Project proposal AI

Tjeerd Morsch S4567013, Ebe Kort S4143299, Senne Hollard S5315751

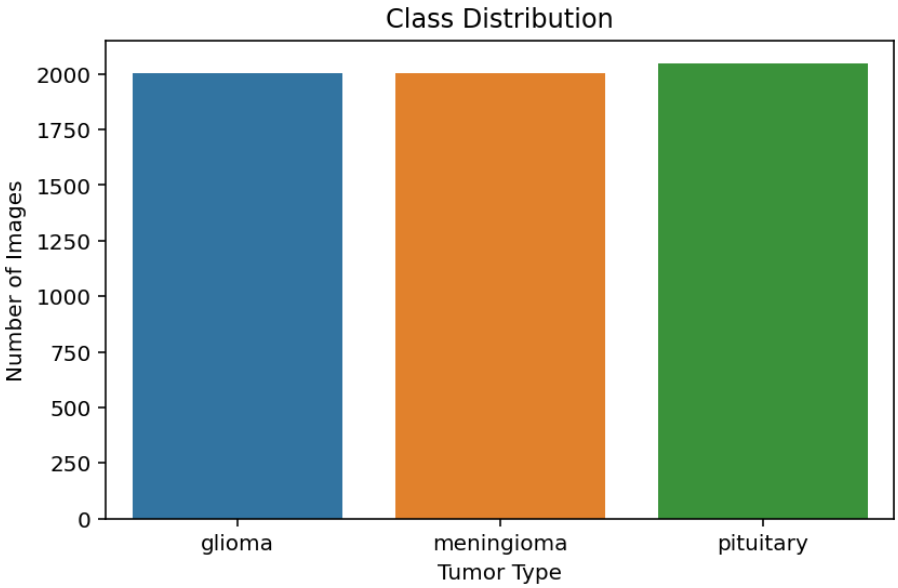
April 2025

1 Preliminary domain knowledge

When a patient is suspected of having a brain tumor, radiologists typically use MRI scans to examine the brain for abnormal growths. However, interpreting these scans is time-consuming, subjective and prone to human error. Since early detection and correct classification of brain tumors is crucial for a good treatment outcome, reliable/explainable Machine Learning solutions could significantly improve patient prognoses through faster and more consistent identification of tumor types.
During this project we aim to build a supervised MultiClass Machine Learning model which can classify three different type (glioma, meningioma, pituitary tumor) of brain tumors based on MRI scan images. To ensure that our model is trustworthy and applicable in a real-world situation we will also incorporate explainability techniques.
We will use a brain cancer MRI dataset from Kaggle.com. This dataset contains 6056 labeled MRI scan images of the three types of brain tumors. Existing medical knowledge suggests that brain tumors can differ in typical location, texture and shape.

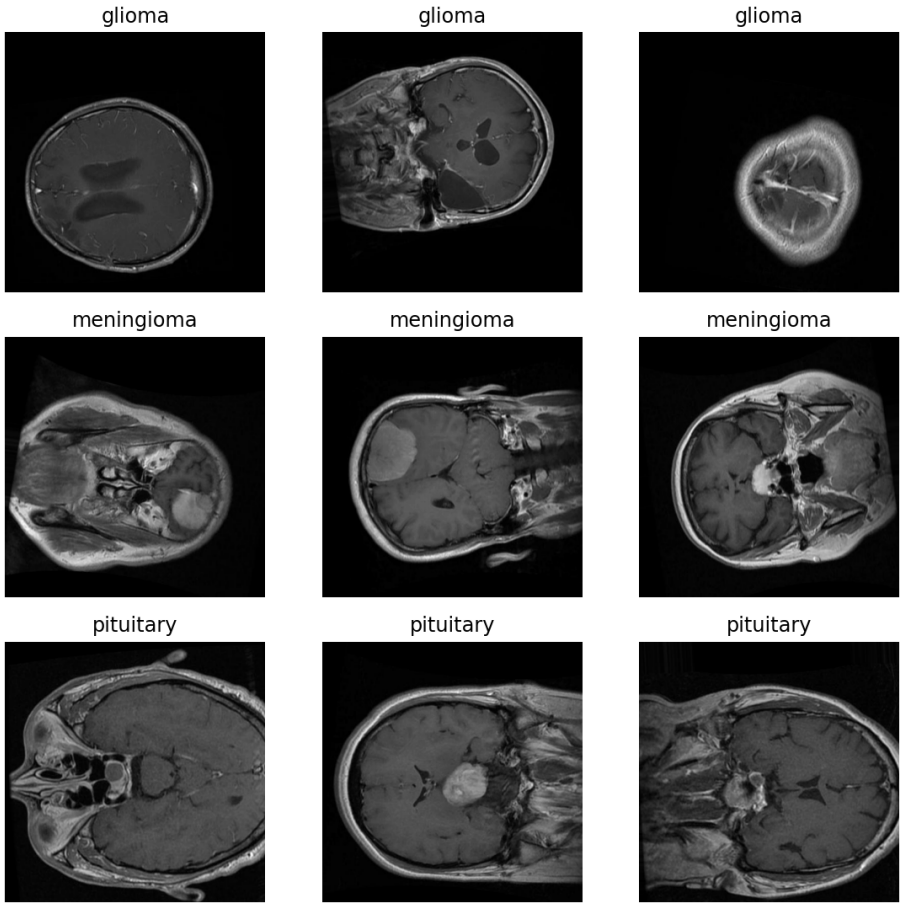
2 Preliminary data exploration

The dataset consists of 6056 grayscaled MRI scan images. All images are one of three types of brain tumors: glioma, meningioma or pituitary tumor. All images are already uniformly resized to 512x512 pixels by the creators of the dataset, this resolution was chosen by medical professionals.



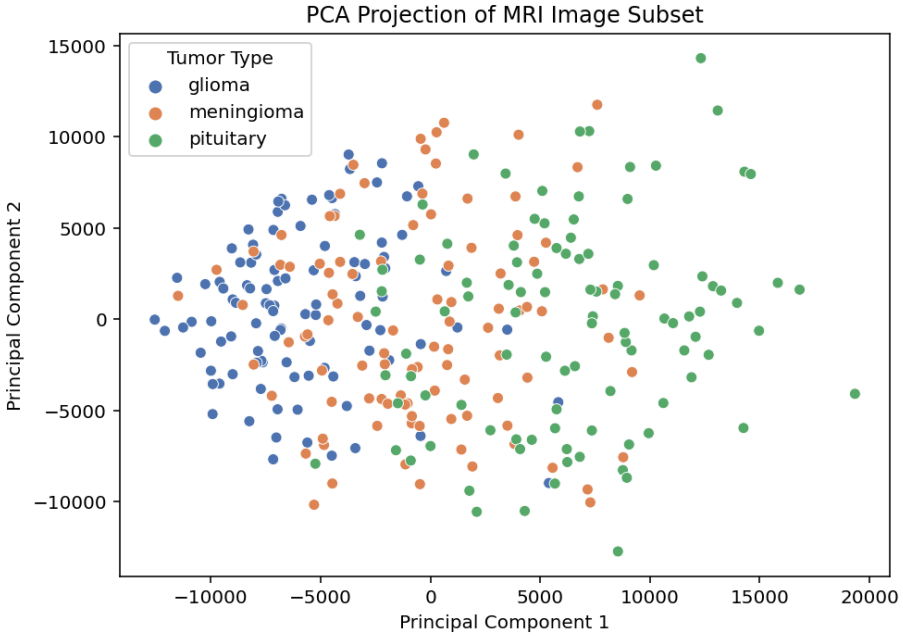
As seen by the class distribution plot we can see that the dataset is balanced (e.g. there are approximately as much images for each type of brain tumor). This is a good characteristic of the data, since this ensures that the trained models are less likely to be biased to a single class due to over-representation.

Figure 2: Sample MRI Scans by Tumor Type



In this graphic we show representative pictures for each class. Here we can indeed see that the different classes differ in location, texture and shape. This confirms that the idea of visual classification of these classes is appropriate for this dataset.

Next we explore the feature space, where we apply Principle Component Analysis (PCA). We apply PCA only to a subset of the images since using the complete dataset for this is computationally to demanding.



While we see that some class overlap, the analysis reveals to us that there is some partial clustering. This again confirms to us that the classes pertain visual differences which are seperable, thus we obtain green flags for using classification.
We have no missing or corrupted images in the dataset. Overall the dataset seems clean, balanced and visually seperable. This comes to no surprise to us as this dataset is graded 10.0 for its usability.

3 Proposed preprocessing

Based on the insights from the data knowledge and exploration we know what preprocessing steps we need to follow.

- We know that all images are already normalized and grayscale. However, we still need to check whether this is actually the case.
- We will need to normalise to pixel intensity to range [0,1]. This will help to improve the numerical stability.
- We will need to split the data into train, test, and validation sets, with percentages 70%, 15%, and 15% respectively. We need to make sure that the classes stay balanced in these sets, this is very important for the multi-classification.

- We can use data-augmentation to increase robustness and reduce the probabilities for overfitting. So by randomly rotating zooming and flipping some images we can do this.
- We will one-hot encode the classes.
- We will resize the images to 224 by 224 pixels since we will be using resnet as a backbone which is trained on ImageNet which uses pixels of that size.

4 Proposed model + baseline

To tackle the problem described in the previous section, we will implement a multiclass classification model. As a backbone we will use a pretrained Resnet50 which is trained on ImageNet, this choice was made since Imagenet contains medical (radiology and pathology) images. Our aim is to apply transfer learning so our model can use the feature extraction done by resnet. We hope that the low and mid level features in ResNet are transferable to the MRI images we have to classify.

On top of this backbone we will be building a classification head. We will use global average pooling to get a vector of a certain length, instead of flattening all features to reduce the length of the vector. Next we will experiment with different numbers and sizes of hidden layers to determine what works best. Finally, the last layer will have 3 nodes, classifying each different class. The activation functions in the hidden layers will use the ReLU activation and the last layer will use a softmax function to output the different probabilities of the classes while making sure they add up to one.

To make sure we use the low and mid level features of the ResNet backbone we will initially freeze some of the layers of ResNet. After a while we will unfreeze them and keep on training the model.

As a baseline we will train a SVM to compare the results of our model.

5 Proposed evaluation

We will be using different metrics to evaluate our model. We will calculate the accuracy to see how our model performs overall, since our classes are somewhat balanced. Additionally, we will be calculating the recall, precision and F1-score per class to determine where our model performs well and where it performs worse. Finally, a confusion matrix will be made to visualize the correct classifications and misclassifications.

We will also be doing some explainable AI to see if the model actually learns features that are part of the tumor regions in the MRI images. We will be doing this using Grad-Cam. Additionally, we will be doing feature visualization to see what the feature mappings of our model are actually learning to try and interpret whether our model is actually learning the right features. As these methods are biased due to human interpretation we will be trying to implement

some quantitative evaluation methods with regards to explainability mentioned in [1].

- correctness: First of we will be doing a Model Parameter Randomization check. We will create a another model with random weights to see if the explanation is different from our trained model to make sure that the explanation is not independent of the model. We will also be doing single deletion to evaluate the change in the output.
- completeness: A preservation check will be done too. We will be given the explanations as data to our model to check whether the same output will be given by the model as the original image.
- continuity: We will check whether the explanations are similar for similar inputs. We will apply data augmentation and check how similar the explanations are.
- contrastivity: A check will be done on whether the explanations of classes differ from one another.
- compactness: We will calculate the size and overlap of explanations and determine the counterfactual compactness to see how many input features need to change to change the prediction of the model.
- composition: We will measure how similar the explanations are to the real data.
- confidence: We will measure how well the confidence of our model aligns with the explanations.
- coherence: We will evaluate the agreement of the methods described above.

6 Model usage

Our model will mainly be used by people working in the healthcare sector. For example radiologists can use this model in a complementary way or use it as a second opinion when identifying different brain tumors. The model will not only be outputting the predicted class. Additionally, it will output the probabilities of each class and return a heatmap generated by Grad-Cam. In this way radiologists, for example, can check this region and determine whether the model was actually right or wrong. Additionally scores will be returned based on the quantitative explainability evaluation methods described in the previous section. This model acts as a complementary method and not as a replacement for clinicians identifying brain tumors. It helps them to improve accuracy, consistency and be more efficient.

Additionally, our model could be used by students. As students are less experienced in the field, they could have a harder time identifying brain tumors.

Our model will help them to point out where to look using grad-cam to try and help them learn.

7 Risk assessment

We only have 6056 images available. This is a relatively low number of images and when we train a model on this dataset it could learn the wrong features because of this. It could for example be the case that these images are not a good representation of the whole population. Therefore, there is a probability that the model overfits on the training-data. If this is the case we could try to reduce the overfitting with regularization techniques.

There is also a risk that some of the images are unclear. This could be the case for the whole image or only for the brain tumor part. We did a manual inspection on the dataset and it looks like the images are clear enough. However, it is still possible that there are images in the dataset that harm the training of the model. There is also a possibility that some of the tumors in the dataset are in a very early stage. We expect that the model will struggle with classifying these images. This could also result in the model learning the wrong features.

8 Individual learning outcomes

Ebe: I want to learn working with Imagenet. I have read about it and always wanted to try it. I think it will be very helpful for future projects.

Senne: I'm really interested in building the model, experimenting with the different number of layers and seeing what works best.

Tjeerd: I'm really interested in the explainability part of the project. I'm looking forward to use Grad-Cam and learn how it works!

9 Relation to grading specification

The novelty of our project lies in combining image classification with explainability. This means that it could be used in a real life setting to assist medical experts. This has been done in the past, but we try a new approach by combining a pretrained ResNet50 backbone with custom classification layers.

The hyperparameters we will primarily focus on are the learning rate of the neural network, the number of hidden layers in the network and the number of nodes per layer in a network.
We implement several preprocessing techniques as described in section 3 such as normalization and splitting the data into train, test and validation sets. Furthermore we will use data-augmentation to increase robustness.

To boost our project we implement the classification head on the pretrained Resnet50. Furthermore, we utilize Grad-CAM to enhance model explainability.

10 Member contributions

We first went over all questions together. We made some decisions about the project and made sure that we are all on the same page. Then we divided the sections. Senne wrote down section 1-3, Ebe 4-6 and Tjeerd 7-10.