

Brain tumor classification

Done by: Turganbekov Rakhat, Boltay Beksltan

Group: BDA 2102

GitHub: <https://github.com/Bek7182/FinalProject>

YouTube:

https://www.youtube.com/watch?v=Xh-grphPO5o&ab_channel=RakhatTurganbekov

Introduction:

- **Problem**

Brain tumors are a significant health concern, with more than 700,000 new cases diagnosed each year worldwide. It is the growth of cells in the brain or near it. Nearby locations include nerves, the pituitary gland, the pineal gland, and the membranes that cover the surface of the brain. Accurate and timely diagnosis is crucial for effective treatment and management of the condition. One approach to improve brain tumor diagnosis is to develop machine learning models that can automatically analyze medical images to detect a tumor and classify its type. In this project, we focus on the classification of four types of brain tumors - glioma, meningioma, pituitary, and no tumor using machine learning algorithms.

- **Literature review**

They used a deep learning model to classify the types of images such as gliomas, meningiomas, non-tumors, and pituitary tumors. The classification process was performed using recurrent convolutional neural networks (CNN). The proposed classifier obtained 95.17% accuracy in classifying brain tumor tissues from MRI images.

Vankdothu, R., & Hameed, M. A. (2022). Brain tumor MRI images identification and classification based on the recurrent convolutional neural network. *Measurement: Sensors*, 24, 100412. <https://doi.org/10.1016/j.measen.2022.100412>

The author uses the complexity of machine learning algorithms, showing and comparing different models in order to create the most productive model. In this way, he was able to achieve the greatest accuracy of his model. Unfortunately, it was not possible to see more detailed and accurate data on its work in the output.(ROC-AUC, confusion matrix)

<https://www.kaggle.com/code/lukasmendes/brain-tumor-cnn-98>

The author of this project presents an excellent structure and layout of the project. He uses good visualization elements at the beginning of a project. Also uses a pre-trained model VGG16 freezes the first layers and fine-tuned the model.

<https://www.kaggle.com/code/mushfirat/brain-tumor-classification-accuracy-96>

On your advice, we turned to this source to see examples of other topics in the field of medicine related to classification. We have identified for ourselves the most productive pre-training models, as well as various models of authors.

<https://paperswithcode.com/task/brain-tumor-segmentation>

- **Current work**

The main IDE we used is Google colabatory, but we used Kaggle for faster training and testing of the model. The dataset of brain tumors was also selected from Kaggle's directory. After installing it, we uploaded all the MRI photos to Google Drive. The dataset consists of two folders, training and test, which in turn contain four folders with photos inside (four classes of tumors). For a more understandable visual representation of the data, we used a pie chart that shows the distribution of images for each class, as well as a histogram to show the total number of images in directories.

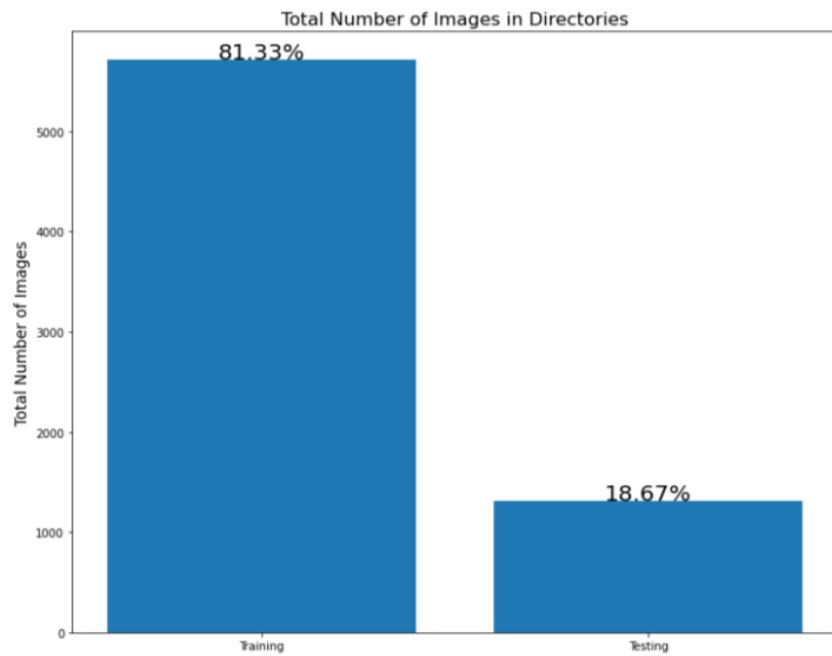
Also, for the example of MRI photos, photos were randomly selected from each class and displayed. At the data pre-processing stage, we standardized the dimensions of the dataset photos and brought them (128,128) in addition to these label photos, as a result of which we received x_train, y_train, x_test, y_test. The next stage of data preparation is converting y_train, y_text labels to one hot encoded vectors, as well as ImageDataGenerator where batches of tensor image data with real-time data augmentation. The project contains three models, the first two were created manually, and the third uses a pretrained ResNet50v2 model with frozen layers and fine-tune of the model.

After each model, the basic metrics of loss and accuracy of the model are derived. To evaluate the model, the Confusion matrix, ROC AUC and Metrics table were used, this makes it possible to make a deeper analysis of each model and compare the performance of all three. In addition, a personal MRI image of Turganbekov Rahat was used to verify the reliability of the model and compare the result with the doctor's conclusion.

Data and Methods:

- **Information about the data**

We are taking a brain tumor MRI dataset (dataset for classifying brain tumors) from Kaggle. It is already divided into testing and training folders with 4 folders for each class in each of them. We also used 10% of the training dataset to create validation data by ImageDataGenerator. MRI Dataset contains 7023 MRI images of the human brain in jpg format, which are classified into 4 classes: glioma, meningioma, no tumor and pituitary. Training and testing sets are divided by 81.33% and 18.67%, respectively.

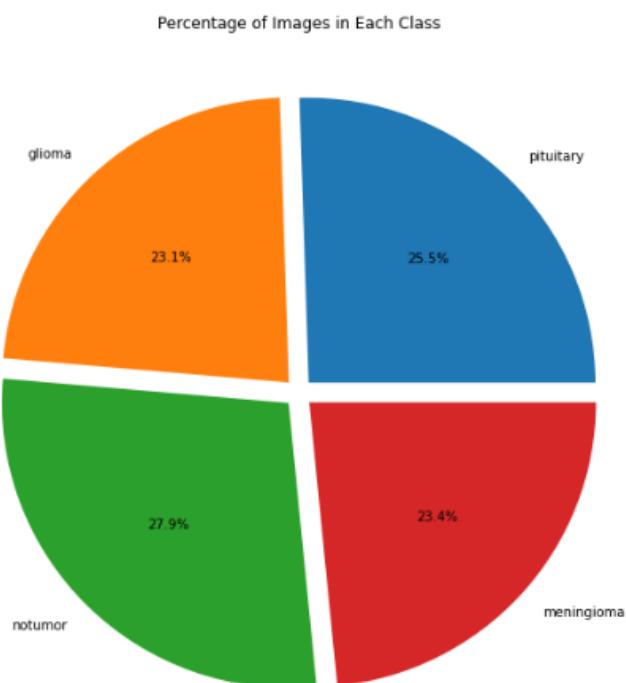


The training dataset consists of:

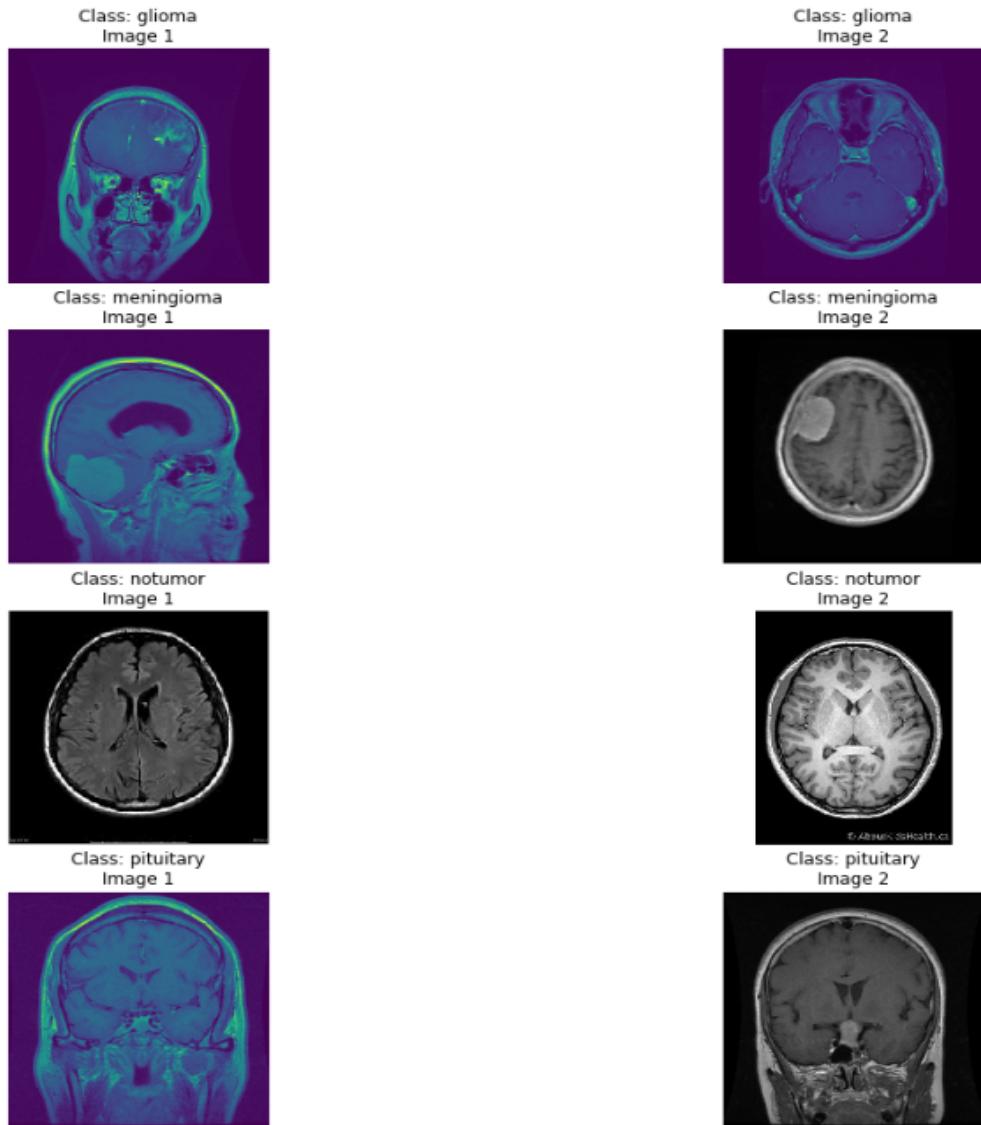
- 1321 - MRI images of glioma
- 1339 - MRI images of meningioma
- 1595 - MRI images of no tumor
- 1457 - MRI images of pituitary

The testing dataset consists of:

- 300 - MRI images of glioma
- 306 - MRI images of meningioma
- 405 - MRI images of no tumor
- 300 - MRI images of pituitary



As we can see, the dataset contains approximately the same number of images for each class, but the number of notumor MRI images(405) is slightly larger than the others(300).



We randomly displayed 2 images for each class from the training set. On some MRI images you can easily notice the presence of a tumor and even its class, but there are also pictures for which it is difficult to come to a definite answer unambiguously.

To create `train_gen`, `test_gen`, `val_gen` we use our `x_test`, `x_train`, `y_test`, `y_train`(`x` for images and `y` for their labels). Also, we rescaled all our images by `1./255` and resize them to `(128,128)` for both Testing and Training sets, and use `width_shift_range = 0.1` and `height_shift_range = 0.1` for augmentation on the Training set. Batch size is `32` and target size is `(128, 128)`

```
data batch shape: (32, 128, 128, 3)
labels batch shape: (32, 4)
```

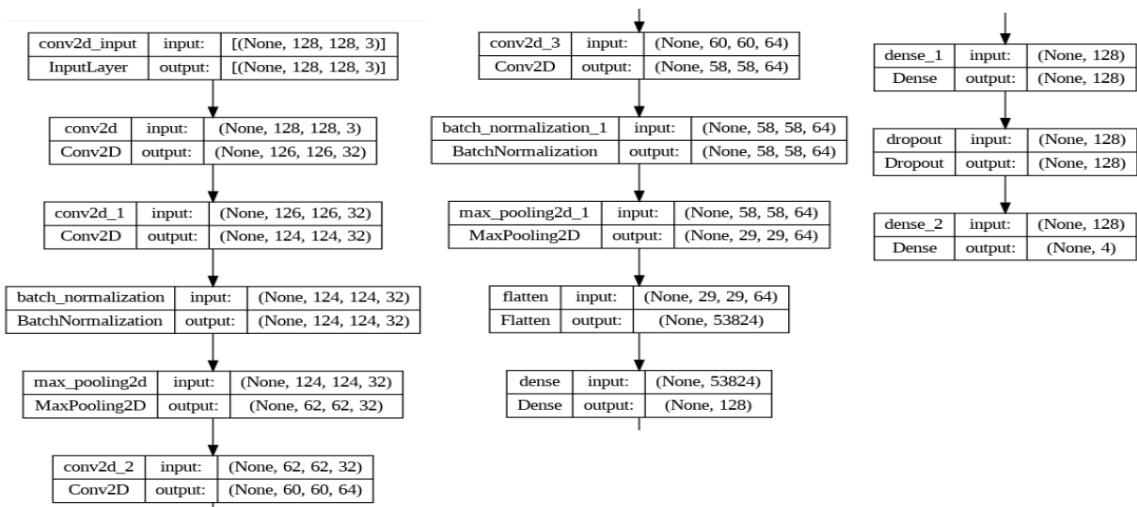
- **Description of ML models**

In our project, we used three models, as mentioned earlier, the first two models were created manually, while the third model is pre-trained with some frozen layers to fine-tune it.

Convolutional Neural Network (CNN) is well-suited for detecting features and patterns in visual data, which makes them useful for identifying and classifying objects in images. Additionally, they have been shown to be effective in classifying other types of data, such as audio, signal, and time-series data. All the models below are based on CNN. Next, we will analyze each model separately.

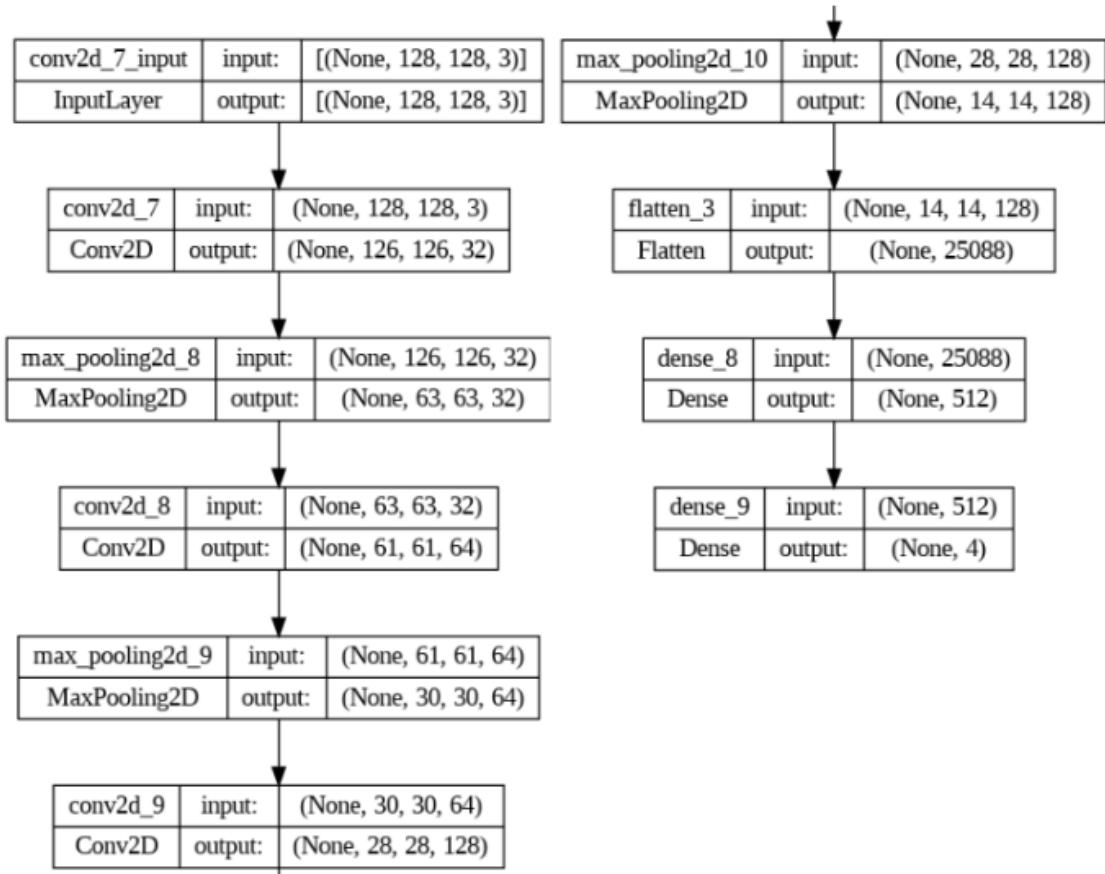
The first model consists of multiple convolutional and pooling layers, followed by fully connected layers. The input to the model is an image with a size of 128x128 pixels and 3 color channels in RGB format. The model predicts one of 4 possible classes using the softmax activation function in the output layer. Conv2D Layer performs convolution on the input image using a set of learnable filters. The first convolutional layer has 32 filters, while the second convolutional layer has 64 filters, each with a kernel size of 3x3. The activation function used in each convolutional layer is ReLU.

After, BatchNormalization Layer normalizes the output of the previous layer to improve the stability and speed of the network during training, MaxPool2D Layer performs max pooling on the output of the previous layer, reducing the spatial dimensions of the data by a factor of 2x2. Dense Layer is a connected layer with 128 neurons and ReLU activation function. Dropout was also used, it randomly drops out a portion of the neurons in the previous layer during training, which helps to prevent overfitting. And last but not least, the Output Layer with 4 neurons and softmax activation function, which outputs the probabilities of each class (glioma, meningioma, no tumor, pituitary). A visual representation of the model is below.



In general the second model can be divided into 2 large parts. The first part consists of three blocks containing convolutional layers and pooling layers. Convolutional layer with 32 filters, each with a size of 3x3, and a ReLU activation function. The input shape of the layer is 128x128x3, which corresponds to an RGB

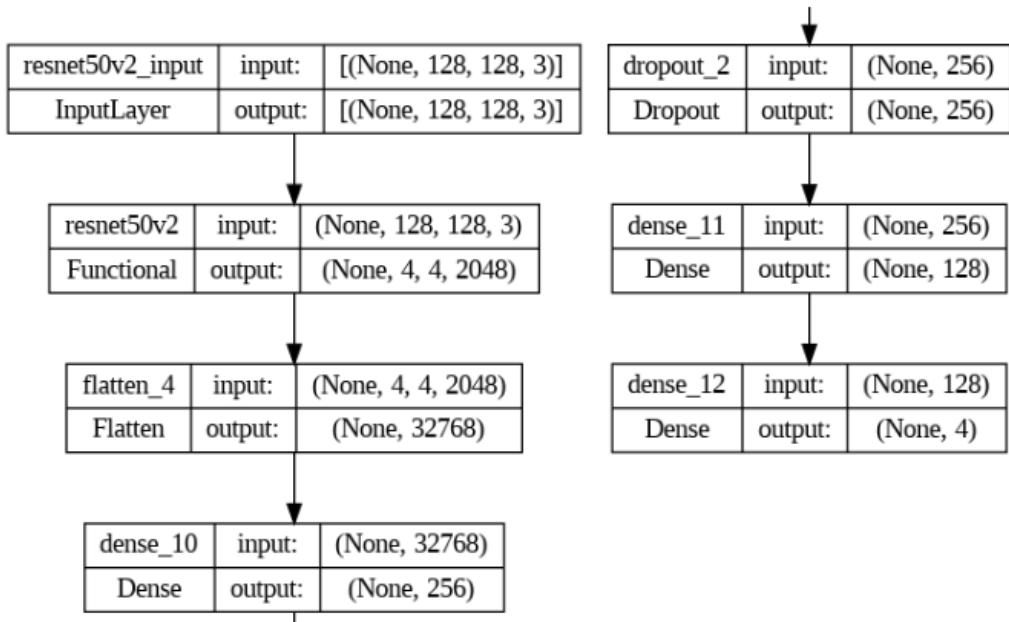
image and max pooling layer with a pool size of 2x2. Each subsequent layer is identical to the previous one, but contains 64 and 128 filters respectively. Second large part consists of flatten and dense layers, first one transforms the output of the convolutional layers into a one-dimensional feature vector and second, dense layer with 512, and 4 units and a ReLU, Softmax activation function. A visual representation of the model is below.



Third model based on the ResNet50v2 architecture. Transfer learning refers to using a pre-trained model as a starting point for a new model instead of training from scratch. In this case, we are using the pre-trained ResNet50v2 model, which has been trained on a large dataset of images to recognize a large number of classes. By using this pre-trained model as a starting point, we can achieve significantly higher performance than training with only a small amount of data.

Firstly, we load the ResNet50v2 model with pre-trained ImageNet weights, and set the top layer to be excluded from training using the parameter `include_top=False`. After that, we define a new model (model 3) and add the pre-trained ResNet50v2 model as the first layer, followed by a flatten layer.

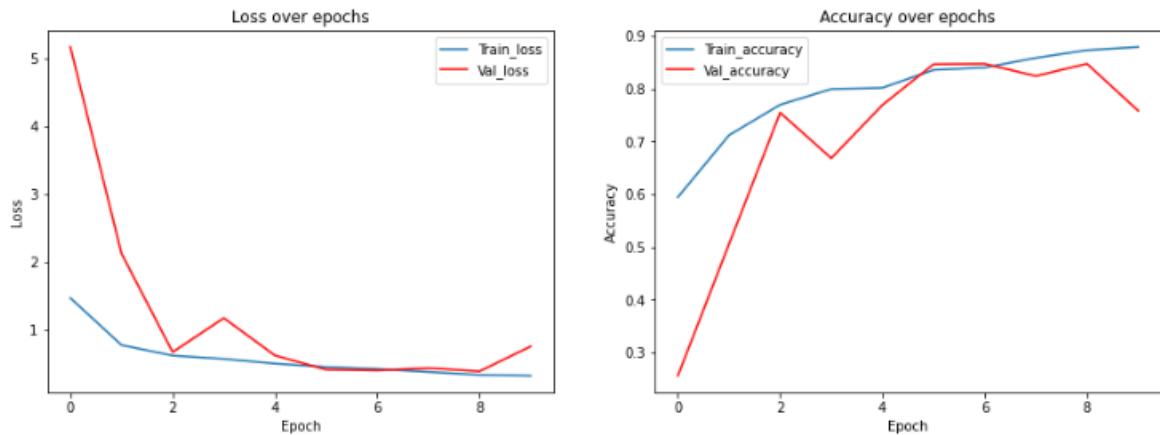
Then, we add two fully connected layers with 256 and 128 neurons respectively, with a dropout layer in between to prevent overfitting, and a final softmax output layer with 4 units for classification of tumor. A visual representation of the model is below.



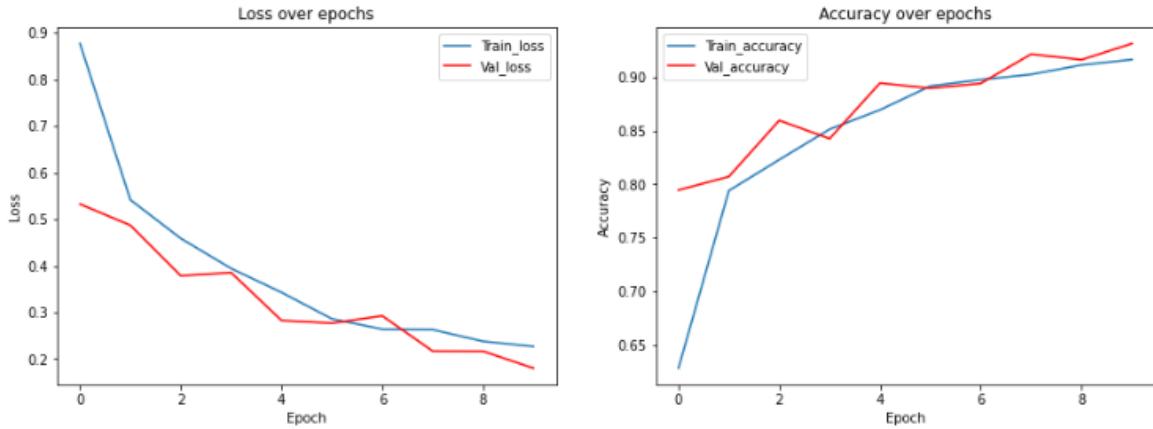
Thus, we have three trained models, and comparisons of their performance displayed below.

Results:

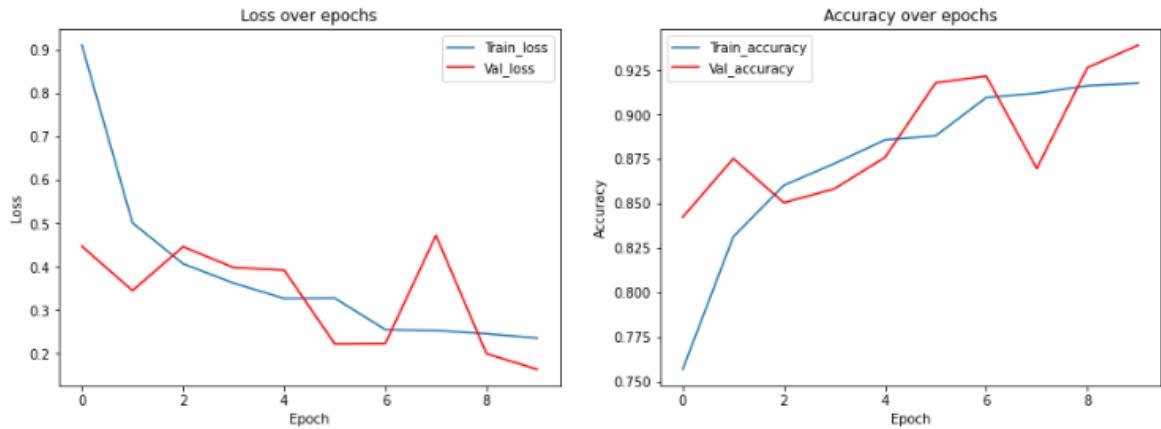
After each model we plot graphs for loss (train_loss, val_loss) and for accuracy (train_accuracy, val_accuracy) to visually see how our models trained.



As we can see after 10 epochs first model achieved accuracy 87.92% and val_accuracy: 75.86%. In general, despite small fluctuations, the accuracy graphs increase, and losses decrease. Val_accuracy was approximately at the level of 80-85% at the 6-9th epochs, but on the 10th epoch it dropped sharply back to 75%. Train loss stopped at 0.32% and val_loss at 0.75%



Next model showed a very smooth graph compared to the others. Overall, accuracy graphs without large changes are gradually increasing, and loss graphs are gradually decreasing. Accuracy reached 91.61% for train and 93.10% for validation and the loss was only 0.22% for train and 0.18% for validation.

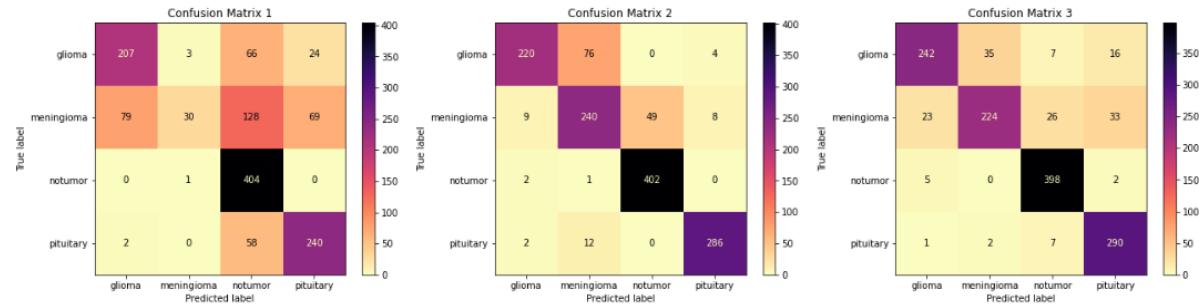


And the 3rd model shows sharper graphs compared to the previous ones with some fluctuations, but despite this, it has the best performance among the models. With results: train_accuracy - 91.77% val_accuracy - 93.91% train_loss - 0.23% and val_loss only 0.16%.

Generally, according to these graphs, we can see that the second and third models have better performance in terms of all metrics compared to the first model. Both the second and third models have lower loss values, higher accuracies and validation accuracies.

Between the second and third models, the third model has a slightly lower loss value and a slightly higher validation accuracy, indicating better performance on unseen data, which indicates that the third model is the most accurate and most robust model of the three.

Then we plot Confusion Matrix for each model to see how were all the predictions distributed



According to this matrix we can see that all 3 models easily find out no tumor class. It means that our models can detect tumors and distinguish an MRI of a healthy person from a person with a tumor.

But when it comes to the classification of the tumor, the first model represents the worst result, especially we can notice that it has problems with meningioma prediction.

On the confusion matrixes, we can clearly see the diagonal of correct predictions in the second and third models. The second model sometimes makes mistakes in predicting gliomas and meningiomas, but it already perfectly determines not only no tumor but also the pituitary.

Well, Model 3 shows excellent results with a well-defined diagonal and correct prediction for all 4 classes, but there are still minor errors in classification first two classes.

First model correctly identified 802/1311(509 was incorrect) 61%

Second model correctly identified 1148/1311(163 was incorrect) 87%

Third model correctly identified 1154/1311 (157 was incorrect) 88%

	Metrics table for First Model			
	precision	recall	f1-score	support
glioma	0.72	0.69	0.70	300
meningioma	0.88	0.10	0.18	306
notumor	0.62	1.00	0.76	405
pituitary	0.72	0.80	0.76	300
accuracy			0.67	1311
macro avg	0.73	0.65	0.60	1311
weighted avg	0.73	0.67	0.61	1311

	Metrics table for Second Model			
	precision	recall	f1-score	support
glioma	0.94	0.73	0.83	300
meningioma	0.73	0.78	0.76	306
notumor	0.89	0.99	0.94	405
pituitary	0.96	0.95	0.96	300
accuracy			0.88	1311
macro avg	0.88	0.87	0.87	1311
weighted avg	0.88	0.88	0.87	1311

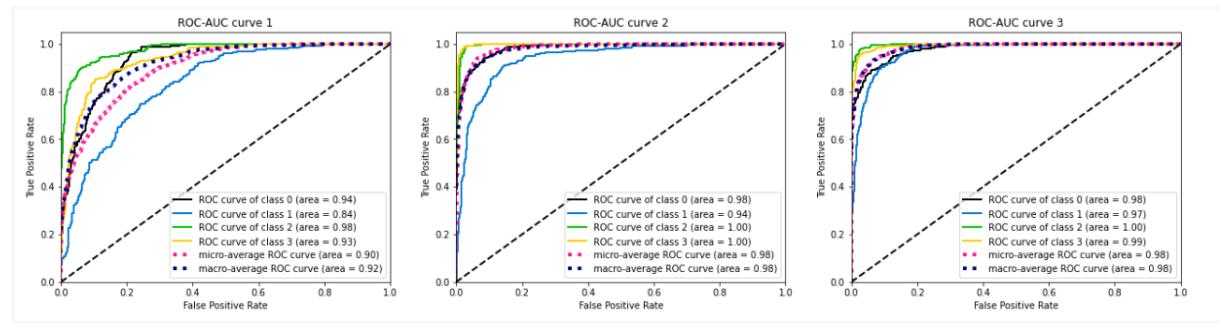
	Metrics table for Third Model			
	precision	recall	f1-score	support
glioma	0.89	0.81	0.85	300
meningioma	0.86	0.73	0.79	306
notumor	0.91	0.98	0.94	405
pituitary	0.85	0.97	0.90	300
accuracy			0.88	1311
macro avg	0.88	0.87	0.87	1311
weighted avg	0.88	0.88	0.88	1311

Also, we construct a Metrics table for our models. It reports several metrics, including precision, recall, and F1-score for each class, as well as overall accuracy and macro and weighted average metrics that summarize the performance across all classes.

The 1st model has an overall accuracy of 67%. The best performance was seen for the no tumor class and the lowest F1-score of 0.18 for the meningioma class. It indicates that the model can not identify meningioma.

The 2nd and 3rd models show significantly better performance than the first model, with overall accuracies of 88%. And with F1-score for meningioma 76%

The 3rd model has the highest F1-scores for all classes, indicating that it performs well in identifying all types of brain tumors. And with F1-score for meningioma equal to 79%



Additionally, we plot ROC-AUC curves based on the model's predictions and the true labels for the test set. Since we have a multiclass classification, it shows the ROC-AUC curve for each class and the micro-avg and macro-avg ROC curves.

Also in the first graph, you can see how the curves for the classes are slightly scattered (closer to the center the worse the results), but in others, they are located close to each other and closer to the upper left corner, which means better results when predicting classes. And 2nd and 3rd models have some classes with an AUC of 1.0, they can be considered as perfect. Overall, the curve for meningioma classification is worse in all models.

In order to finally verify the correctness of our model, a real MRI of Turganbekov Rahat's brain was uploaded to classify the tumor. In addition to this, we will contact a specialized neurologist, Daibekov I.O. to obtain a professional opinion on the finished MRI, and also asked for a comment on some MRI images from our Dataset. According to the doctor, it is not difficult to make sure that a person does not have a brain tumor, since the structure of people's brains is the same, and even if small errors differ in details, they do not mean that a person is sick. If a person has progressive tumors, it is also not a big problem to identify them.

The really urgent problem is the detection of tumors at their initial stage, when a person is unaware of any changes in his body, in such situations, conventional MRI is powerless, both for the doctor and for our model. In such cases, you need to resort to an MRI with a contrast agent that will highlight all the problem areas of the brain.

In conclusion, the doctor confirmed the correctness of the classification of diseases, but recommended focusing on more contradictory samples of photographs where the diseases are not so pronounced.

Unfortunately, the doctor agreed to give only an oral comment, linking this with a lack of time. The following proofs of the work done are presented below.

МРТ ЛИДЕР

ЦЕНТР МАГНИТО-РЕЗОНАНСНОЙ ТОМОГРАФИИ
ПРИЕМ НА ИССЛЕДОВАНИЕ ПО ЗАПИСИ

Режим работы: Круглосуточно.

ЗАПИСЬ НА ОБСЛЕДОВАНИЕ

Ф.И.О. пациента Тургебеков Раат

Возраст пациента 2002 г.

Предварительный диагноз

МСТ зон. суп. лев.

Дополнения ЖКХ

MРТ Контраст Тип носителя: Диск USB носитель

Ф.И.О. врача Ахметов Р.Р.

Номер телефона врача

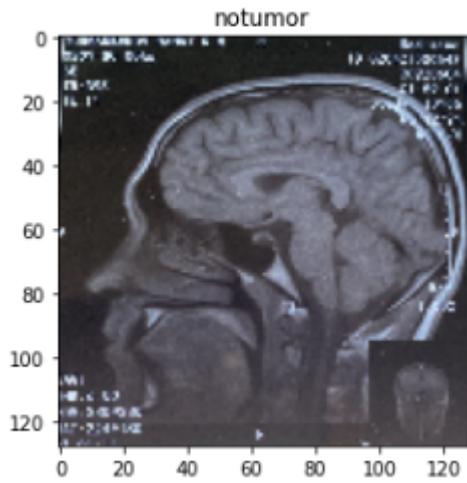
Подпись

г. Нур-Султан, район Алматы,
ул. Күнші Діна, 30
+7 (717) 277 99 01, +7 707 877 99 01
+7 (717) 277 99 07, +7 707 877 99 09
 mrt_lider_astana_nur_sultan www.mrt-lider.kz

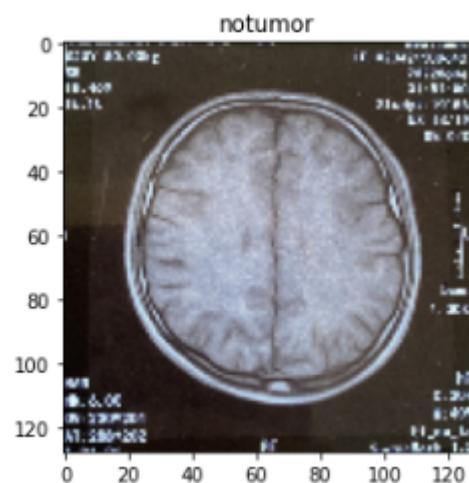


Own MRI.

Prediction: notumor



Prediction: notumor



Discussion:

- **Critical review of results**

In general, our models coped well with the task and were able to distinguish and identify brain tumors from our MRI images, and the second and third models even showed high detection accuracy, which we achieved in just 10 epochs of training. Of course, we encountered certain difficulties during the execution of the project:

- 1) There are certain difficulties in distinguishing meningiomas and gliomas, because these are two tumors of the central nervous system and both tumors affect the brain and spinal cord.
- 2) Since this was our first experience in building such models, some tasks took a very long time to complete.

3) There were also some difficulties in finding a specialist in this field who is ready to give us a comment. We had to consult with a specialist only during the display of our personal MRI image

- **Next steps**

In the future, in order to improve our work, in addition to increasing the database and using other pre-trained models, increasing the number of epochs. We want to optimize the work and make the interface more convenient and accessible to people by creating a telegram bot or a website.

We would also like doctors to directly help in the further development of our project so that we have more reliable information and conclusions. And also add other diseases besides tumors and increase the number of classes for tumor classification.

Reference:

1. *Image resizing with opencv: Learnopencv #.* LearnOpenCV. (2021, October 19). Retrieved February 19, 2023, from <https://learnopencv.com/image-resizing-with-opencv/#read-image>
2. Mayo Foundation for Medical Education and Research. (2023, February 10). *Brain Tumor*. Mayo Clinic. Retrieved February 15, 2023, from <https://www.mayoclinic.org/diseases-conditions/brain-tumor/symptoms-causes/syc-20350084>
3. GeeksforGeeks. (2021, May 31). *Python: Os.path.join() method*. GeeksforGeeks. Retrieved February 19, 2023, from <https://www.geeksforgeeks.org/python-os-path-join-method/>
4. Team, K. (n.d.). *Keras Documentation: Python & numpy utilities*. Keras. Retrieved February 19, 2023, from https://keras.io/api/utils/python_utils/#to_categorical-function
5. *Tf.math.argmax : tensorflow V2.11.0*. TensorFlow. (n.d.). Retrieved February 19, 2023, from https://www.tensorflow.org/api_docs/python/tf/math/argmax
6. Scikit-Learn. (n.d.). *Scikit-learn/confusion_matrix.py at 8c9c1f27b7e21201cffb118934999025fd50cca · scikit-learn/scikit-learn*. GitHub. Retrieved February 19, 2023, from https://github.com/scikit-learn/scikit-learn/blob/8c9c1f27b/sklearn/metrics/_plot/confusion_matrix.py#L339
7. *How to plot ROC curve in python*. Stack Overflow. (1961, July 1). Retrieved February 19, 2023, from <https://stackoverflow.com/questions/25009284/how-to-plot-roc-curve-in-python>
8. Кодкампа, Редакция (2022, August 17). *Как интерпретировать отчет о классификации в sklearn (с примером)*. кодкамп. Retrieved February 19, 2023, from <https://www.codemcamp.ru/blog/sklearn-classification-report/>
9. Lang, N. (2022, October 24). *Using convolutional neural network for Image Classification*. Medium. Retrieved February 20, 2023, from <https://towardsdatascience.com/using-convolutional-neural-network-for-image-classification-5997bfd0ede4>
10. Google. (n.d.). *Практикум по машинному обучению: классификация изображений | machine learning | google developers*. Google. Retrieved February 20, 2023, from <https://developers.google.com/machine-learning/practices/image-classification/convolutional-neural-networks?hl=ru>
11. *What is transfer learning? [examples & newbie-friendly guide]*. What Is Transfer Learning? [Examples & Newbie-Friendly Guide]. (n.d.). Retrieved February 20, 2023, from <https://www.v7labs.com/blog/transfer-learning-guide>
12. Keita, Z. (2022, September 21). *Classification in machine learning: A guide for beginners*. DataCamp. Retrieved February 20, 2023, from <https://www.datacamp.com/blog/classification-machine-learning>

