model = tf.keras.Sequential([
    # Input tensors are from range 1 to 255 so need rescaling.
    tf.keras.layers.Rescaling(1./255, input_shape=(64, 64, 1)),
    # 3 Convolutional + MaxPool Layers
    tf.keras.layers.Conv2D(filters = 10, kernel_size=(5, 5)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(filters = 90, kernel_size=(5, 5)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(filters = 50, kernel_size=(5, 5)),
    tf.keras.layers.MaxPooling2D(4, 4),
    # Flatenig and Clasifying the Data.
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(1, activation="sigmoid")
])

# Using Adam Optimizer LR has to be very small for the model to perform < 0.0001
optimizer = tf.keras.optimizers.Adam(learning_rate=0.00008)
metrics = ["accuracy", tf.keras.metrics.AUC(), tf.keras.metrics.Precision(), tf.keras.metrics.Recall()]

model.compile(optimizer=optimizer, loss="binary_crossentropy", metrics=metrics)

model.build()
model.summary()

```

Those are the final settings of the provided model.

There were numerous difficulties in the training process the most notable are:

1. Leveraging the complexity and accuracy of the model. I've been experimenting with different combinations of layers including Convolutional Layers and Dense Layers (which were deleted in the end as they did not improve the accuracy).
2. Finding optimal learning rate. As you can see learning rate in this model is particularly small, I've found that learning rate greater than 0.001 was unable to train at all. (Accuracy == 0.5). After trial and error I've found the most optimal values (0.00007 - 0.00008)

Final Results

```

# Loading the best performing model
model_loaded = tf.keras.models.load_model("Model 9867 Accuracy")

# Best Model Evaluation
model_loaded.evaluate(test)
model_loaded.evaluate(test2)

19/19 [=====] - 0s 3ms/step - loss: 0.0641 - accuracy: 0.9867 - auc_2: 0.9968 - precision_1: 0.9806 - recall_1: 0.9934
8/8 [=====] - 0s 3ms/step - loss: 0.0989 - accuracy: 0.9842 - auc_2: 0.9860 - precision_1: 0.9748 - recall_1: 1.0000

```

The final version of the model achieved great results with 98% Accuracy, AUC == 0.99, 98% Precision and 99% Recall in the main dataset and comparable results from the dataset from other source.

Positive results with the data from other source show the versatility of the model.

Apart from great accuracy there are some drawbacks to this model:

1. It is unable to analyse brain scans using different perspectives.
2. It is unable to distinguish between different types of cancer.
3. In case of very small cancerous tissue volume it may struggle to make proper classification, because of heavy downscaling of resolution (64x64 pixels).