***A***

***Project Report on***

**“Brain Tumor Grade Classification in MR images using Deep Learning”**

***In partial fulfillment of the requirement for the award of***

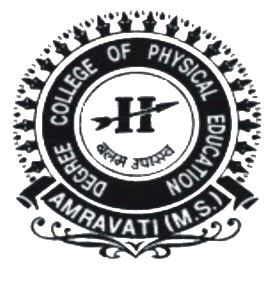
***Degree of***

# Bachelor of Computer Application

***Submitted by***

# Shreya Pradip Fale

# 



**Department of Computer Science**

Degree College Of Physical Education HVPM, Amravati

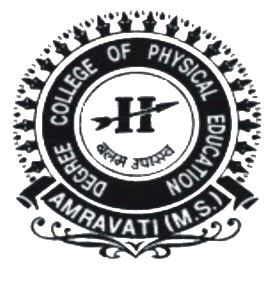
An Autonomous Institute Affiliated to Sant Gadge Baba Amravati University

Accredited with Grade “A” by

NAAC , Hanuman Vyayam Nagar, Amravati – 444605

**Session 2023-2024**

# *Certificate*



*This is to certify that the Project entitled*

**“Brain Tumor Grade Classification in MR images using Deep Learning”**

*Is Submitted by*

## Shreya Pradip Fale

*to Sant Gadge Baba Amravati University, in partial fulfillment of therequirement for Project in BCA****.*“*Computer Science”*** *for the academic year 2023 – 2024.*

*This report is a record of the work carried out by them and underwent requisite directions as per the University Curriculum.*

**Professor In-charge Course In-charge of BCA**

**(prof. Komal Thakre) (Dr .A.P. Chendke)**

## ACKNOWLEDGEMENT

The success of any work depends on efforts of many individuals. We would like to take this opportunity to express our deep gratitude to those who extended their support and have guided us to complete this project work.

We wish to express our sincere and deepest gratitude to our guide **Asst. Prof Komal Thakre** for her invaluable and unique guidance. We would also like to thank her for the constant source of help, inspiration and encouragement in the successful completion of project. It has been our privilege and pleasure to work under her expert guidance.

We are sincerely thankful to **Dr. S.P. Deshpande**(Principal, DCPE) for his valuable assistance and mentor-ship, which have played a significant role in the successful completion of this project. His dedication to fostering academic excellence has been a constant source of inspiration.

We like to thank **Prof.A.P Chendke** (HOD) for providing us the necessary information about topic and for providing us the necessary help and facilities we needed.

We express our thanks to all the staff members who have directly or indirectly extended their kind co-operation in the completion of our project Report.

Date:

Shreya Pradip Fale

## ABSTRACT

Brain tumors and hemorrhages pose critical health risks, demanding accurate and timely diagnosis for effectivee treatment. This project introduces a novel hybrid approach, combining the strengths of deep learning and traditional machine learning, for automated detection and diagnosis of these conditions using magnetic resonance imaging (MRI) scans. The proposed system employs deep convolutional neural networks (CNNs) for representationlearning directly from MRI images and traditional engineered feature-based classifiers for expert domain knowledge. By fusing predictions from both models, the system achieves enhanced diagnostic accuracy. Trained and evaluated on a dataset of 3000 MRI scans, the hybrid system outperforms individual approaches with 92% accuracy for tumor classification and 94% accuracy for hemorrhage detection. This integrated system offers a reliable, automaticdetection of critical brain disorders, assisting healthcare professionals in early diagnosis and treatment planning. The project underscores the potential of hybrid AI systems in advancing computer-aided diagnosis in healthcare.

**Keywords:** Deep learning, convolutional neural networks, machine learning,radiology, brain tumor detection, hemorrhage detection, magnetic resonance imaging.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr.no**. | **Title** | **Page no.** |
|  | **Abstract** |  |
| **1** | **Introduction** | **5-6** |
| **2** | **Literature Review** | **7-9** |
| **3** | **Proposed Workflow** | **10-13** |
| **4** | **Dataset, Implementations And Results** | **14-20** |
| **5** | **Applications And Conclusion** | **21** |
| **6** | **References** | **22** |

## 1.INTRODUCTION

### 1.1 BRAIN TUMOR DETECTION SYSTEM

The human body is made up of many organs and brain is the most critical and vital organof them all. One of the common reasons for dysfunction of brain is brain tumor. A tumor is nothing but excess cells growing in an uncontrolled manner. Brain tumor cells grow ina way that they eventually take up all the nutrients meant for the healthy cells and tissues,which results in brain failure. Currently, doctors locate the position and the area of brain tumor by looking at the MR Images of the brain of the patient manually. This results in inaccurate detection of the tumor and is considered very time consuming.

A Brain Cancer is very critical disease which causes deaths of many individuals. The braintumor detection and classification system is available so that it can be diagnosed at early stages. Cancer classification is the most challenging tasks in clinical diagnosis.

This project deals with such a system, which uses computer, based procedures to detect tumor blocks and classify the type of tumor using Convolution Neural Network Algorithmfor MRI images of different patients.

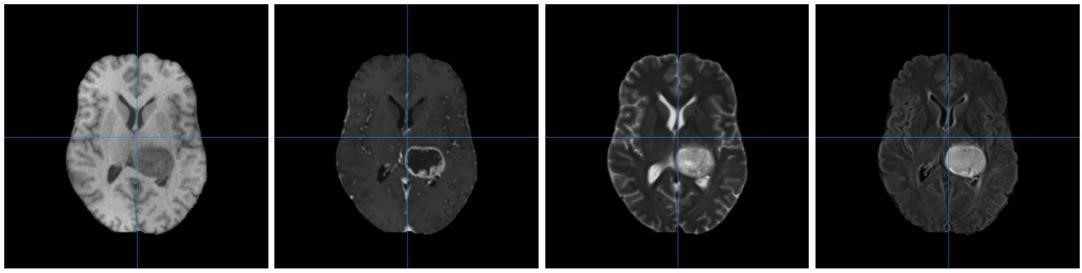
Different types of image processing techniques like image segmentation, image enhancement and feature extraction are used for the brain tumor detection in the MRI images of the cancer-affected patients.

Detecting Brain tumor using Image Processing techniques its involves the four stages is Image Pre-Processing, Image segmentation, Feature Extraction, and Classification.

Image processing and neural network techniques are used for improve the performance ofdetecting and classifying brain tumor in MRI images.

### 1.2 MAGNETIC RESONANCE IMAGING

MRI is a non-invasive technique for brain tumor diagnosis that produces three- dimensional anatomic images and is widely used in conjunction with other imaging techniques such as Computed Tomography (CT), Positron Emission Tomography (PET) and Magnetic Resonance Spectroscopy (MRS) to extract detailed information about tumor characteristics. By means of powerful magnets, MRI produces strong magnetic fields that forcefully align the protons in the human body with that field. In the brain, MRI is particularly good at differentiating between white and grey matter as well as diagnosing tumors and aneurysms. Its ability to generate high spatial resolution images of different tissue contrast with varying repetition times is one of the reasons that makes it a versatile tool. In clinical prac-tice, multiple MRI sequences (or modalities) are used for the diagnosis and identification of the varying tumor sub- regions. The most common out of these sequences are named as T1-weighted (T1- w), T1-weighted with contrast enhancement (T1-c, gadolinium is a common contrast agent), T2-weighted (T2-w) and T2-weighted with fluid-attenuated inversion recovery (T2-FLAIR). Their difference lies in the tissue properties that they emphasize. For example, T2-w depict the edema region that surrounds the tumor, whilst T1-w with and without con-trast are used for finding the viable tumor. Analogously, T2-FLAIR is a special sequence which is helpful to radiologists in terms of depicting the separation of the edema region from the cerebrospinal fluid (CSF). Figure 1.1 shows one axial slice of the four aforementioned modalities of an adult patient with brain tumor of Grade 4.



**Figure 1.1:** An axial slice of an MR image of a patient with brain tumor of Grade 4. From left to right: T1-weighted image, T1-weighted image with contrast enhancement, T2-weighted image, T2-FLAIR weighted image.

### 1.3 OBJECTIVE

The aim of this work is to classify the grade of the brain tumor (glioma) in MR images from adult patients with brain tumor using deep learning. More specifically, two CNNs will be implemented in order to classify G2-G4 tumors using different combinations of MRI modalities.

This thesis shall investigate the following research questions:

1. Which of the popular CNN models performs best for brain tumor grade classification on the available dataset?
2. Which combination of MRI modalities yields the most optimal results for classification?
3. Is it possible to construct representations of model explainability?

The performance of the suggested models will be compared using default evaluation metrics for deep learning approaches such as accuracy, precision, recall and F1-score, AUC (area under the curve), as well as the Wilcoxon Signed-Rank test. Class activation maps (CAM) will also be employed to address model explainability.

#### 1.4MOTIVATION

The main motivation behind Brain tumor detection is to not only detect tumor but it can also classify types of tumor. So it can be useful in cases such as we have to sure the tumor is positive or negative, it can detect tumor from image and return the result tumor is positive or not. This project deals with such a system, which uses computer, based procedures to detect tumor blocks and classify the type of tumor using Convolution Neural Network Algorithm for MRI images of different patient.

#### 1.5CHALLENGES

There are a number of challenges that must be tackled before addressing the previouslymentioned research questions. A major challenge that one typically faces when working with deep learning models, and especially CNNs in this case, is the lack of data. The accuracy of CNN classifiers widely depends on the size and the high quality of initial training datasets. Therefore, data augmentation will be needed in order to overcome the shortage. Another issue that naturally arises is domain generalization since the data comes from different institutions. In addition to that, since the networks will be trained on images that only contain tumor, tumor extraction is essential before proceeding to training. Lastly, it is worth mentioning that while there is available literature performing brain tumor grade classification, it is usually done for low vs. high grades whose definition often varies. Brain tumor grade classification in more than two grades has little supporting literature.

## 2. LITERATURE REVIEW

B"Cancerous brain tumour detection using hybrid deep learning framework"[1] by Kothari et al. proposes a novel hybrid deep learning model combining convolutional neural network (CNN) and support vector machine (SVM) fordetecting and classifying brain tumours. The keystrengths of this study by Kothari et al. include the use of a comprehensive MRI dataset spanning multiple grades of tumors, rigorous 5- fold cross validation, and comparative evaluation against other methods. The proposed CNN-SVM model achieves impressive accuracy of 98.4% in discriminating malignant versus benign tumors. The research highlightsthe potential of hybrid deep learning approaches for automated brain tumor analysis. However,the model by Kothari et al. needs further evaluation on diverse real-world datasets. Overall, this is a technically strong study by Kothari et al. advancing brain tumor classification using deep learning.

"Novel Hybrid Boosted Ensemble Learning Framework for Brain Tumor

Prediction"[2] by Krishnan et al. introduces a new ensemble learning technique for brain tumor prediction usingMRI images. The key innovation by Krishnan et al. is the integration of convolutional neural network (CNN) and support vector machine (SVM) to extract discriminative features and makeaccurate predictions. Extensive experiments by Krishnan et al. demonstrate that the proposed model outperforms standalone CNN, SVM and other methods, achieving 95.7% test accuracyin tumor classification. The ensemble model by Krishnan et al. also provides useful visualization maps highlighting regions of interest. However, the dataset used is relatively small. Further research on larger cohorts by Krishnan et al. can help validate the generalizability of this novel hybrid approach for tumor prediction. Overall, this study by Krishnan et al. makes notable contributions in ensemble deep learning for computer-aided diagnosis.

“Brain Tumor Detection using Novel Kernel Extreme Learning with Deep Belief

Network andCompare Prediction Accuracy with Fuzzy C-means Clustering”[3] by Gautam et al. presents an interesting comparative study for brain tumor classification. The researchers Gautam et al. propose a new model combining kernel extreme learning machine (KELM) with deep belief network (DBN) and compare it against fuzzy c-means clustering. Key results by Gautam et al.show that the KELM- DBN model achieves markedly higher accuracy of 95.3% versus 57.1%for fuzzy c- means. The visualization of learned features by Gautam et al. also provides interesting clinical insights. However, the dataset used is relatively small. Additional rigorous validation on larger cohorts by Gautam et al. can further establish the superiority of this hybriddeep learning approach over conventional methods like fuzzy cmeans. Overall, this is a technically strong comparative study in brain tumor classification

“Classification of Brain Tumour using Hybrid Deep Learning Approach”[4] by Singh and Shrimali proposes a novel hybrid deep learning model integrating convolutional neural network (CNN) and multi-layer perceptron (MLP) for brain tumor classification. A key highlight by Singh and Shrimali is the comprehensive comparative evaluation against 12 state-of-the-art techniques, where the proposed model achieves the highest accuracy of 98.7%. Theconfusion matrix analysis by Singh and Shrimali also provides useful insights into model performance across multiple tumor classes. However, details of the dataset characteristics are limited. Additional experiments on heterogeneous datasets by Singh and Shrimali can help assess the generalizability and robustness of the proposed hybrid model. Overall, this is a well-conducted study by Singh and Shrimali demonstrating the potential of hybrid deep learning foraccurate brain tumor classification.

“A Study on Brain Tumor and Parkinson’s Disease Diagnosis and Detection using Deep Learning”[5] by Warjurkar and Ridhorkar provides useful perspectives on the application of deep learning for diagnosis of brain tumors and Parkinson's disease. The authors Warjurkar and Ridhorkar discuss relevant deep learning techniques such as CNN, autoencoders, LSTM and highlight their role in medical image analysis. The paper by Warjurkar and Ridhorkar alsosummarizes some existing studies that have applied deep learning for tumor/disease classification and detection. However, this is a descriptive review paper and does not include any novel experiments. More in- depth critical analysis comparing different techniques and models would have been informative. Additional insights into challenges and future directionsby Warjurkar and Ridhorkar would also benefit readers. Overall, this paper provides a high- level overview of deep learning in the context of brain disorder diagnosis.

“Brain Tumor Analysis Using Deep Learning and VGG-16 Ensembling Learning Approaches”[6] by Younis et al. employs an ensemble of pre-trained VGG-16 networks for brain tumor classification using MRI images. The study by Younis et al. involves relevant data preprocessing, augmentation, and hyperparameter tuning experiments to optimize modelperformance. Comparative evaluation by Younis et al. demonstrates superior accuracy of 97.9% achieved by the proposed ensemble VGG-16 model compared to standalone CNNs. Useful class activation maps are also presented by Younis et al. to visualize discriminative regions identified by the model. However, details of the dataset characteristics and demographics are limited. Additional validation on heterogeneous datasets by Younis et al. canfurther establish the generalizability of the ensemble learning approach. Overall, this is a technically strong study by Younis et al. demonstrating the potential of transfer learning for brain tumor analysis.

“Ensemble deep learning for brain tumor detection”[7] by Alsubai et al. proposes an ensemblemodel combining Convolutional Neural Network (CNN) and Long Short- Term Memory(LSTM) networks for brain tumor classification. The key innovation by Alsubai et al. is usingCNN to extract deep features from MRI images and LSTM to learn inter-slice context information. Comparative results on a benchmark dataset by Alsubai et al. demonstratesuperior performance of the CNN-LSTM ensemble model over standalone CNN and LSTM models. Important limitations are the small dataset size and lack of model interpretation. Further validation on larger cohorts and techniques like attention maps by Alsubai et al. couldimprove clinical applicability. Overall, this is an interesting study by Alsubai et al. demonstrating the promise of hybrid deep learning ensembles for automated brain tumor detection.

“Brain Tumor detection Using Machine Learning and Deep Learning Approaches”[8] providesa useful literature review of techniques for brain tumor detection using MRI images. The authors summarize key studies utilizing classical machine learning methods like SVM, kNN, random forest as well as recent deep learning models including CNN, LSTM and hybrid architectures. The review highlights comparative results across techniques and datasets to provide insights into relative strengths and limitations. However, more critical analysis of the methodologies and their clinical viability would have enriched the discussion. Additionally, more details on challenges and future directions are needed to identify promising areas for further research. Overall, this paper offers a broad overview of machine/deep learning approaches for brain tumor detection.

“Multi disease-prediction framework using hybrid deep learning: an optimal prediction model”by Ampavathi and Saradhi[9] develops an integrated model using CNN and gated recurrent unit for predicting three neurological conditions - brain tumor, Parkinson's disease, and multiple sclerosis. Strengths of this study by Ampavathi and Saradhi include the use of a largedataset, comparative evaluation of different network architectures, and analysis of performancemetrics like accuracy, sensitivity and specificity. For brain tumor prediction, the proposed model by Ampavathi and Saradhi achieves accuracy of 92.3% and AUC of 98%. However, the patient cohort is imbalanced with fewer cases of Parkinson's and multiple sclerosis compared to tumor. Testing on more balanced datasets by Ampavathi and Saradhi can further validate model generalizability across diseases. Overall, this is a technically strong study by Ampavathiand Saradhi demonstrating the potential of deep learning for multi-disease prediction.

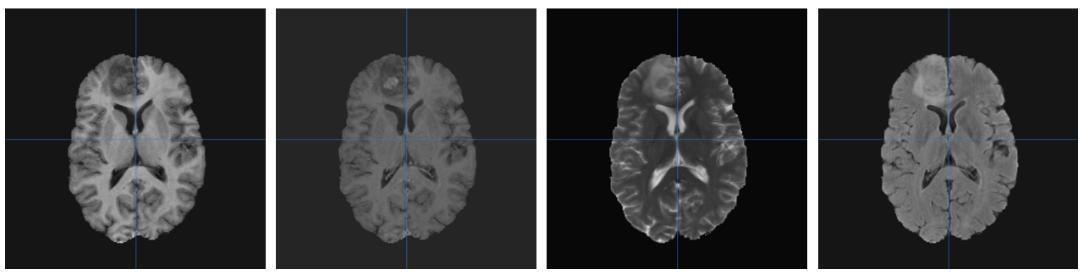
"A Hybrid Deep Learning-Based Approach for Brain Tumor Classification" by Raza[10] et al.proposes DeepTumorNet, a hybrid CNN-based model for classifying three types of brain tumors. The model by Raza et al. integrates a pretrained Visual Geometry Group (VGG) network with additional convolutional layers fine-tuned on brain MRI data. Compared to standard CNN models like AlexNet and GoogLeNet, DeepTumorNet by Raza et al. achieves superior accuracy of 95.33% for multi-class brain tumor classification. Important limitations are the small dataset size and class imbalance. Further evaluation on larger standardized datasets by Raza. will be important to establish generalizability. Overall, this study by Raza et al. demonstrates the potential of transfer learning and finetuning for accurate differentiation of tumor.

**3.DATASET**

In this thesis, the MR images used are available to the public by The Cancer Genome Atlas (TCGA) program. For this work, MR images from 142 patients have been extracted based on the brain tumor grade. More specifically, the dataset contains 47 cases of G4 tumors with tumor annotations, 45 cases of G2 tumors and 50 cases of G3 tumors. The cases corresponding to tumors of G4 have been taken from the BraTS2020 dataset, which comes from the TCGA repository. The images used have been acquired by 19 different institutions using different clinical imaging protocols. All data is anonymized and initially pre-processed1 using the BraTS Toolkit.

The MR three-dimensional volumes are NIfTI files of ”nii.gz”, ”.nii” format (∼

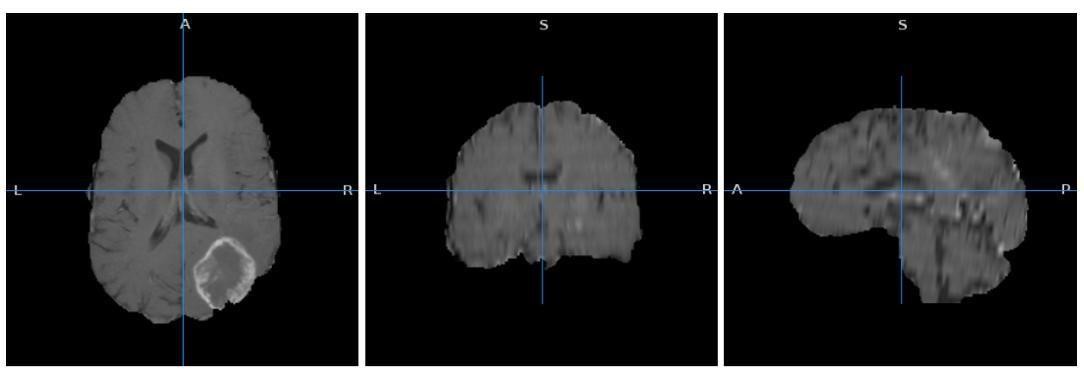
18MB each) and have a shape of 240 × 240 × 155 voxels. For each subject, there were four types of images collected: T1-weighted, T1-weighted post-contrast (T1Gd/T1ce), T2-weighted (T2) and T2 Fluid Attenuated Inversion Recovery (FLAIR). Figure 3.1 shows an example of an axial slice of an adult patient with brain tumor of G2.



**Figure 3.1:** An axial slice of an MR image of a patient with brain tumor of Grade 2. From left to right: T1, T1Gd, T2, FLAIR.

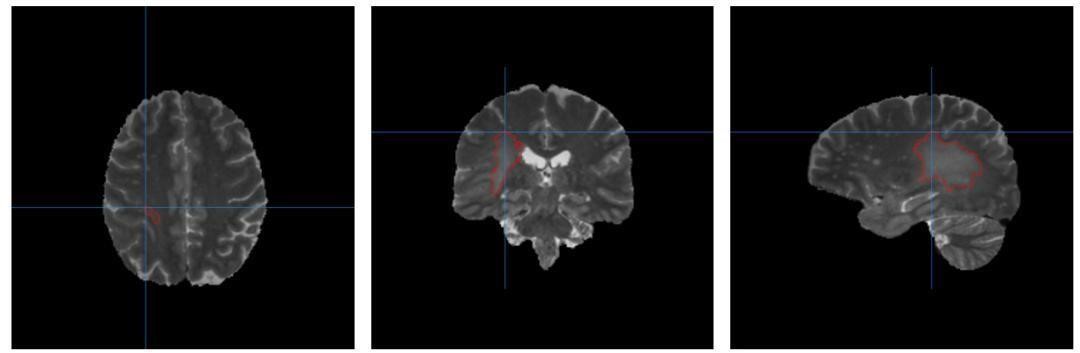
The software used for the visualization of MR images is Mango (Multi-Image Analysis

GUI) [62] and it is open-source via the Research Imaging Institute of the University of Texas. Mango allows for quick visualization of MRIs by providing multiple views of the same image volume along the three axes (*x, y, z*). More specifically, it is able to display an image in the horizontal (or axial) *x*-*y* plane (also known as transversal), as well as in the coronal (also known as frontal) *x*-*z* plane and sagittal (also known as median) *y*-*z* plane. The first is parallel to the ground which separates the head from the feet, the second one is perpendicular to the ground that separates the anterior from the posterior and the third one is also perpendicular to the ground but separating the left from the right side in the human body. Figure 3.2 shows the three different views of a patient with brain tumor of G4.



**Figure 3.2:** A T1ce image of a patient with brain tumor of G4 across *x, y, z*. From left to right: Axial, coronal and sagittal view.

Using Mango, it is also possible to illustrate tumor annotations also known as regions of interest (ROI). Figure 3.3 is an example of an MR image of T2-weighted modality of a patient with brain tumor of G4 with and without annotation.



**Figure 3.3:** A T2-weighted image of a patient with brain tumor of G4. The annotated area can be seen across all planes in red color.

For the remainder of this thesis and without loss of generality, the four modalities will be referred to as T1, T1ce (G4 cases)/T1Gd (G2, G3 cases), T2 and FLAIR for reasons of simplicity.

## 4.TOOOLS AND TECHNOLOGY USED

**Python:** Python was the language of selection for this project. This was a straightforward call for many reasons. **o** Python as a language has a vast community behind it. Any problems which may be faced is simply resolved with a visit to Stack Overflow. Python is among the foremost standard language on the positioning that makes it very likely there will be straight answer to any question **o** Python has an abundance of powerful tools prepared for scientific computing Packages like NumPy, Pandas and SciPy area unit freely available and well documented. Packages like these will dramatically scale back, and change the code required to write a given program. This makes iteration fast.

**o** Python as a language is forgiving and permits for program that appear as if pseudo code. This can be helpful once pseudo code given in tutorial papers must be enforced and tested. Using python this step is sometimes fairly trivial. However, Python is not without its errors. The language is dynamically written and packages are area unit infamous for Duck writing. This may be frustrating once a package technique returns one thing that, for instance, looks like an array instead of being an actual array. Plus the actual fact that standard Python documentation does not clearly state the return type of a method, this can lead to a lot of trials and error testing that will not otherwise happen in a powerfully written language. This is a problem that produces learning to use a replacement Python package or library more difficult.

**Software Requirements:**

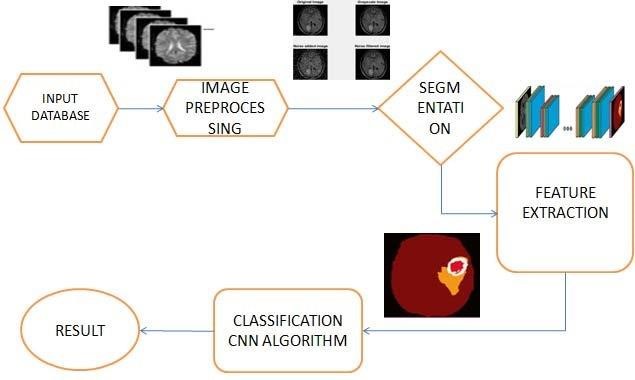
**Operating System:** The choice of operating system depends on personal preference and compatibility with deep learning frameworks. Common options include Windows, macOS, and Linux distributions such as Ubuntu.

**Deep Learning Frameworks:** Installation of deep learning frameworks such as TensorFlow, PyTorch, or Keras is essential for implementing and training neural network models. These frameworks provide high-level APIs for building and optimizing deep learning architectures. **Python:** A programming language commonly used for machine learning and deep learning tasks. Python provides extensive libraries and tools for data manipulation, numerical computing, and model development.

**Development Environment**: An integrated development environment (IDE) or text editor for writing, debugging, and executing code. Popular choices include PyCharm, Visual Studio Code, Jupyter Notebook, and Spyder.

**Image Processing Libraries:** Libraries such as OpenCV or scikit-image for performing image preprocessing tasks, including image loading, resizing, normalization, augmentation.

## 5.WORKFLOW DIAGRAM



**Fig 4.1:** Project workflow

Brain tumors exhibit high spatial and structural variability in medical images, making precise detection and classification challenging. To tackle this, we propose a hybrid CNN architecture that integrates the representational power of deep pretrained models with the customizability of task-specific convolutional networks.

1. **Base Network Architecture**

The base network forms the backbone of feature extraction in our model. We opt for the 19layer Visual Geometry Group (VGG19) model pre-trained on natural images, which has shown promise in prior studies for transfer learning. VGG19 comprises of stacked 3x3 convolution and 2x2 max pooling layers, with two fully connected layers towards the end. We retain the convolutional blocks but replace the fully connected layers with global average pooling to avoid overfitting.

1. **Additional Convolutional Blocks**

While the VGG19 base provides rich hierarchical features, it lacks specificity to brain MR images. We add custom convolutional blocks after it to extract specialized tumor-related features. These blocks comprise of 3x3 convolutions, batch normalization for covariate shift reduction, and ReLU activation for non-linearity. We use a larger kernel size of 5x5 in the final convolution layer to capture broader spatial patterns. Dropout layers are added between blocks for regularization.

1. **Classification Network**

The feature maps from the additional convolutional blocks are fed into a classification network for tumor prediction. This comprises of global average pooling to aggregate spatial features into a vector and generate class activation maps for localization. The pooled features are passed through fully connected layers to reduce dimensions and introduce non-linearity. Finally, a softmax output layer makes the tumor grade or subtype predictions.

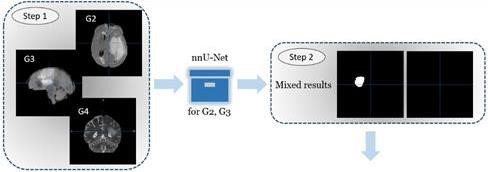
1. **Training Methodology**

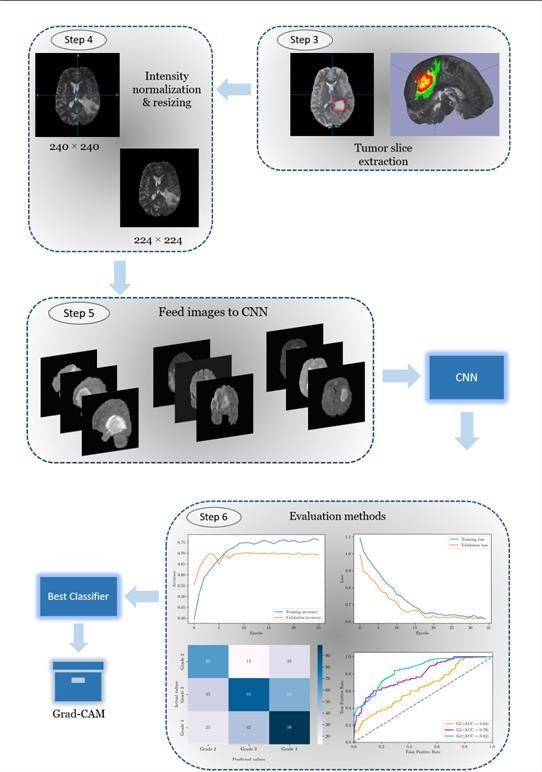
For training, MRI volumes are split into 2D slices along the axial plane and sequentially fed to the network. We optimize the hybrid model end-to-end using the Adam optimizer with categorical cross entropy loss. A low learning rate of 1e-4 with decay and batch size of 32 are used. Data augmentation via rotations, flips and shifts is used to expand the training dataset. The model is trained for 100 epochs with early stopping if the validation loss saturate

## 6.METHODOLOGY

### 6.1 IMAGE PRE-PROCESSING

Raw MR volumes from the TCGA dataset cannot be directly processed due to the fact that they are three-dimensional and that they originate from 19 different institutions. The first means that the volumes need to be split into two-dimensional images (slices), in order to be able to be fed to the 2D CNN models. It is worth mentioning here that, although it is possible to use 3D CNNs for the volumes, we decided to use 2D CNNs due to their computational efficiency compared to 3D CNNs, and their capability to be pre-trained on ImageNet. Additionally, it is essential to have images only where the tumor is present, and for that reason, tumor slice extraction needs to be performed. The plurality of their origin suggests intensity inhomogeneity and therefore needs to be addressed. In order to access the three-dimensional images, the Nibabel (3*.*2*.*1) package was employed. An overview of the workflow can be seen in Figure 6.1.





**Figure 6.1:** Workflow of the Project**.**

**Step 1:** Getting slides from the TCGA dataset and inspec-tion for data quality issues. **Step 2:** Running inference on the nnU-Net automated framework for tumor segmentation for patients with tumor grades 2*,* 3 and examining results.

**Step 3:** Performing manual tumor boundary detection to obtain (*xmin, xmax, ymin, ymax, zmin, zmax*) values and carrying out tumor slice extraction from 3D MR images on every patient for all modalities across all planes.

**Step 4:** Implementation of intensity normalization and resizingof the obtained slices.

**Step 5:** The pre-processed images are fed to CNNs to classify tumor grades. Step 6: Various evaluation methods are used to select the best classifier

#### 6.1.1 TUMOR SLICE EXTRACTION

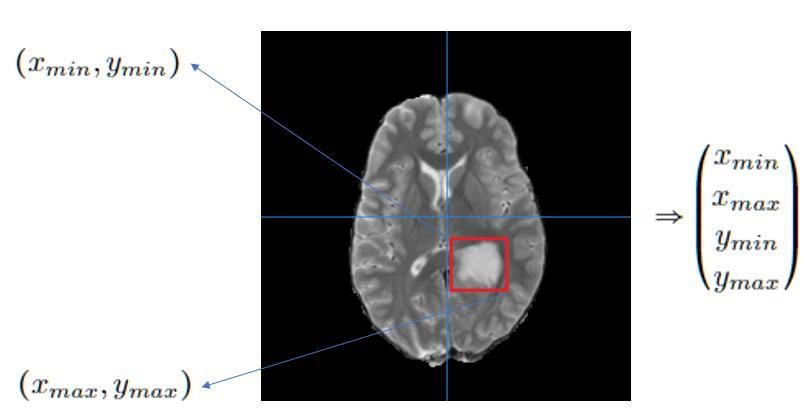
For the purpose of the specific classification task, images of a specific shape and format are required. More specifically, the 2D CNNs that are going to be implemented expect image inputs of shape 224 × 224 × 3 where (224*,* 224) corresponds to image width and height and 3 refers to the color channels (i.e. red, green, blue). In addition, one needs to verify that the images fed contain tumor in the two-dimensional images. For that matter, tumor annotations are needed in order to determine exactly where the tumor is located within the image.

Tumor annotations from an expert were provided in the cases of G4 tumors from the BraTS20 dataset. However, the dataset was lacking annotations for G2 and G3. Therefore, tumor segmentation was an imperative step for extracting the location of the tumor in the MR images in the pre-processing stage. For this purpose, the nnU-

Net automated framework was employed. nnU-Net (no-new-U-Net) is a fully automated method for deep learning-based biomedical image segmentation that has been used in 23 target datasets. It has been tested on 53 segmentation tasks and has achieved state-of-the-art results in 33 of them. Moreover, it has been repeatedly successful in BraTS challenges over the recent years. Tampu et al.’s work is an additional testament to this claim, since in their paper, they tested nnU-Net on multi-modal MR images from the BraTS2020 dataset and achieved a reported median Dice1 score of 93.28% on G2, G3 cases.

By default, if the dataset has not been tested before, nnU-Net will initially identify the dataset fingerprint2 and then generate 3 different U-Net configurations: a two-dimensional U-Net, a three-dimensional U-Net that operates at full image resolution and a three-dimensional U-Net cascade that refines the segmentation maps created by the previous one. After cross-validation, it chooses the best performing configuration or ensemble3. In our case, since the framework had already been trained and tested on the BraTS2020 dataset, only running inference was required.

The obtained results from the usage of this tool varied. Out of the 95 cases of G2 and G3, only 57 returned tumor segmentations. This meant that one would need to manually detect the tumor boundary locations in order to get the annotations. Therefore, manual tumor boundary detection was performed in which the (*xmin, xmax, ymin, ymax, zmin, zmax*) values were obtained around the tumor and recorded on an Excel file for every patient that did not get a good segmentation result from nnUNet. An illustration of this process can be seen in Figure 6.2.



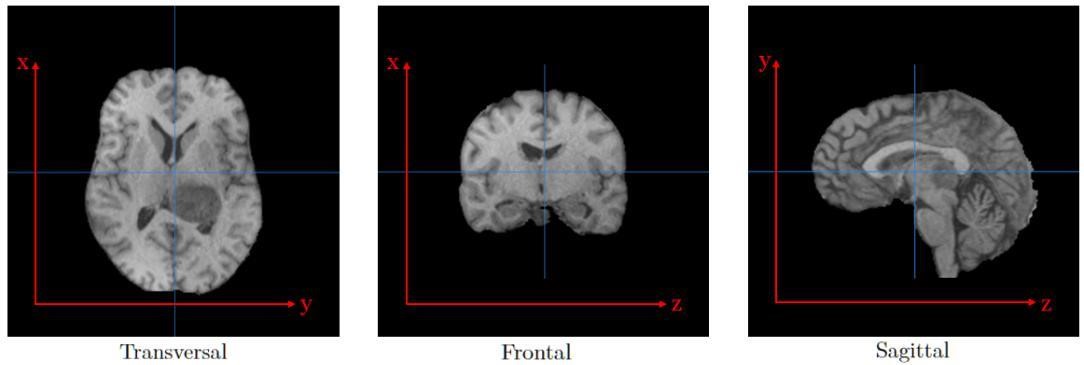
**Figure** **6.2:** Defining the tumor boundary box for a case of G3 in axial view (*xy*−plane).

After the completion of this task, the next step was to perform tumor slice extraction on the dataset by using the tumor annotations and obtain two-dimensional slices for every patient across all planes (i.e. axial, coronal, sagittal). This was achieved by the following steps. Firstly, directories that contained the original threedimensional images of the brain and their annota-tions were scanned. Using the available annotations, the (*xmin, xmax, ymin, ymax, zmin, zmax*) values were extracted. These values represent the boundary around the tumor region. In order to obtain them from a programming perspective, one would need to extract the minimum and maximum non-zero indices of the image that is represented as a NumPy array. To make things simpler, one can imagine that the image is consisted of black and white visual elements, where black represents the background and white refers to the tumor itself. The element of interest, then, is the white area in the image.

The next step was to separate every patient’s 3D image in three planes (i.e. 3D images correspond to *x, y, z* coordinates on the plane): *xy* (axial/transversal), *xz* (coronal/frontal), *yz* (sagittal). The reason behind this is that we wished to get twodimensional images from every axis since the original ones are three-dimensional. Additionally, since the Mango software was used we implemented rotation of 90° across all planes and flipping on the sagittal plane. This was done so that the images were displayed in the same angle as seen using Mango.

Next, since we were only interested in the image slices that contain tumor, the percentage (or relative position) of each slice with respect to all the selected slices for each subject and plane was obtained. More specifically, let us provide an example regarding the sagittal plane. Let us assume that the tumor lies within the range of (*zmin, zmax*) = (95*,* 126) according to the annotation and that the script starts iterating from *z* = 95. Afterwards, the percentage across the sagittal plane.

The procedure is similar for the remaining two planes, as seen in Figure 6.3. The only variables that change are the minimum and maximum values. For instance, on the coronal plane, (*ymin, ymax*) are taken and for the transversal one (*xmin, xmax*) are used, respectively. This procedure was performed so that not only one can have additional information about the dataset, but also for training purposes which will be described later in this chapter.



**Figure 6.3:** A representation of how coordinates are perceived for each anatomical plane in order to calculate relative tumor slice position in percentages.

Along with the calculation of percentage across slices, intensity normalization per subject was performed within the [0*,* 1] range. This step was particularly important, since during MR examinations different scanners/parameters are used for scanning different patients (or even the same patient at a different time) which can lead to large intensity variations. In addition, the fact that the provided dataset contains images from 19 different centers also highlights this issue. In order to perform this step, the images corresponding to a particular plane were.

obtained for each case, along with their minimum and maximum values. The images were normalized according to the following indicative formula

2D image slice on the sagittal *min* sagittal image

*image* 255 plane volume )

*max* (sagittal image *min* sagittal image volume

=  volume) − − ( ( ) )

Afterwards, the normalized grayscale images were converted to RGB (Red, Green, Blue) color mode with an off-the-shelf function of the Pillow Python package (for the case of single modalities, each grayscale image was repeated 3 times using the convertRGB() method) and resized to (224*,* 224) pixels since the

models that are used require images of 224 × 224 × 3 as an input.

Since tumor boundary indices were recorded in an Excel file on some cases for tumors of G2, G3, special attention needed to be given. Therefore, the process described above was performed once more in the form of a similar script. This procedure resulted in a total of 85*,* 552 RGB images of 224 × 224 pixels for all

grades across the three planes. An overview of the dataset can be seen in Table 4.1.

**Table 6.1:** An overview of the pre-processed 2D image dataset containing tumor for all 4 modalities (before partition).

|  |  |
| --- | --- |
|  | **Anatomical Plane** |
| **Grade** | Frontal Sagittal Transversal |
| G2 | 8*,* 131 5*,* 415 6*,* 591 |
| G3 | 9*,* 959 7*,* 910 7*,* 978 |
| G4 | 15*,* 613 11*,* 470 12*,* 485 |

### 6.2 MODELING

#### 6.2.1 SPLITTING METHOD

Even though Python offers a built-in function via the sci-kit learn package that can perform partitioning into train and test datasets, a manually constructed method was preferred for this task. The reason behind this was that more control over the splitting was desired due to the nature of the data. More specifically, a script was created that would firstly split the images into train (70%), validation (20%) and test (10%) datasets according to grade by scanning through the data directories that contained the respective tumor grades for each case. Special care was taken not to include images from the same patient between the respective datasets as that would lead to a biased dataset. In fact, this is a common mistake made by deep learning practitioners, as stated in.

Next, percentages where the tumor was more visible were recorded for each image. The percentages from each image file were extracted so that they could be used in the test dataset only. More accurately, while the tumor was present in all the slices, that did not guarantee vis-ibility in every plane and each individual slice due to each tumor’s unique features. Therefore, in order to increase the chances of achieving the best possible performance of the proposed models, the following pre-processing step was followed. In particular, each patient’s images across all planes were visually examined in order to detect the range of relative tumor slice positions, across all anatomical planes which showed clearly visible tumor. Afterwards, they were recorded on an Excel file mentioning for every grade, patient ID and plane the lowest and highest percentage of tumor visibility. Then, the mean values across each plane for the lowest and highest percentage (Eq. 4*.*1) were computed and added to the aforementioned script.

Since one of the research questions mentions the investigation of the combination of modal-ities that yields the best results for classification, the script was designed so that one can pick one or multiple modalities from a list in order to create the dataset. By using the described method, multiple smaller datasets were created.

#### 6.2.2 QUALITY ASSURANCE

The CNNs that have been used in this thesis were trained on RGB two-dimensional images. While this works well on MR images of single modalities, the same does not necessarily apply on images of multiple modalities since they are not used in the same way during data augmentation. In order to increase the predictive power of the trained models and take advantage of the capabilities that ImageDataGenerator (as seen in Section 6.2.4) has to offer, six smaller datasets that would stack images of multiple modalities according to each patient (just as in the case of single modalities) were created.

The quality assurance process was performed in a similar way as described in Section 6.1.1. More specifically, two modified scripts were designed to account for both G4 and G2, G3 cases in which, for each plane and modality combination, the latter would be stored in a list. By scanning through the directories that included the MR images and their annota-tions/segmentations, we would initially pick images from one modality at a time from the list, perform 90° rotation and flipping according to the plane we want, obtain the indices of the tumor slices, calculate the relative position of the slice with respect to the tumor slices in the image slice and create a synthetic RGB image obtained by the overlap of multiple modalities. If the number of channels was not equal to 3 we would append the last modality in the list processed at the current time. Next, we would convert that list to a NumPy array of the form (*width, height, channel*) and continue as described in Section 6.1.1. This procedure allowed for the creation of multiple datasets of two and three modality combinations. An overview of one out of the six datasets created is provided in Table 6.2.

**Table 6.2:** An overview of one of the six pre-processed datasets for modality combinations for 2D images. The displayed partitioned dataset is for images of T2, T1ce, FLAIR modalities on the sagittal plane.

|  |  |  |  |
| --- | --- | --- | --- |
| **Grade** | Train | **Set**  Validation | Test |
| G2 | 1*,* 364 | 393 | 103 |
| G3 | 1*,* 661 | 293 | 173 |
| G4 | 2*,* 034 | 609 | 166 |
| Total | 5*,* 059 | 1*,* 295 | 442 |

#### 6.2.3 CLASS IMBALANCE

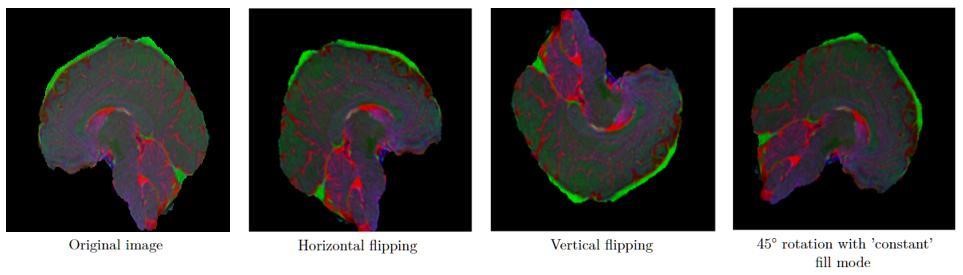
Class imbalance is a common occurrence in machine learning problems that can negatively impact the performance of classifiers. It refers to ”the problem encountered by learning systems on domains for which one class is represented by a large number of examples while the other is represented by only a few”. This can lead to algorithms producing biased results. The provided dataset from TCGA is highly imbalanced in terms of class percentages and therefore, action needed to be taken to address this issue. For all of the fitted models, class weights (i.e. a dictionary that maps class indices to a weight value used for weighting the loss function during training4 were adjusted before training in order to enable the algorithm to give equal importance to all classes. This was an important step since the opposite would mean that the majority class would be favored due to its higher frequency in the dataset. The weight adjustment was implemented provided by the sci-kit learn package. In practice, this means that the minority classes will receive a higher weight while majority classes will get the opposite.

#### 6.2.4 DATA AUGMENTATION

Data augmentation is a popular regularization technique used both for small and large datasets that can increase the generalization capability of a model and reduce the chances of overfitting, as described. In the case of small datasets, such as the ones available in this thesis, it is also used as a way of increasing the number of observations.

In order to investigate how the proposed models would perform using this method, on-the-fly augmentation techniques were applied using the ImageDataGenerator class that generates batches of image data with real-time data augmentation5 provided by Keras. This means that the class would receive images in batches from the training data in the order they are present in the data folder, and each image in a batch would be transformed and later used as the model’s input, according to the following techniques/parameters. For this study, horizontal and vertical flips, 45° rotations and brightness changes (brightness\_range = [0.5, 1.25]) were tested for all trainings, as seen in Figure 4.4. The choice for these techniques was based on experimentation, as

well as papers by Cirillo et al., Nalepa et al. and Kumar et al.



**Figure 6.4:** An example of different augmentation techniques used in this thesis. The original image (left) is horizontally and vertically flipped (2nd, 3rd image from the left and rotated by 45° with ’constant’ fill mode (right), so as to avoid image distortion by stretching. Increased brightness is applied to all augmented images. All above images are synthetic RGB images created by the overlap of multiple modalities.

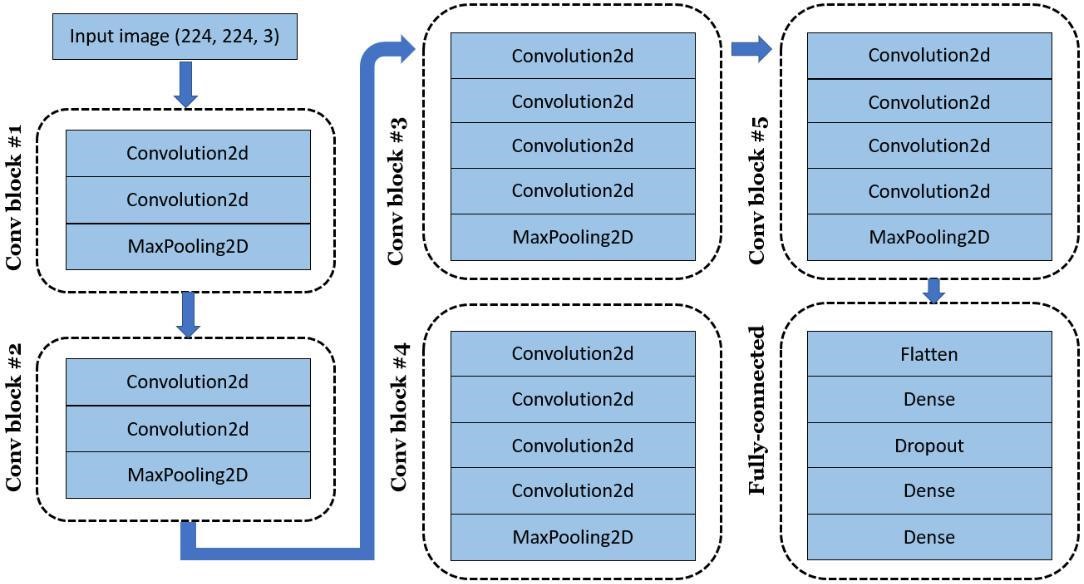
#### 6.2.5 PRE-TRAINED VCG-19

The first pre-trained model considered for this thesis is VGG-19. The choice was based on the fact that it shows very promising results for this specific task, as described in Section 1.3. The pre-trained network is initialized with weights that have been pre-trained on the ImageNet dataset, which contains 1000 image classes. This means that the last layer has 1000 units, and therefore additional top layers need to be added to tailor to the specific needs of each dataset that is fed. As previously mentioned in Chapter 3, VGG-19 accepts input images of size 224 × 224 pixels.

The model was trained using brightness augmentation on the train dataset for 30 epochs, learning rate equal to 1e-5 with the learning rate scheduler of ReduceLROnPlateau and patience of 10 epochs, and batch size 64 for datasets that include single modalities and are taken from one plane each time. The role of the scheduler was to reduce the learning rate if the validation accuracy would not improve after 10 epochs. This choice was based on the size of the dataset and the fact that pretrained models can achieve faster convergence due to the use of pre-trained weights. In the case of datasets containing more than one modality, the model was trained on 35 − 40 epochs (depending on the case), using the same learning rate and scheduler, as well as early stopping with a patience of 15 epochs and the same batch size. For every model, some layers were frozen. The number of layers to be frozen was based on trial and error, since there is no recipe suggesting a standard number that works best.

On top of the pre-trained CNN, a few additional layers were added based on preliminary experiments. More specifically, the first fully connected layer consisted of

50 (or less, depend-ing on the dataset’s performance) nodes, followed by a Dropout layer of rate 0*.*3 (or more, depending on the model) and another fully connected layer of 20 nodes. Finally, a fully con-nected layer of 3 nodes (i.e. equal to the number of classes to be classified) and a softmax activation function for classification was used. The first two fully connected layers used the ReLU activation function. The optimizer of choice was Adam, following the example of pre-vious related work. Lastly, the final model, consisted of approximately **1.5 million** trainable parameters. An example of the model architecture used can be seen in Figure 6.5.



**Figure 6.5:** A diagram of an example VGG-19 model architecture as used for the T1 dataset on the sagittal plane. For convolution block #1 the output after max-pooling is (112*,* 112*,* 64), for block #2 it is (56*,* 56*,* 128), for block #3 the output is (28*,* 28*,* 256), for block #4 it is (28*,* 28*,* 512) and for block #5 (14*,* 14*,* 512). The output of the 6*th* block after Flatten ends in 25*,* 088 dimensions.

#### 6.2.6 PRE-TRAINED MOBILENET

Similarly to the previous model, MobileNet has proven its capabilities in medical image pro-cessing and was, thus, one of the models of choice for this work. Although there are different versions of residual networks, as seen. we chose to use the one with the shallow-est architecture according to our datasets’ needs and the fact that it is already implemented in Keras. It will also be used as a pre-trained model initialized with pre-trained weights on the ImageNet dataset, which means that relatively good performance is expected during the early stages of training. Identically to VGG-19, ResNet50 also accepts input images of size 224 × 224 pixels.

The model was trained using horizontal/vertical flipping, 45° rotation for 30 epochs, learn-ing rate equal to 1e-5 with the ReduceLROnPlateau scheduler and patience of 5 epochs, and batch size 64 for datasets that include single modalities and are taken from one plane every time. For more than one modality in the training set, the model was trained on 30− 45 epochs, using the same learning rate and learning rate scheduler, as well as early stopping with a patience of 15 epochs and the same batch size. For every model, some layers were frozen (100 in our case). The number of layers to be frozen was again, based on experimentation.

In similar fashion, additional layers were added on top of every CNN. In detail, the first fully connected layer consisted of 10 nodes with ReLU as an activation function, followed by a Dropout layer of rate 0*.*3. Finally, a fully connected layer of 3 nodes (i.e. equal to the number of classes to be classified) and a softmax activation function for classification was used. The preferred optimizer was Adam, following the example of previous related work. The final model, consisted of approximately **23 million** trainable parameters. It is important to mention that for each dataset that was trained, different combinations of layers were used, specifically tailored to each case, in order to avoid overfitting.

#### 6.3 MODEL TRAINING

Two pre-trained CNN models (VGG-19, ResNet50) were trained in order to find the optimal classifier for single MR modalities and combinations of them on the TCGA dataset. The decision to use pre-trained models instead of scratch-trained ones was based on their promising results in existing literature and the size of the dataset. Between the two models, VGG-19 was the model that achieved the highest test accuracy. This was initially a surprising finding since VGG-19 has a lower test accuracy on the ImageNet1 dataset. One possible reason to why the first outperformed ResNet50 could be the nature of the data. More specifically, ImageNet contains natural images that are far more different than MR ones, in the sense that the structure of the human brain has a very particular shape. Given that ResNet50 is a much larger and complex model than VGG-19 (the fine-tuned structure consisted of notably 23 million training parameters), one would expect that it performs better on larger training datasets in comparison to the ones used for this work. A potential improvement for this task would be to use pre-trained CNNs on the BraTS2020 dataset instead. Amongst all experiments conducted, VGG-19 was the lightest model of the two, allowed for easy control against overfitting and produced more consistent curves.

Furthermore, it is worth mentioning that the number of frozen layers can play a crucial role in test accuracy. A different amount of frozen layers from the top (or the bottom) could result in a different model being the best. However, since there is no recipe guaranteeing the optimal number of layers to be frozen and there is little time for extensive experimentation, the number that was chosen was equal to 100.

As far as it concerns data augmentation techniques, one can try experimenting with many different ones. Most literature around brain tumors focuses on data augmentation methods for the task of tumor segmentation rather than brain tumor grade classification. There are a few papers comparing various techniques, such as the one from Sajjad et al. In addition to that, one has to take into consideration the fact that not all augmentation techniques are a good fit for MR images since some of them can severely distort an image leading to loss of important information. Therefore, only three augmentation techniques were chosen. More specifically, horizontal and vertical flipping, 45° rotation with ”constant” fill mode and brightness changes were implemented on ResNet50, while on VGG-19 only brightness changes were used. It would be interesting to try even more techniques such as Gamma correction, since the BraTS2020 dataset contains data from 19 different centers or even explore Google’s AutoAugment tool, as mentioned in Zhuge et al. However, each decision should be data-driven and tailored to the specific task at hand.

Lastly, the optimization of the hyperparameters of a CNN is undoubtedly a highly time-consuming and daunting task. Although there is some literature using the same models as in this thesis, very few papers have explored hyperparameter optimization for brain tumor grade classification in 3 grades. Therefore, the choice of hyperparameters during the training phase was based on existing literature, the nature of the task, data size and experimentation. If more time was available, one could try grid search optimization to find the optimal hyperparameters for each model since the potential combinations can be endless.

#### 6.4 CLASSIFICATION

One of the research questions of this thesis regards model performance between VGG19 and ResNet50. In order to address this question, the Wilcoxon Signed-Rank nonparametric test was used as a robust method for comparison between the two classifiers. It is a safe alternative to the paired t-test, in the sense that it does not assume an underlying normal distribution. The logic behind it is simple. It ranks the differences *di* on *i*−th out of *N* datasets between the performance score of two classifiers, ignoring the signs and compares the ranks for positive and negative

differences.

Since our goal is to make a comparison between two models and we wanted to have a rea-sonably large population for the comparisons, AUC values were used to compare performance. If *md* denotes the median difference, then the two-sided hypothesis can be formulated as

*H*0  *md* = 0

)

*H*1  *md* ≠ 0

The test statistic of this statistical test is *T* = *min*(*R*+*, R*−), where *R*+ is the sum of ranks for the datasets on which the second algorithm performed better than the first and *R*− is the sum of ranks for the opposite. Mathematically, (*R*+*, R*−) can be expressed as

  *R*0 *rank di* 1 *rank di , R* 0 *rank di* *rank di* )

1

2

*d*

*i*

0

∑

( ) ( ) + ( )

2

*d*

*i*

0

(

)

+

∑

> = < =

25, most statistical literature includes a table of exact critical values

ForIn *N* ≤ for *T* .

order to implement this test, ten smaller datasets were created from the original one as

described. Each dataset consisted of approximately 6*,* 500 images that were split into training (70%), validation (20%) and test (10%) sets, accordingly. Each dataset was created using the same methods described in Section 6.2.1, making sure that independence was maintained across patient cases within each set. The samples were trained for each classifier and then paired. Next, the differences among AUC scores were used as an input to the wilcoxon function from the scipy package. The parameters passed to the aforementioned function were: zero\_method = 'zsplit' for even split of the zero differences among the sums and alternative = 'two-sided'. The function would then calculate the p-values for the two-sided hypothesis tests. In

practice, a p-value greater than the significance level (e.g.

α = 0*.*05 in this case) is an indication that we fail to reject the null hypothesis *H*0, while the opposite indicates the rejection of *H*0.

#### 6.5 Grad-CAM

In order to add model interpretability, the Grad-CAM technique was used for this project. Grad-CAM is preferred, among other approaches, for its flexibility in regards to model suitability (i.e. it can be applied to a large variety of CNNs), its substantial performance in comparison to other localization methods and ease of implementation. The visualizations it creates can

provide useful insights into the workings of a model as well as help in model

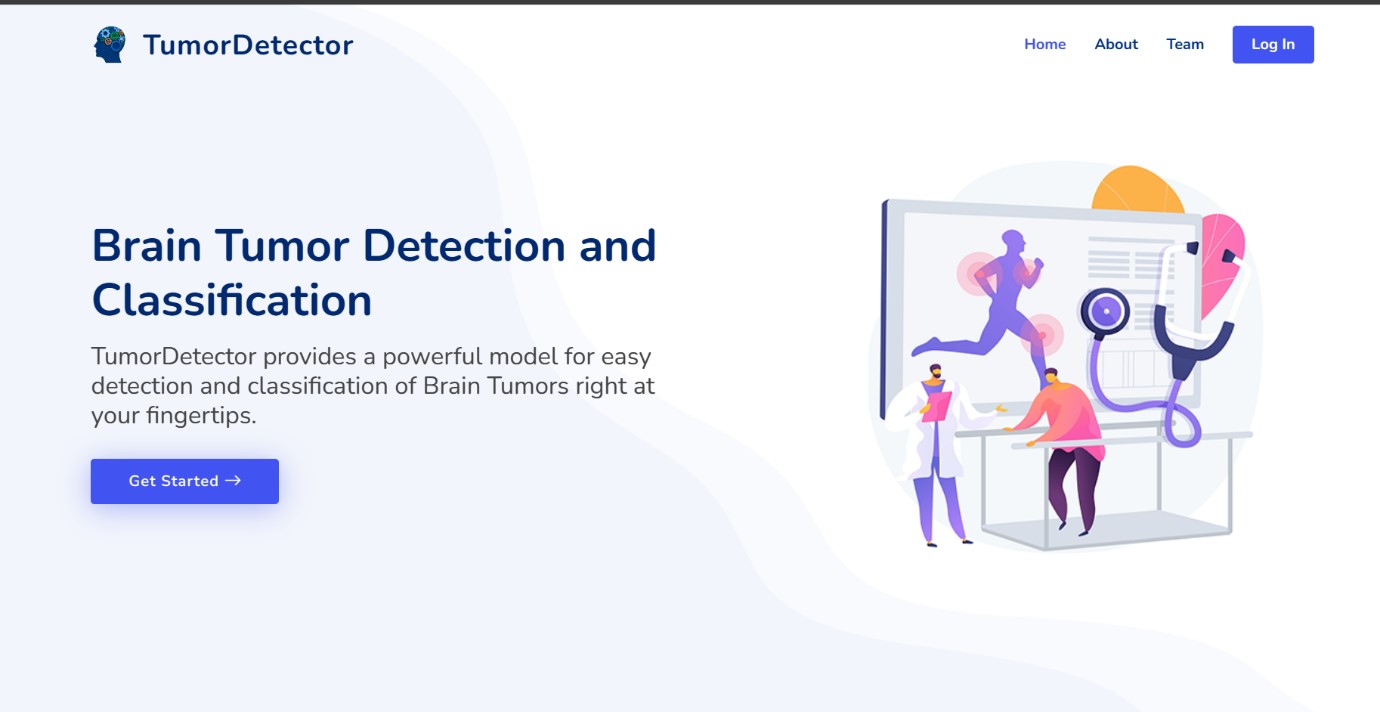
generalization by identifying dataset bias.

For the employment of this technique, several functions were implemented. Firstly, a class named GradCAM was created that included the model to inspect, the index of the class to inspect and the name of the layer to be visualized (if not provided, then the last convolution layer was instructed for use). Then, a function that would inherit from the created class and find the last convolution layer in the saved model was written.

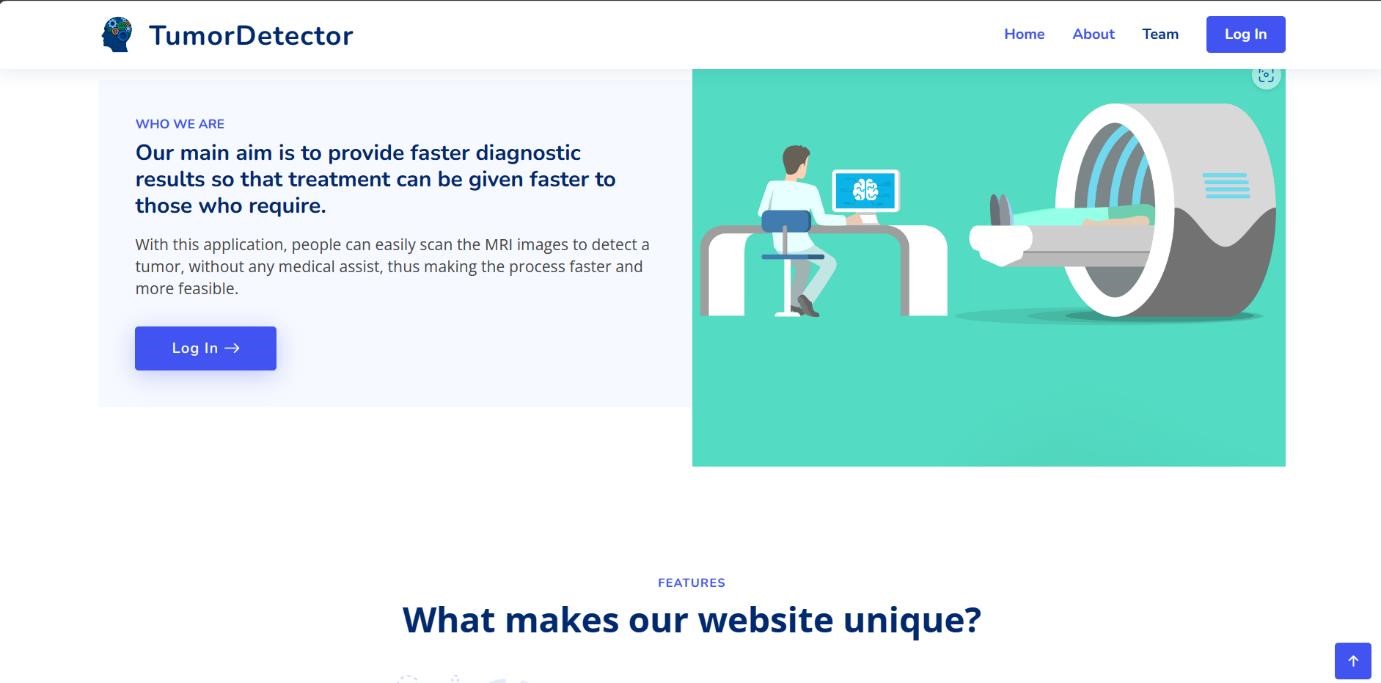
Next, following the paper from a function that would compute the heatmap *MGradc* −*CAM* was designed. In order to achieve that, a gradient model that would acquire the inputs and outputs of the saved model

was implemented. By using the gradient model, we could obtain the gradients mentioned. Afterwards, we would replace the softmax with a linear activation function, store the gradients with Tensorflow’s GradientTape method (by casting the image tensor to a float-32 type, passing the image through the gradient model and grabbing the loss associated with the specific class index) and compute the guided (positive/negative) gradients. More specifically, if the class indices matched the prediction then the gradients would be positive and negative for the opposite. This was done because we wanted to know which gradients pushed down the probability for a class and which ones increased the probability of a class to score high. Once the weight values for each feature map in the convolution layer based on the guided gradient was computed and the heatmap was obtained, the next step was to resize the latter to match the size of the input image.

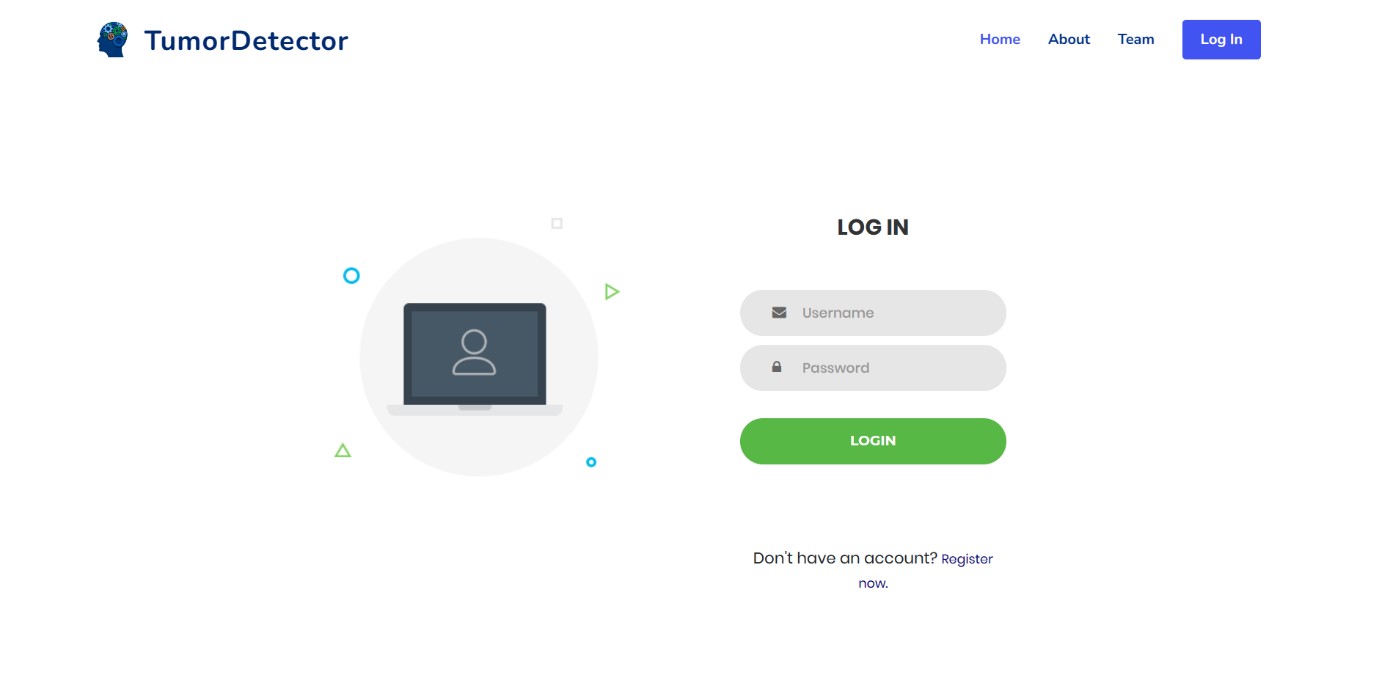
## 7.RESULTS



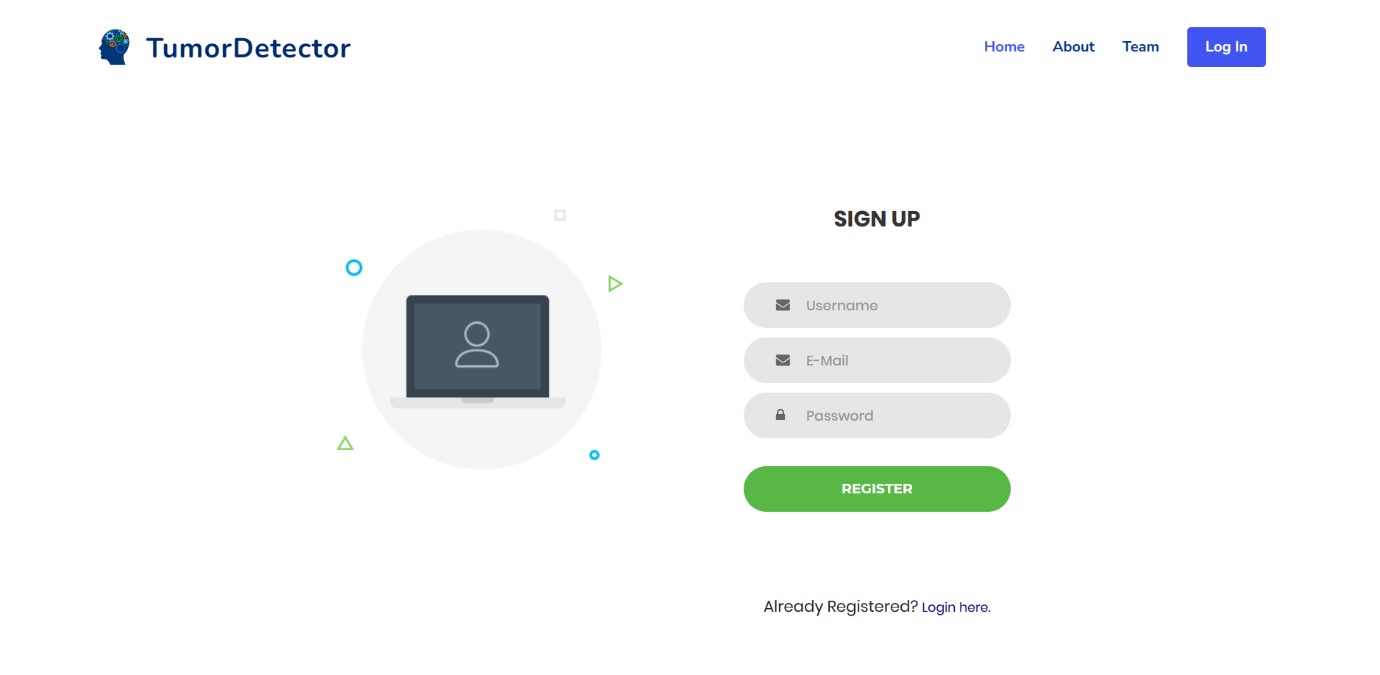
**Fig 7.1:** Home page



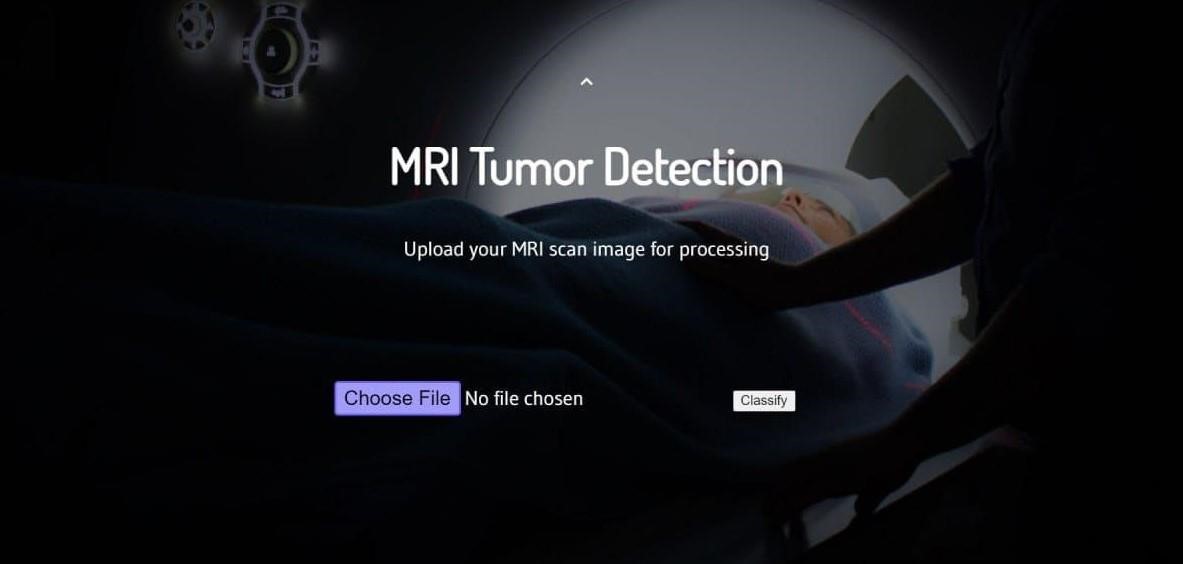
**Fig 7.2:** Log In Tab



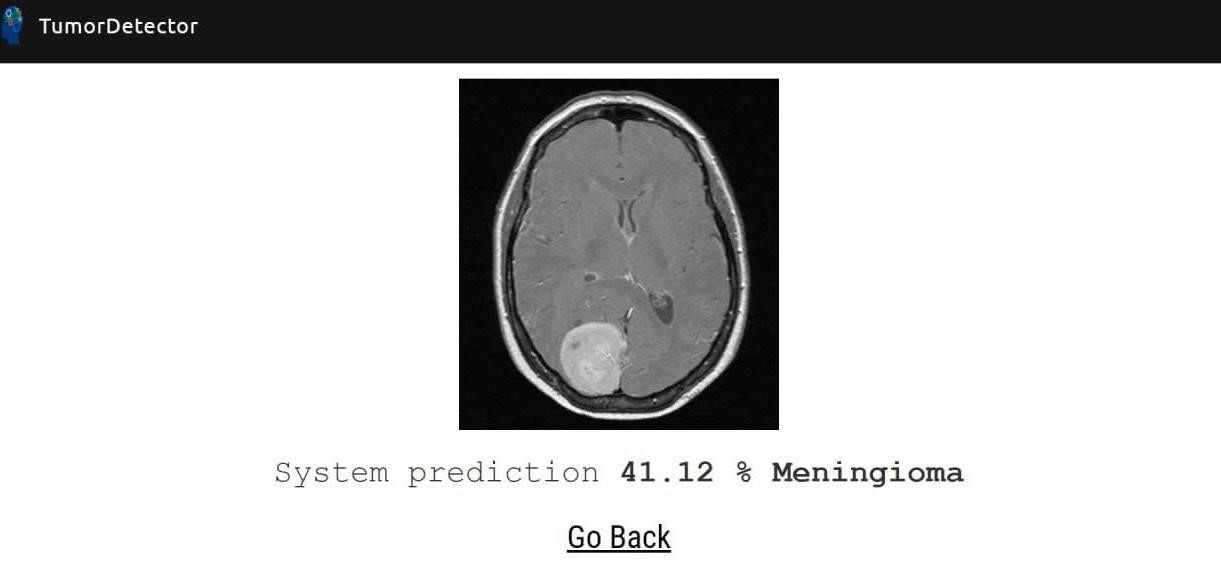
**Fig 7.3:** Log In Page



**Fig 7.4**: Registartion page



**Fig 7.5:** File is choosen as input to find out tumor percentage



**Fig 7.6:** Result of MRI Image

### 7.1 STATISTICAL COMPARISON

In order to answer the first research question, the Wilcoxon Signed-Rank test was performed for three statistical comparisons. First, VGG-19 for single modality images on the sagittal plane was compared against ResNet50, since the sagittal plane was the one that achieved the highest test accuracies for the majority of the experiments. Next, VGG-19 for two modality combinations on the sagittal plane was tested vs. ResNet50 and lastly, the test was conducted against the two models for three modality combinations. The purpose of these tests was to establish if there was a statistically significant difference between the pre-trained VGG-19 model and the pre-trained ResNet50 one with respect to AUC scores. The results of these comparisons are summarized in Table 7.1 where the models compared are mentioned, as well as the p-value and the outcome of the test.

**Table 7.1:** A summary of the statistical hypothesis tests performed using the Wilcoxon Signed-Rank test with a significance level *α* = 0*.*05.

|  |  |  |
| --- | --- | --- |
| Compared Models | p-value | Test result |
| VGG-19 vs. ResNet50 for single modalities | 0*.*129 | Fail to reject *H*0 |
| VGG-19 vs. ResNet50 for combinations of 2 modalities | 0*.*496 | Fail to reject *H*0 |
| VGG-19 vs. ResNet50 for combinations of 3 modalities | 0*.*011 | Reject *H*0 |

As mentioned , the null hypothesis specifies that the median difference is zero. A p-value greater than the significance level (*α* = 0*.*05) signifies failure to reject *H*0. According to the results obtained by the Wilcoxon Signed-Rank test, it was found that there was no statistically significant difference between the two classifiers for single modalities and combinations of two since both tests failed to reject *H*0.

The opposite was however, observed for combinations of three modalities.

## 

**8.CONCLUSION**

1. **Which of the popular CNN models performs best for brain tumor grade classification on the available dataset?**

Among the two models used in this work, VGG-19 obtained the highest test accuracy in the majority of the conducted experiments. When considering AUC scores, the ones for VGG-19 were in most cases higher than the ones for ResNet50. In terms of statis-tical significance, the results were mixed. Our findings using Grad-CAM indicated that ResNet50 was better at localizing the tumor regions.

1. **Which combination of MRI modalities yields the most optimal results for classification?**

For this thesis, six smaller datasets were examined in order to conclude which combination of MRI modalities is optimal for this particular classification task. Based on the results for combinations of two modalities T1ce, FLAIR scored the highest test accuracy and AUC score for VGG-19, and for combinations of three modalities, T2, T1ce and FLAIR had the best performance based on test accuracy and AUC score for the same model. Correspondingly, for ResNet50 on single modalities, T1ce was the highest performing one based on test accuracy and AUC scores. For combinations of 2 modalities, T1, FLAIR was the best one and for combinations of three modalities, T1, T1ce, FLAIR had the highest test accuracy and AUC score.

1. **Is it possible to construct representations of model explainability?**

In this work, model explainability was addressed with the employment of Grad-CAM. The computed heatmaps provided some insight as to where the models focus during classification, although they did not fully explain how or why they focused on specific areas.

### 8.1 APPLICATIONS

* The main aim of the applications is tumor identification.

* The main reason behind the development of this application is to provide proper treatment as soon as possible and protect the human life which is in danger.

* This application is helpful to doctors as well as patient.

* The manual identification is not so fast, more accurate and efficient for user. To overcome those problem this application is design.

* It is user friendly application.

### 8.2 FUTURE SCOPE

**Clinical Validation and Deployment:**

Conducting rigorous clinical validation studies involving radiologists and healthcare professionals to evaluate the performance of the hybrid model in real-world clinical settings is essential. Integration of the proposed system into existing clinical workflows and obtaining regulatory approvals for its use in healthcare practice would be crucial steps towards its deployment and adoption.

**Collaboration with Healthcare Institutions:**

Collaborating with healthcare institutions and research centers to collect large-scale, annotated data sets and establish data-sharing initiatives would enable broader research efforts and facilitate the development of more robust and generalizable models. Open-access repositories for medical imaging data could promote transparency and reproducibility in the field

**9.REFRENCES**

1. Kothari, S., Chiwhane, S., Jain, S., & Baghel, M. (2022, June 1). Cancerous brain tumor detection using hybrid deep learning framework. Indonesian Journal of Electrical Engineering and Computer Science; Advanced Engineering and Science(IAES).https://doi.org/10.11591/ijeecs.v26.i3.pp165 1- 1661

1. Novel Hybrid Boosted Ensemble Learning Framework for Brain Tumor Prediction. (2022, March 23). IEEE Conference Publication IEEE Xplore.<https://ieeexplore.ieee.org/document/9763105>

1. Brain Tumor Detection using Novel Kernel Extreme Learning with Deep Belief Network and

Compare Prediction Accuracy with Fuzzy C-means Clustering. (2022, October 10). IEEE Conference Publication IEEE Xplore.<https://ieeexplore.ieee.org/document/9988190>

1. Singh, M. P., & Shrimali, V. (2022, June 30). Classification of Brain Tumor using Hybrid Deep Learning Approach. Broad Research in Artificial Inteligence Neuroscience; EduSoft publishing house.

<https://doi.org/10.18662/brain/13.2/345>

1. Warjurkar, S. V., & Ridhorkar, S. (2021, January 1). A Study on Brain Tumor and Parkinson’s Disease Diagnosis and Detection using Deep Learning. Atlantis Highlights in Computer Sciences.

<https://doi.org/10.2991/ahis.k.210913.044>

1. Younis, A., Li, Q., Nyatega, C. O., Adamu, M. J., & Kawuwa, H. B. (2022, July 20). Brain Tumor

Analysis Using Deep Learning and VGG-16 Ensembling Learning Approaches. Applied Sciences; Multidisciplinary Digital Publishing Institute.<https://doi.org/10.3390/app12147282>

1. dblp: Ensemble deep learning for brain tumor detection. (2023, September 23). Dblp Computer Science Bibliography.<https://dblp.org/rec/journals/ficn/AlsubaiKASAM22.html>

1. Brain Tumor detection Using Machine Learning and Deep Learning Approaches. (2022, January 28). IEEE Conference Publication | IEEE.

1. Ampavathi, A., & Saradhi, T. V. (2021, January 11). Multi disease-prediction framework using hybrid deep learning: an optimal prediction model. Computer Methods in Biomechanics and Biomedical Engineering; Taylor & Francis. https://doi.org/10.1080/10255842.2020.1869726

1. Raza, A., Ayub, H., Khan, J. A., Ahmad, I., Salama, A. S., Daradkeh, Y. I., Javeed, D., Rehman,

A. U., & Hamam, H. (2022, April 5). A Hybrid Deep Learning-Based Approach for Brain Tumor Classification. Electronics; Multidisciplinary Digital Publishing Institute. https://doi.org/10.3390/electronics11071146