MRI Brain Tumor Detection

Akshil Anilkumar Mundakatil

San Jose State University

ISE 244: AI tools and System engineering

Professor Shilpa Gupta

04 – 14 - 2023

MRI Brain Tumor Detection

Problem Definition

Cancer is characterized by uncontrolled growth and proliferation of abnormal cells. These cells can form tumors that invade nearby tissues and spread to other body parts through the bloodstream or lymphatic system.

Brain tumors are abnormal cell growths within the brain or skull. They can be devastating and significantly impact patients' lives. Early detection of brain tumors is crucial because it is the only way to prolong patients' lives. Brain tumors can cause a wide range of symptoms, depending on their location and size. Common symptoms include headaches, seizures, changes in vision or hearing, personality changes, difficulty speaking or understanding language, weakness or numbness in the limbs, balance problems, and memory issues.

They are classified into two categories Malignant and Benign.

- **Benign Brain Tumors**: Non-cancerous, typically slow-growing tumors with well-defined borders but they can still cause significant symptoms due to pressure on brain structures. Common types include meningiomas and pituitary adenomas. They are usually treatable with surgery.
- **Malignant Brain Tumors**: Cancerous, aggressive tumors that multiply and invade surrounding tissues. They require complex treatment strategies like surgery, radiation, and chemotherapy. Examples include glioblastomas and medulloblastomas, often posing serious risks due to their rapid growth and potential for recurrence.

As discussed in article [5] there are different types of tumors.

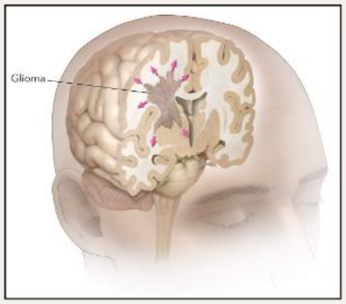
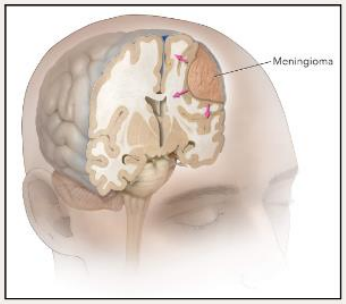
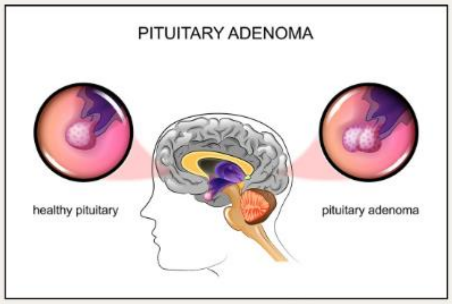
TUMOR TYPES	IMAGES
Glioma tumor: Gliomas are a category of brain tumors arising from glial cells, which support neurons in the brain. They can be benign or malignant, with the latter often being aggressive and fast-growing.	
Meningioma tumor: They originate from the meninges, the protective membranes covering the brain and spinal cord. They are usually benign and slow-growing but can still cause significant neurological symptoms due to pressure on brain structures.	
Pituitary tumor: Also known as pituitary adenomas, form in the pituitary gland, which regulates hormones in the body. Most pituitary tumors are benign, but they can disrupt hormonal balance, leading to conditions like Cushing's disease or acromegaly.	

Table 1. Tumor types

Detecting brain tumors early is important because it gives doctors a better chance to treat them before they cause serious damage. When brain tumors are caught early, they're usually smaller and haven't spread, which makes them easier to remove or manage with treatments like surgery or radiation. This means people have a better shot at recovering and can keep their quality of life intact. If a tumor grows too large, it can press on critical parts of the brain, messing with things like speech, movement, or memory. By catching it early, doctors can help avoid these complications and keep things from getting worse, giving patients a better chance for a healthy recovery.

The manual interpretation of MRI scans for brain tumor detection is getting much more difficult day by day as the number of patients is increasing. The entire process is time-consuming and prone to errors. Misdiagnosis can have dangerous consequences on a patient's life. Therefore, creating a platform for detecting and identifying brain tumors, including their specific types is essential. A model is proposed in the paper [1] that successfully detects whether a tumor is present or not from the given MRI images.

However, the method which the author proposes in paper [1] has a lot of shortcomings that need to be addressed.

- The model was trained on a small dataset, consisting of only 253 images.
- Its function was only restricted to predicting whether a tumor was present or not.
- The images used for training cover only the top section of the brain.
- Consequently, the model is limited to detecting tumors from a single 2D perspective; it cannot detect tumors from the sides or the back of the brain.
- Additionally, there are no plans to create a platform that would allow general users to upload images and check for tumors.

To address these issues, we've developed a more robust model along with a platform where users can upload an image to get information about the presence of a tumor. The new model is also designed to detect the type of tumor, providing more detailed insights.

Project Objectives

The Project objects are highlighted below.

- Find a dataset that includes not only various types of brain tumors but also MRI images taken from multiple angles and perspectives.
- Apply data augmentation techniques to increase dataset quality and diversity.
- Implement Convolutional Neural Network (CNN) models to accurately classify brain tumors from MRI images.
- Evaluate the model's performance using metrics such as accuracy, precision, recall, and F1-score to ensure reliable classification.
- Deploy the model on a user-friendly platform for ease of use in clinical or research

Objective 1: Identify a dataset containing a variety of brain tumors, along with MRI images captured from multiple angles and perspectives.

To accomplish our first objective, we need to find a dataset that offers greater variety than the one discussed in the paper [2]. The dataset [6] used in that paper is divided into two categories: with tumor and without tumor. It comprises a total of 253 images, with an approximate 60:40 split between the two categories.

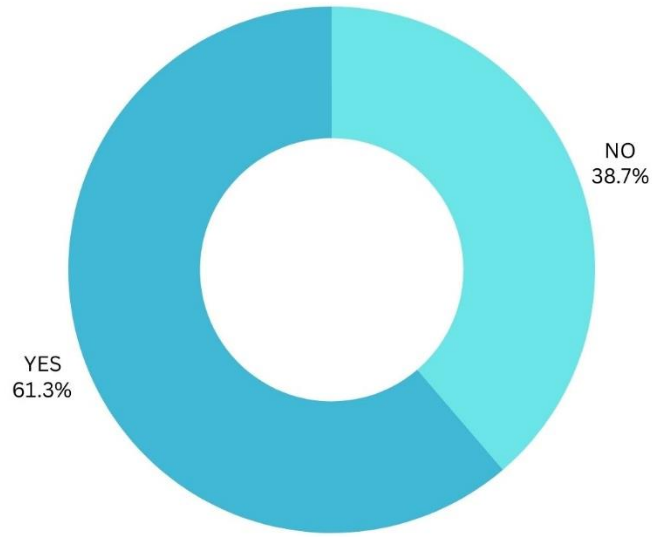


Fig 1. The division between tumor and non-tumor images in Dataset 1

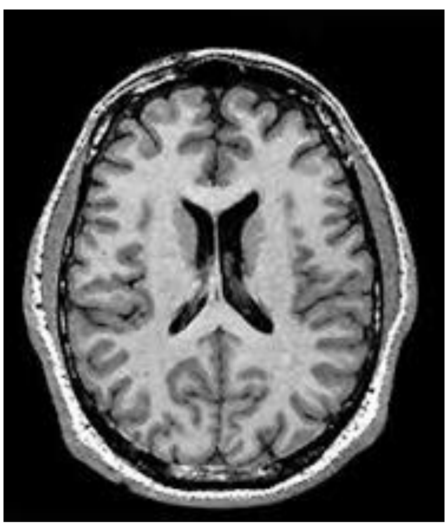


Fig 2: Image without tumor

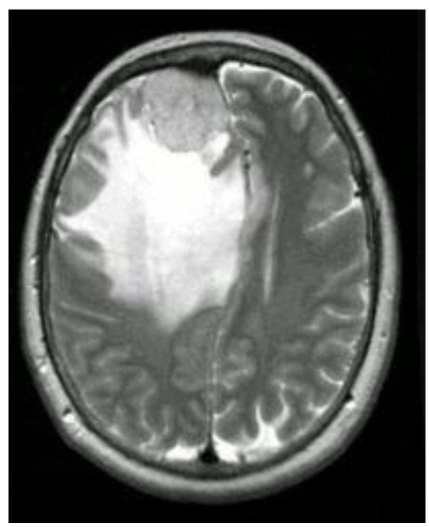


Fig 3. Image with tumor

In the new dataset [7], there are four categories of images. The dataset contains three types of brain tumors: glioma, meningioma, and pituitary, along with a fourth category comprising images with no tumor.

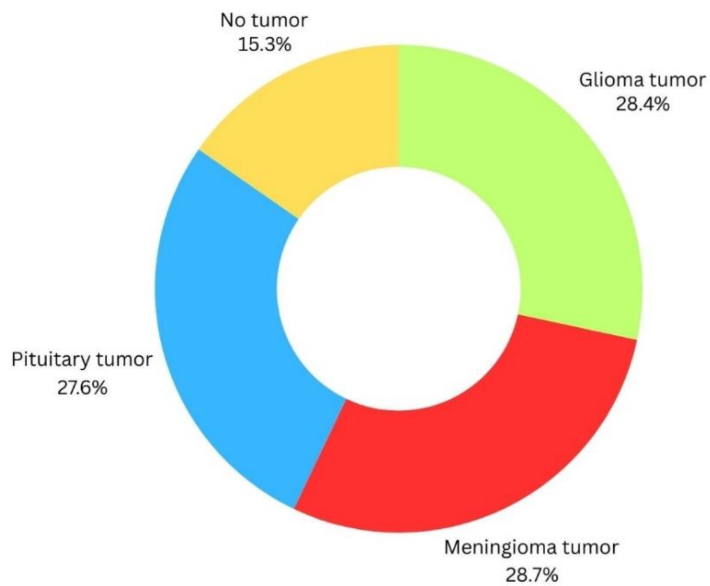


Fig 4. Split between different kinds of tumors.

Additionally, the images in this dataset have been taken from different angles, allowing for a 3D detection perspective in the model. This dataset includes 3,264 images: 926 belong to glioma, 937 to meningioma, 901 to pituitary, and 500 are non-tumor images. Examples are shown in Fig [5].

Objective 2: Apply data augmentation techniques to increase dataset quality and diversity

Data augmentation in this model is used to artificially increase the dataset size, improve model generalization, simulate real-world variations, reduce overfitting, enhance model robustness, and balance class distribution.

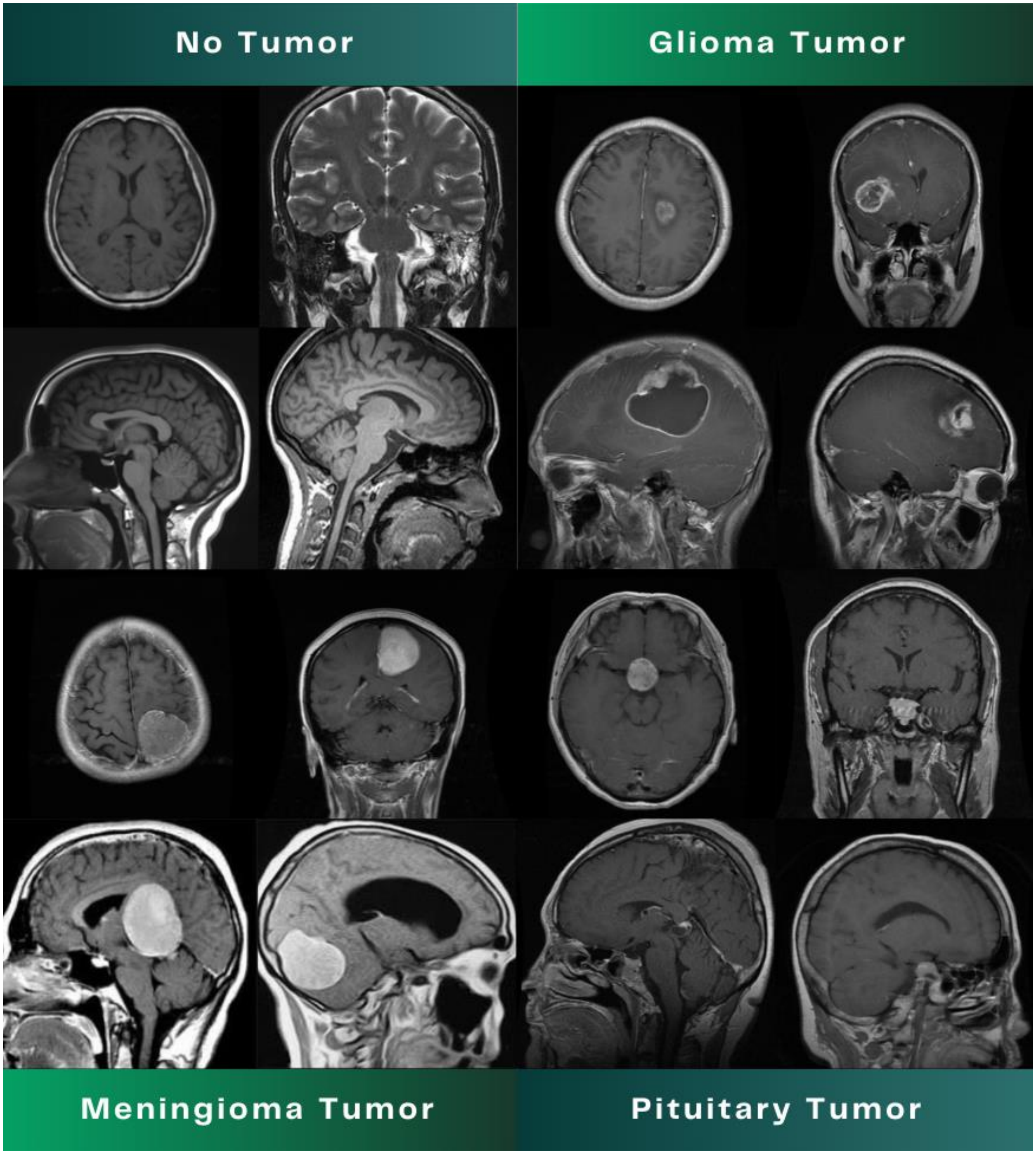


Fig 5. Different types of tumors and their images are used in the dataset.

Objective 3: Implement Convolutional Neural Network (CNN) models to accurately classify brain tumors from MRI images.

Design and implement a CNN model that accurately classifies brain tumors using MRI images. We will use a method called Transfer learning as it greatly increases the accuracy and reduces training time.

Objective 4: Evaluate the model's performance using metrics such as accuracy, precision, recall, and F1-score to ensure reliable classification.

Since we're using transfer learning, we'll evaluate various NN models and select the one that performs best. The criteria for evaluating the models will include F1 scores, accuracy, precision, and recall.

Objective 5: Deploy the model on a user-friendly platform for ease of use in clinical or research

Once we've selected the best-performing model, we'll deploy it to a user-friendly platform, allowing users to easily upload their MRI scans and receive results.

Analysis

To build a new and improved model, we start by acquiring a new dataset that includes brain images from various angles and encompasses different tumor types. Additionally, we need a dataset with a larger volume of images to facilitate more effective model training. Dataset [2] suits the bill as it not only has a lot of images but also has images for tumors of different types. Below are the features of the said dataset.

Dataset

- Comprises images from various angles.
- Categorizes images into four tumor categories.

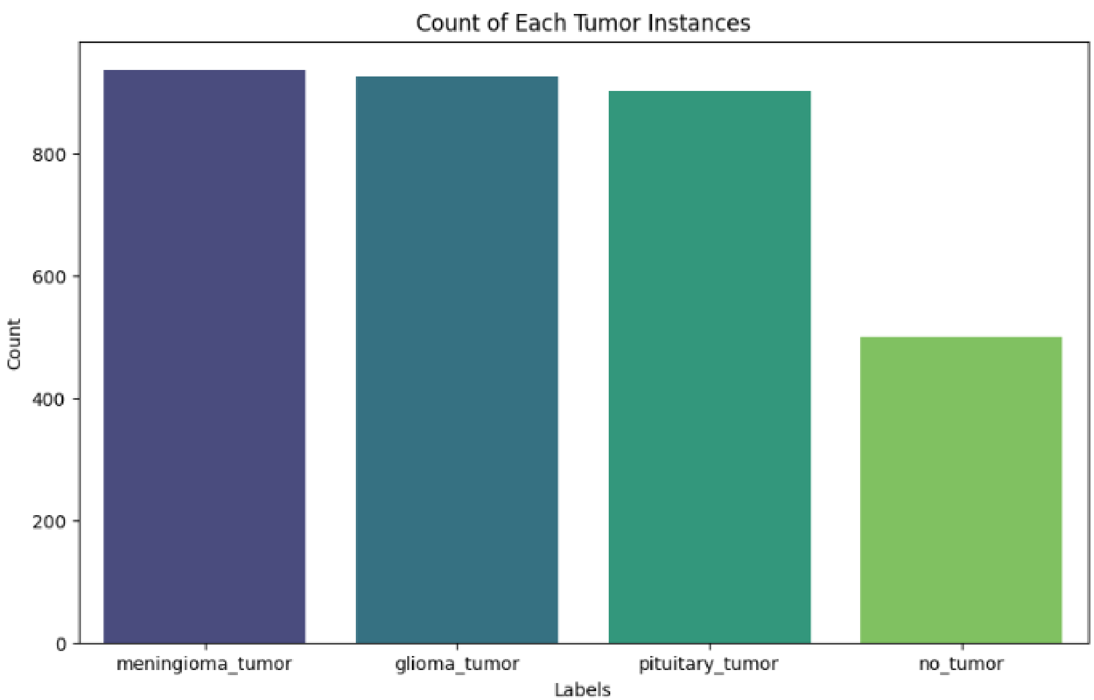


Fig 6. Dataset distribution

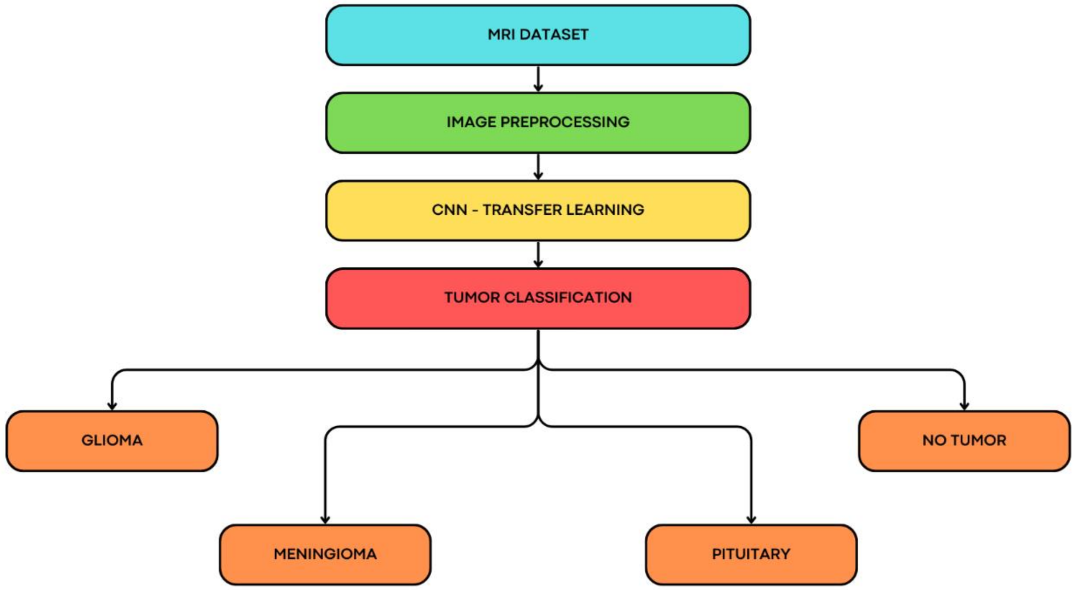


Fig 7. Model Architecture

Image Preprocessing: This is the second step of our project. Image preprocessing is essential as it helps our model to have a uniform and compatible set of images

Standardization: This involves resizing images to a uniform size, we set it to 150x150 pixels so that it works with the model that we have designed.

Data Augmentation: Data Augmentation is applied to increase the diversity of the training data and improve model generalization. The techniques used by our model are

- Rotation: Images are rotated by a certain degree to simulate variations in viewing angles, for this model we set it to 15 degrees.
- Horizontal Flip: mages are flipped horizontally to account for potential left-right orientation differences.

- Fill Mode: When performing transformations that create new pixels (e.g., rotation), a fill mode is used to determine how to fill in these new pixels. We use the fill mode nearest for this model.

We then perform shuffling to prevent the model from learning any sequence or order in the data. When shuffling is applied the dataset is shuffled randomly.

As we do for all machine learning models. We divide the dataset into 3 parts, train, test, and validation.

Train – Test – Validation split: We first divide the data into train and test sets, maintaining a split ratio of 9:1. Then, from the training dataset, we further split it into two parts: train and validation datasets, with a split ratio of 9:1. We also apply one hot encoding to the Y labels as they are categorical in kind.

CNN – Transfer Learning

For this model, we will be using Transfer learning.

- Transfer learning is employed to leverage pre-trained CNN models, that have been trained on large-scale image datasets.
- By initializing the model with pre-trained weights, the CNN can quickly learn relevant features from brain MRI images, even with limited labeled data.
- Transfer learning allows one to achieve higher classification accuracy, faster convergence, and better generalization compared to training CNNs from scratch.

- The pre-trained model's ability to capture generic image features, combined with fine-tuning on the brain tumor dataset, enables the CNN to learn discriminative features specific to different tumor types, improving diagnostic accuracy.

Model Architecture

We have used different neural network architectures. These include DenseNet121, MobileNetV2, and ResNet101V2 Out of all, the best-performing model was EfficientNetV2B1. EfficientNetV2B1 belongs to the EfficientNet family of convolutional neural networks. EfficientNetV2B1 is a variant of EfficientNet optimized for efficiency and performance in image classification tasks.

EfficientNet Overview:

- EfficientNet is characterized by its unique scaling method, which balances model depth, width, and resolution to achieve better performance with fewer parameters.
- The architecture consists of multiple blocks, each comprising convolutional layers, batch normalization, activation functions, and skip connections.
- EfficientNet achieves superior performance by efficiently scaling the depth, width, and resolution of the network based on a compound coefficient, allowing it to adapt to different resource constraints.

EfficientNetV2B1:

- EfficientNetV2B1 is a specific variant of EfficientNet that incorporates improvements in model design and training techniques.
- It introduces enhancements such as Swish activation functions, drop connect regularization and stochastic depth regularization.
- EfficientNetV2B1 retains the efficiency and effectiveness of the original EfficientNet while further improving performance and robustness.

Model Layers:

- Input Layer: The input layer accepts images with a predefined size of 150x150 pixels with three color channels (RGB).
- Convolutional Layers: These layers consist of convolutional filters that extract features from input images. EfficientNetV2B1 employs a series of convolutional layers with varying filter sizes and depths to capture hierarchical features.
- Pooling Layers: Pooling layers reduce the spatial dimensions of feature maps, helping to extract key features while reducing computational complexity.
- Global Average Pooling Layer: This layer aggregates spatial information across all feature maps by computing the average value of each feature map. It helps in reducing the spatial dimensions to a fixed size, enabling the model to handle inputs of variable size.
- Dropout Layer: Dropout is a regularization technique that randomly drops a fraction of neurons during training to prevent overfitting. We set it to 0.5

- Dense (Fully Connected) Layer: The dense layer consists of neurons connected to all neurons in the previous layer. It performs classification by mapping extracted features to output classes.
- Output Layer: The output layer uses a SoftMax activation function to produce class probabilities for each tumor type, enabling multi-class classification.

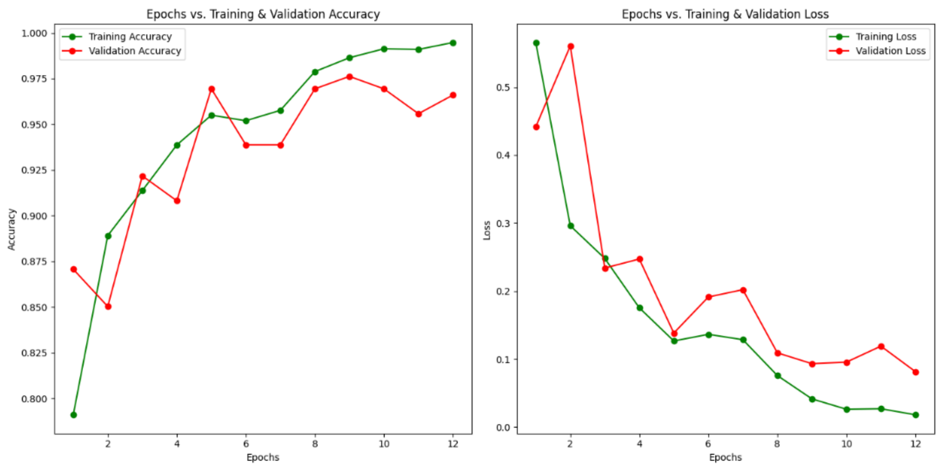
Training and Evaluation:

- The model is trained using the Adam optimizer with a categorical cross-entropy loss function.
- Training is performed over multiple epochs, with early stopping based on validation loss to prevent overfitting. We have used 12 epochs to train the model
- A separate test set evaluates Model performance using accuracy and other relevant metrics.
- Confusion matrices and classification reports are generated to assess the model's performance across different tumor types.

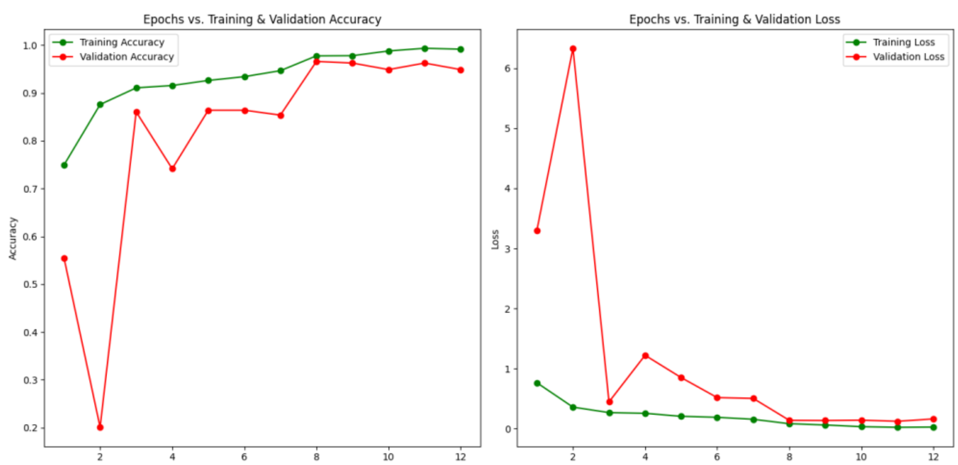
Results

As discussed earlier, we have employed various neural networks and evaluated their performance using different evaluation parameters to determine the best model among them. The evaluation parameters include accuracy, F1 score, precision, and recall. Initially, we examine the training versus validation accuracy graph and the training versus validation loss graph for all the models.

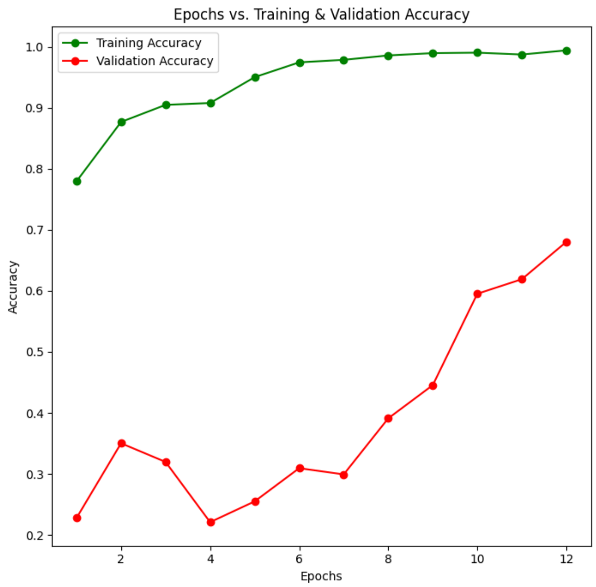
1) EfficientNetV2B1

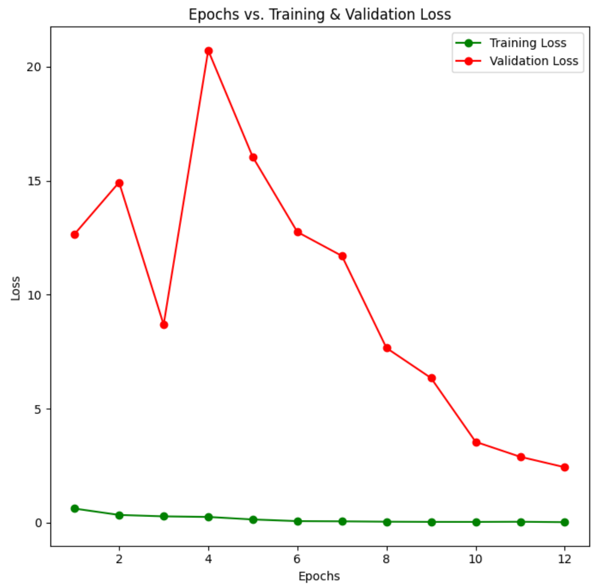


2) DenseNet121

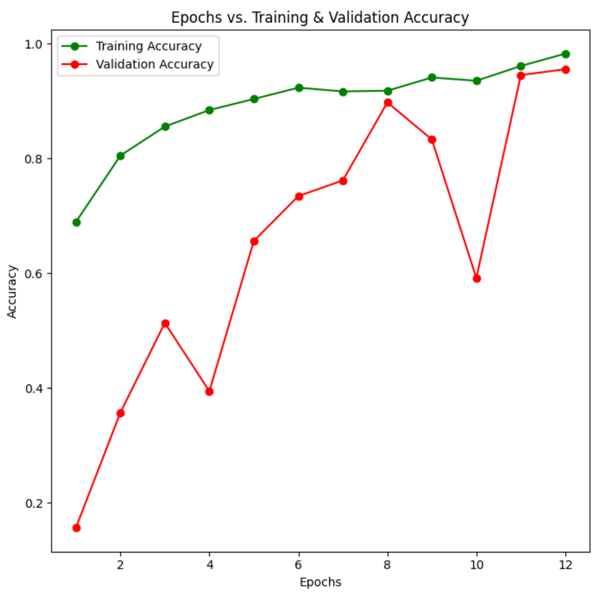


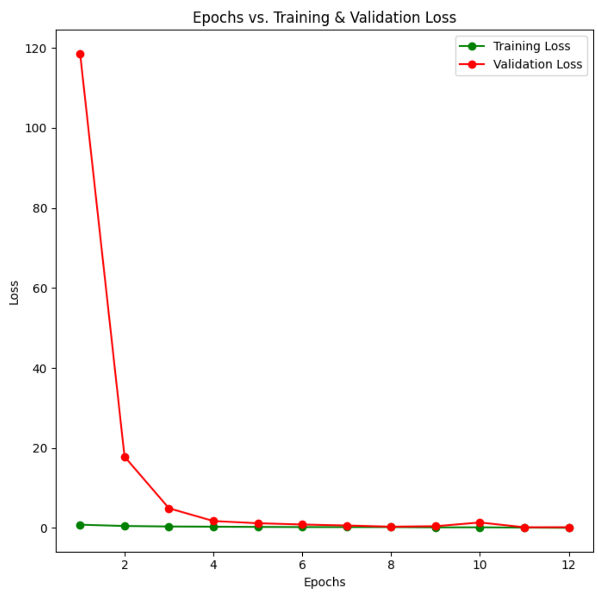
3) MobileNetV2



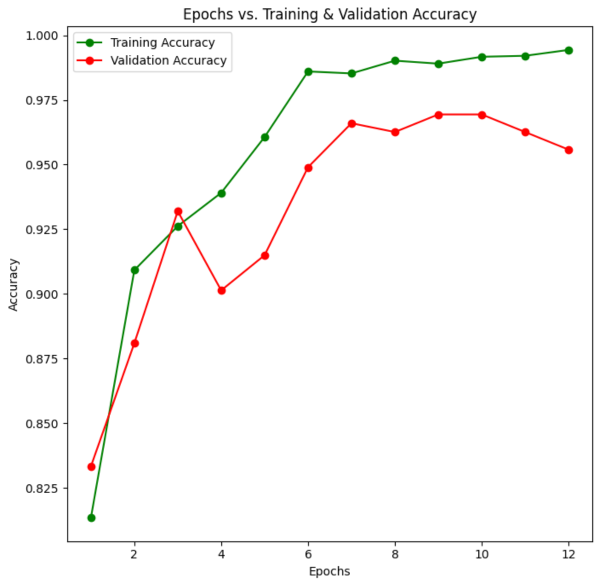


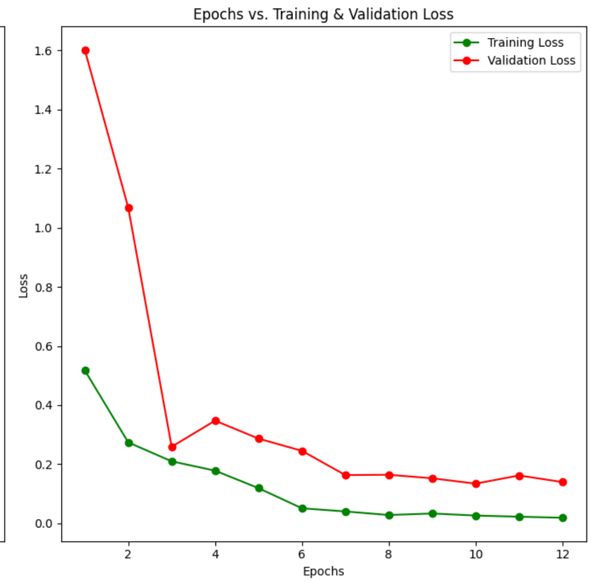
4) ResNet101V2



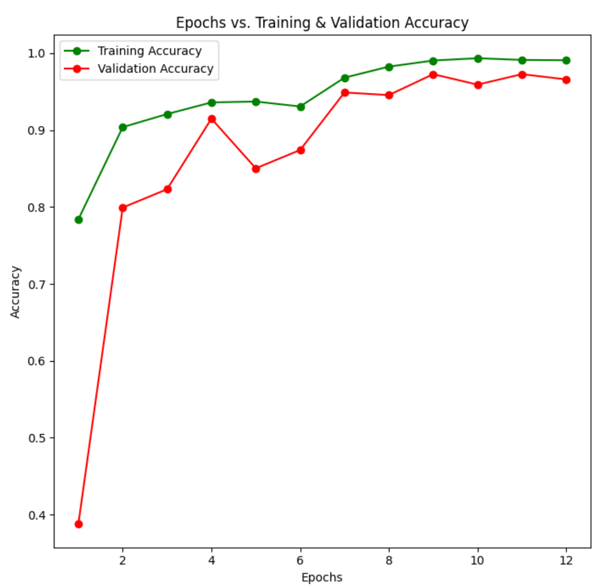


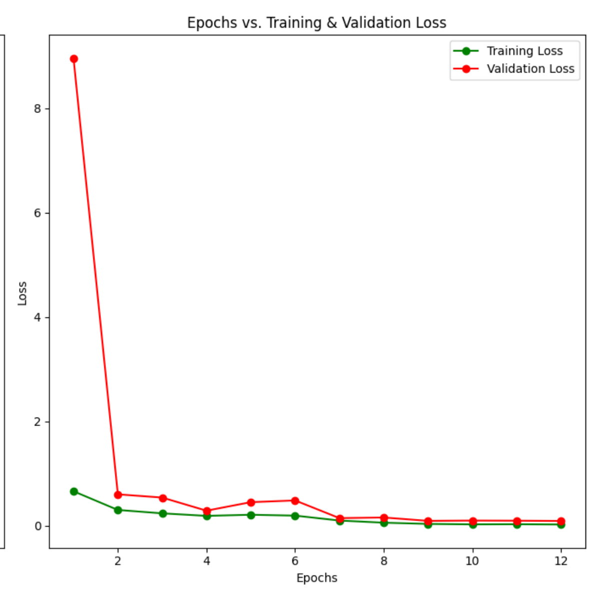
5) Xception





6) InceptionV3





- EfficientNet shows a steady improvement in both training and validation accuracy it makes a slow and steady ascent, while losses gradually decrease. This smooth progress is because the model knows how to keep things balanced.
- DenseNet's training accuracy shoots up quickly, thanks to its clever way of connecting features, which helps it learn better from the data. Validation accuracy follows a similar path.
- On the other hand, MobileNet doesn't perform as well, you can tell from just looking at the images.
- For ResNet, both training and validation curves show a steady increase, although not as smooth. Losses drop quickly for both sets.
- With Xception, accuracy steadily climbs during training, and the same goes for validation. It's good at minimizing loss during training, and that is reflected in both training and validation sets.
- InceptionV3 sees a big jump in accuracy early on during training, and validation catches up quickly. Losses decrease rapidly at first, but it might slow down later on.

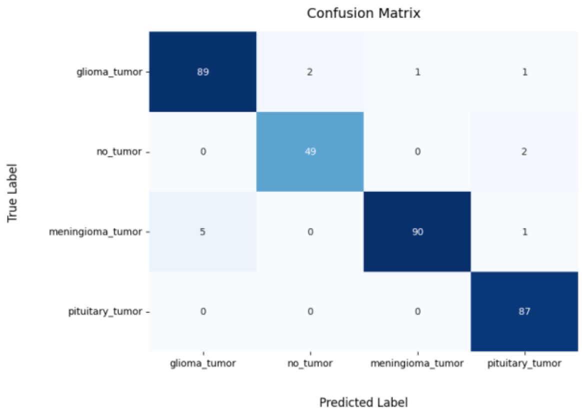


Fig 10. EfficientNet Confusion Matrix

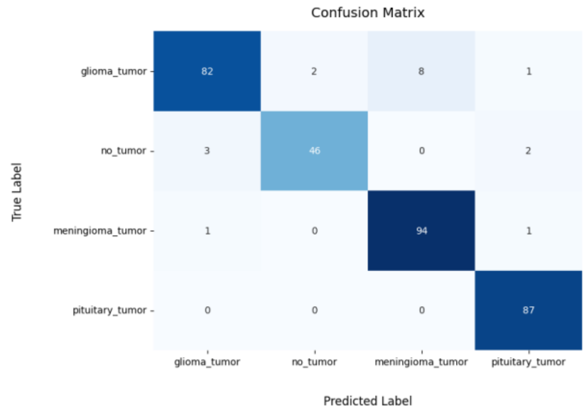


Fig 12. ResNet Confusion Matrix

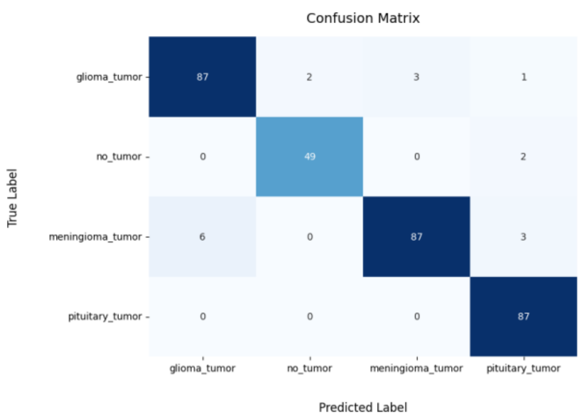


Fig 11. DenseNet Confusion Matrix

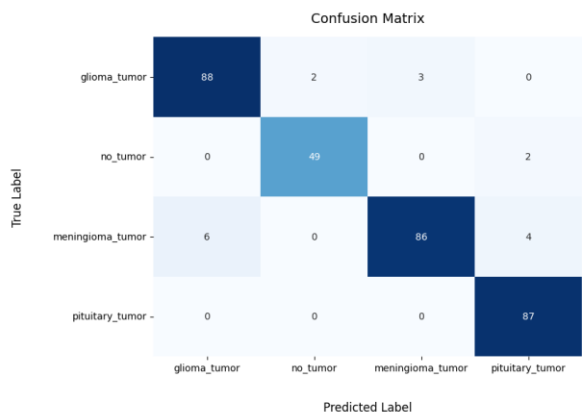


Fig 13. Xception Confusion Matrix

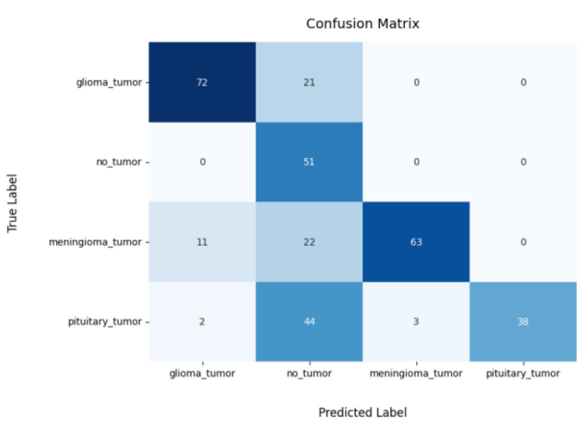


Fig 12. MobileNet Confusion Matrix

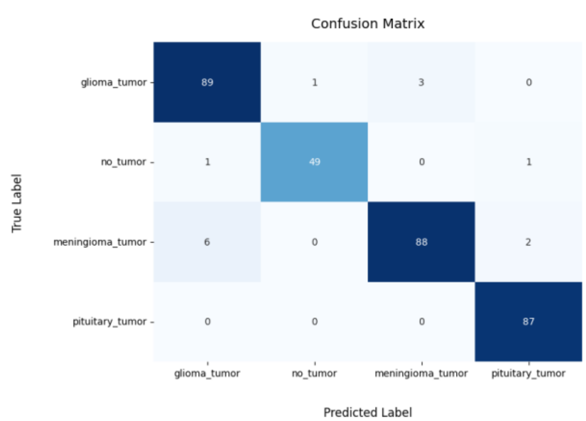


Fig 15. Inception Confusion Matrix

By carefully evaluating the confusion matrix and the evaluation parameters in Table [2], we come to know that EfficientNet and Inception perform the best compared to the rest. Keeping that in mind, we proceed with EfficientNet as our primary model for deployment. We will use this model to create a platform where users can easily upload images and get results.

Here are a few example images tested on the model, along with the predicted label and the associated confidence parameter.

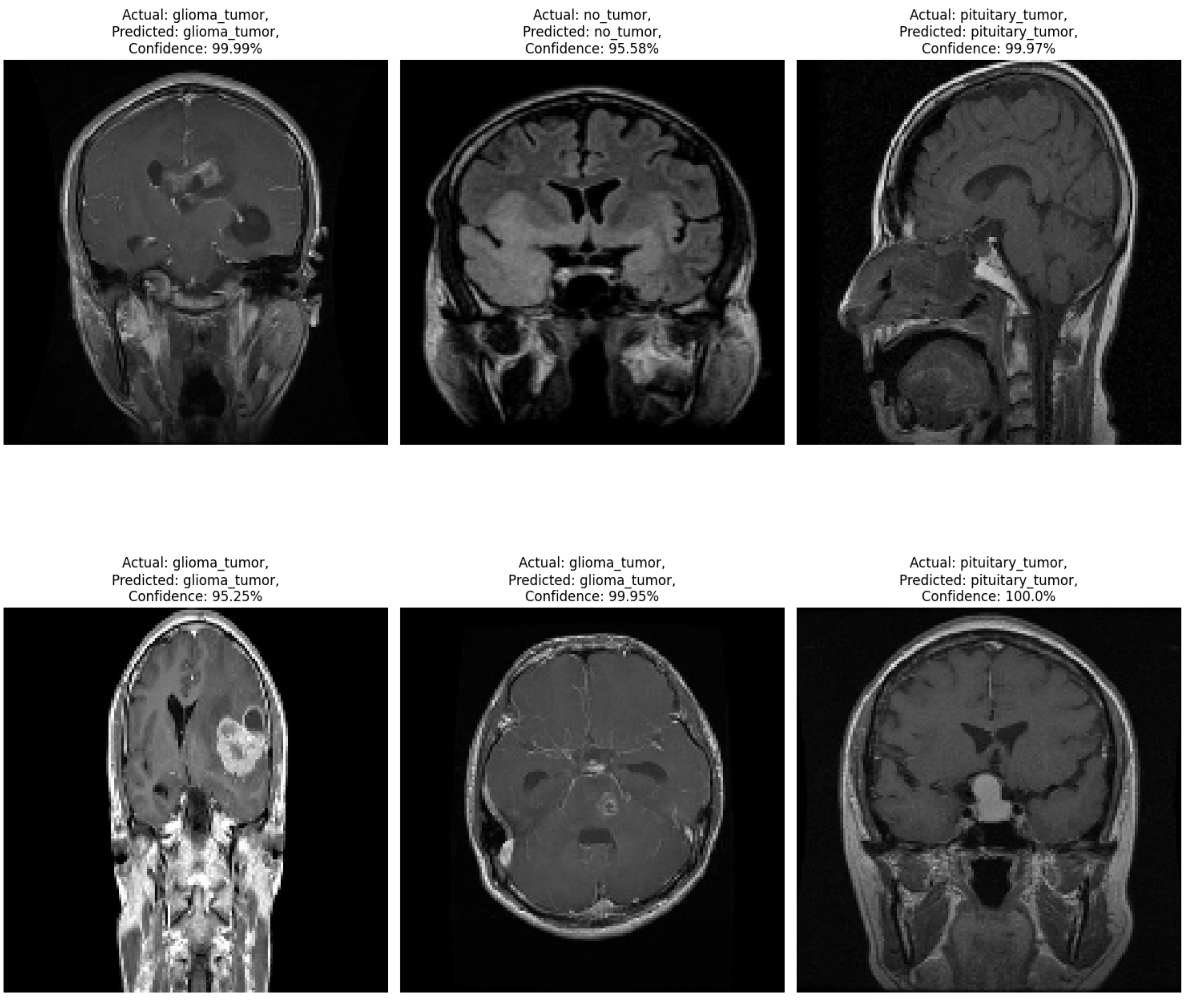


Fig 10. Example images with predicted values versus actual values, along with their confidence scores.

The proposed EfficientNet model achieves excellent performance in brain tumor classification, with an accuracy of 96% on the test set. It demonstrates robustness in distinguishing between different tumor types, as evidenced by the confusion matrix and classification report.

Discussion

To determine which model performs the best, we compare all the evaluation parameters discussed earlier for each model used. Based on these results, we decide which model to deploy on a platform. To achieve this, we first examine Table [2] and understand the results.

	ACCURACY	PRECISION	F1	RECALL
EFFICIENTNET	96%	96%	96%	96%
DENSENET	95%	95%	95%	95%
RESNET	94%	95%	94%	94%
MOBILENET	69%	84%	70%	69%
XCEPTION	95%	95%	95%	95%
INCEPTION	96%	96%	96%	96%

Table 2. Parameter comparisons.

Table 2 provides insights into every model's performance.

- EfficientNet: Best performing model with 96% accuracy, F1 score, recall, and precision.
- Inception: Best performing model with 96% accuracy, F1 score, recall, and precision.
- DenseNet: Model with 95% accuracy, F1 score, recall, and precision. A good and stable architecture that we can use.
- Xception: Similar to DenseNet, a model with 95% accuracy, F1 score, recall, and precision.
- ResNet: Shows a slight drop in accuracy, F1 score, and recall (94%) compared to Xception and DenseNet, but precision remains the same.
- MobileNet: A model that doesn't perform well in this case, with 69% accuracy, 84% F1 score, 70% recall, and 69% precision.

In the current proposal, we have applied preprocessing, data augmentation, and transfer learning to build a model that can accurately predict the presence of a tumor and classify it into its respective category, if it exists. Our model results conclude that Inception and EfficientNet proved to be the best-performing algorithms in this case. This decision was evaluated based on the parameters we set, including F1 score, accuracy, precision, and recall.

EfficientNetV2

The model that is suitable for deployment seems to be EfficientNetV2 which is validated through parameter comparison.

EfficientNetV2: EfficientNetV2 is an advancement over the original EfficientNet architecture, which aims to achieve higher accuracy while maintaining efficiency in terms of computational resources [8]. The architectural breakdown is as follows.

The architecture for EfficientNetV2 is shown in the Table [3]. This is taken from the paper [8] "EfficientNetV2: Smaller Models and Faster Training".

Stage	Operator	Stride	#Channels	#Layers
0	Conv3x3	2	24	1
1	Fused-MBConv1, k3x3	1	24	2
2	Fused-MBConv4, k3x3	2	48	4
3	Fused-MBConv4, k3x3	2	64	4
4	MBConv4, k3x3, SE0.25	2	128	6
5	MBConv6, k3x3, SE0.25	1	160	9
6	MBConv6, k3x3, SE0.25	2	256	15
7	Conv1x1 & Pooling & FC	-	1280	1

Table [3] EfficientNetV2 architecture

- Early Layer Blocks: EfficientNetV2-S uses both MBConv and Fused-MBConv blocks in the initial layers for efficient feature extraction.
- Smaller Expansion Ratios: It opts for smaller expansion ratios in MBConv blocks to reduce memory overhead.
- Kernel Sizes: Prefers smaller 3x3 kernel sizes but compensates by adding more layers to capture complex features.
- Last Stride-1 Stage: Removes the last stride-1 stage from the original EfficientNet due to its large parameter size and memory overhead.

Architecture Breakdown:

- Stage 0: Starts with a 3x3 convolutional layer with a stride of 2, generating 24 channels.
- Stages 1-6: Uses a mix of Fused-MBConv and MBConv blocks, increasing channel count and possibly reducing spatial dimensions.
- Stage 7: Combination of operations including 1x1 convolution, pooling, and fully connected layers, possibly for feature extraction and classification.

After selecting the model, we create a keras file for the EfficientNet model. We develop a FastAPI application to access the pre-trained TensorFlow model. The next step is to verify if the model produces accurate results. For this purpose, we access the localhost domain using Postman and upload an image to check if the model predicts successfully. In the project scenario, the model successfully predicted values for uploaded images.

Following this, we create a front-end website to facilitate easy image uploads for users and to display results instantaneously. After linking the model with the front end, we conduct several rounds of testing to ensure that it predicts and displays results accurately.

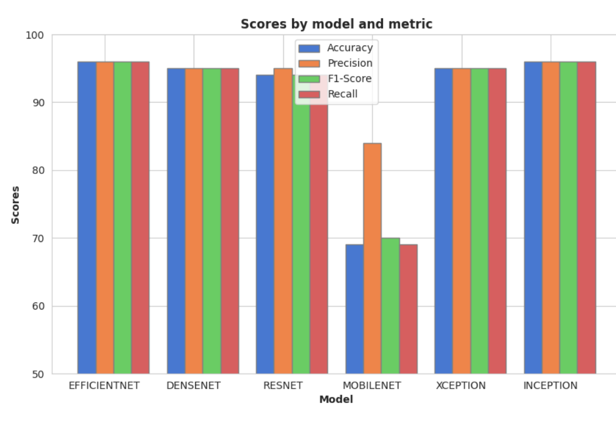


Fig 11. Scores vs Model.

Evaluation and Reflection

For the proposed project, various neural network architectures were evaluated, and their results were compared. The models that performed the best were EfficientNetV2 and Inception. We selected EfficientNetV2 as the model to be deployed so that it can be easily used by the general public.

Fig [12] and Fig [13] represent a basic UML diagram and a block diagram for this scenario respectively.

UML Diagram: The UML diagram provides a simple representation of the scenario. There are two actors: the doctor or patient and the model. The doctor uploads the image, and the model predicts and returns the result.

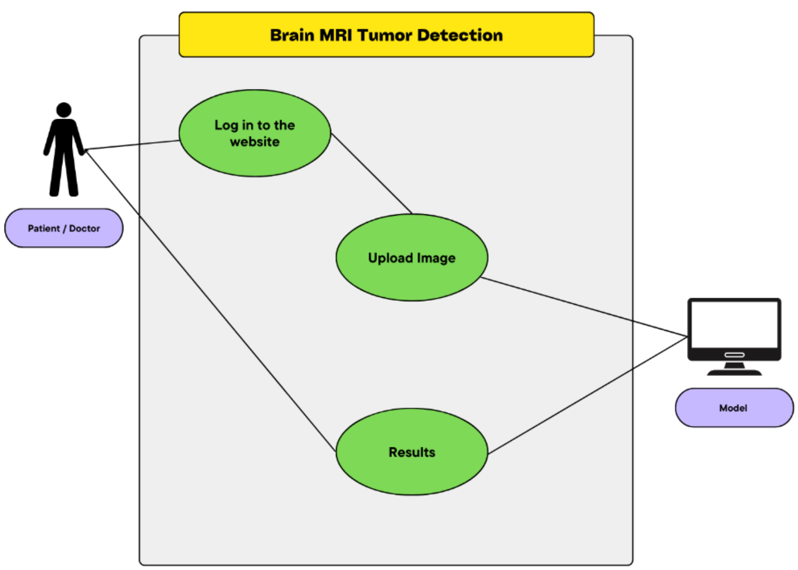


Fig 11. UML diagram

Block Diagram: The block diagram illustrates the inner workings of the model. It includes different components like the user interface (front end), backend, and admin, which come together to create a robust platform for brain tumor detection.

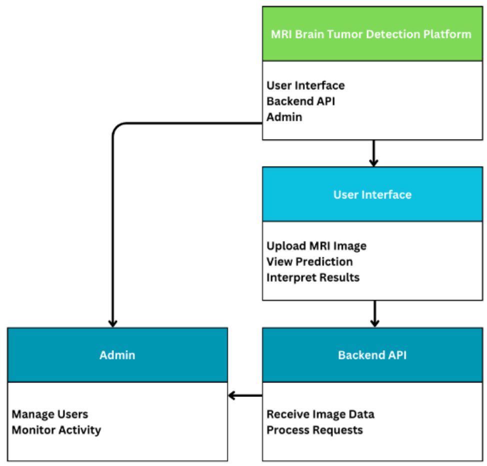


Fig 12. Block diagram

The website will function as described in the UML diagram. Fig 13 – 14 shows how the website will function.

At the main home page, simply upload the image that you want to be predicted, and the model will provide an output based on its prediction. The output will include the tumor type along with the confidence percentage.

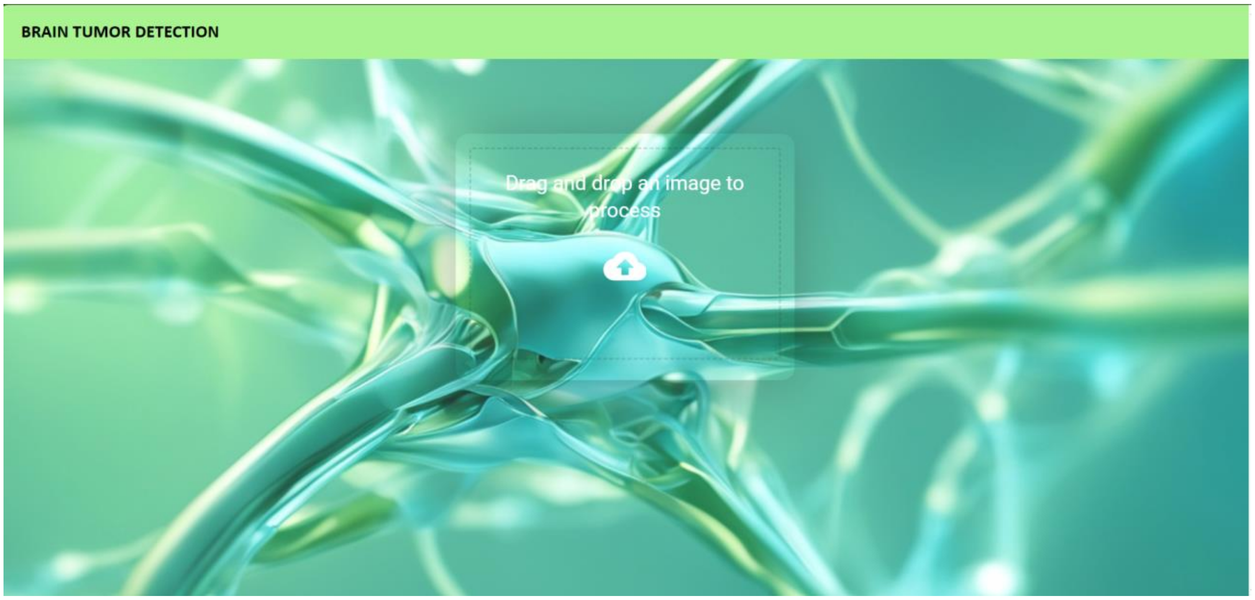


Fig 13. The main page of the website

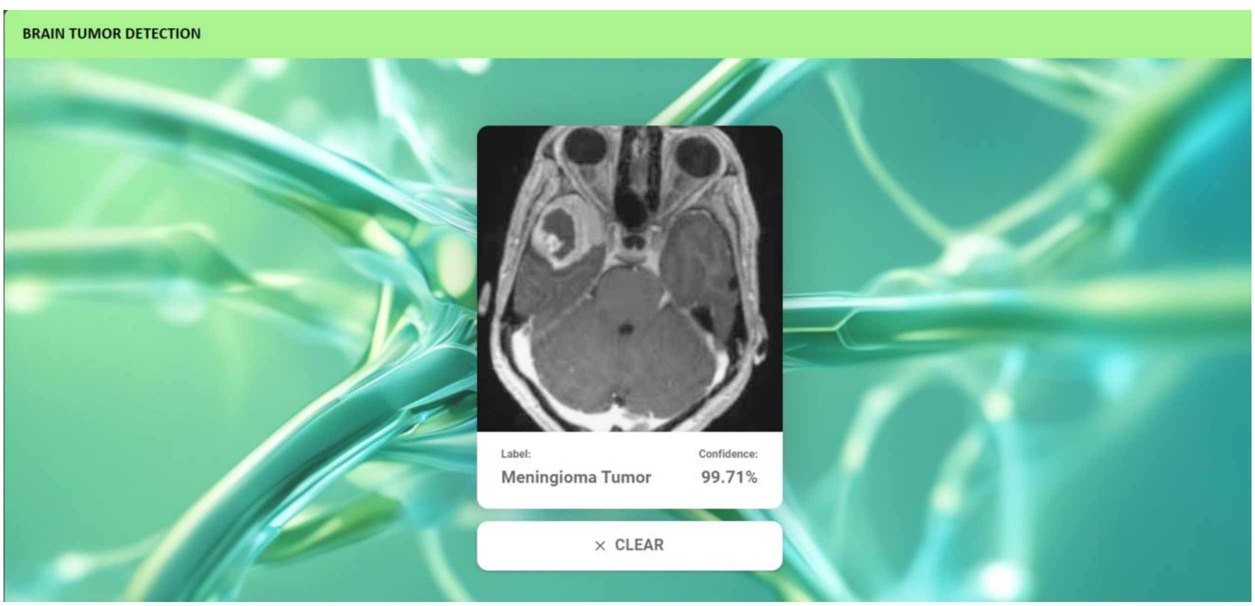


Fig 14. Upload the image and obtain results.

Certain assumptions were put forth during the project's life cycle. These involved

1) The previous model's [1] inability to detect 3D MRI scans.
2) Lack of data [1].
3) Not being able to detect the tumor type
4) Model's Accuracy on base CNN.
5) The platform lacks [1] sufficient user-friendliness for regular users to upload images and obtain results.

These assumptions were tackled with the following solutions.

1) Using a new dataset [7] which had images from all sides of the brain.
2) The new dataset contains twelve times more images.
3) New dataset having data for 3 kinds of tumor.
4) Using transfer learning as it is advantageous and provides predictions with higher accuracy. Generally, pre-trained models effectively learn discriminative features from limited labeled data, accelerating the model convergence and enhancing the model's generalization capabilities
5) Creating a web platform where users can upload images and get results.

Impact of the Project

The solution presented in this project holds significant implications for clinical practice and research in the field of neuroimaging. By automating the process of brain tumor detection and classification, our model can assist radiologists and clinicians in making timely and accurate diagnoses, leading to improved patient outcomes and treatment planning. Moreover, the deployment of the model on a user-friendly platform enables seamless integration into existing healthcare systems, empowering healthcare providers with advanced AI tools for efficient decision-making. In a global context, our solution has the potential to address the growing demand for diagnostic imaging services, particularly in resource-constrained settings where access to specialized expertise may be limited.

Conclusion and Future Work

The project presents an effective technique for detecting and classifying brain tumors. It utilizes data preprocessing and data augmentation techniques. Transfer learning is employed through six different neural network architectures: MobileNet, EfficientNet, ResNet, DenseNet, Inception, and Xception. They achieved accuracies of 96% for EfficientNet and Inception, 95% for both DenseNet and Xception, 94% for ResNet, and 69% for MobileNet. EfficientNet is selected as the model to be deployed on a platform, facilitated by FastAPI.

This project has numerous applications, such as improving tumor diagnosis, facilitating targeted treatment planning, and aiding practitioners in making informed decisions.

Future work involves enhancing real-time tumor detection, potentially using YOLO during scanning. Tumor segmentation needs to be included, and the model should be trained to learn from previous cases to understand tumor behavior and growth rates, as well as effective mitigation strategies. Further training can enable the model to assist doctors in locating tumors during operations and providing guidance. By leveraging previous case records, the model can suggest various treatment and drug usage options, benefiting people worldwide.

Cancer is a significant challenge, and early detection, accurate diagnosis, and effective treatment can help eradicate this ailment.

References

[1] **MRI brain tumor detection and classification using parallel deep convolutional neural networks:**

Takowa Rahman, Md Saiful Islam, Measurement: Sensors, Volume 26, 2023, 100694, ISSN 2665-9174

https://doi.org/10.1016/j.measen.2023.100694.

(https://www.sciencedirect.com/science/article/pii/S2665917423000302)

[2] **Enhancement of brain tumor classification from MRI images using multi-path convolutional neural network with SVM classifier:**

Sahar Khoramipour, Mojtaba Gandomkar, Mohsen Shakiba,

Biomedical Signal Processing and Control, Volume 93, 2024, 106117, ISSN 1746-8094,

https://doi.org/10.1016/j.bspc.2024.106117.

(https://www.sciencedirect.com/science/article/pii/S1746809424001757)

[3] **Dermatological infrared thermal imaging with human-machine interaction image diagnostics interface using DenseNet:**

Xing She, Honglei Lu, Qian Liu, Peng Xie, Qian Xia, Journal of Radiation Research and Applied Sciences, Volume 17, Issue 1, 2024, 100826, ISSN 1687-8507, https://doi.org/10.1016/j.jrras.2024.100826.

(https://www.sciencedirect.com/science/article/pii/S1687850724000104)

[4] **Crossover smell agent optimized multilayer perceptron for precise brain tumor classification on MRI images:**

Muthuvel Arumugam, Arunprasath Thiyagarajan, Lakshmi Adhi, Shyamala Alagar, https://doi.org/10.1016/j.eswa.2023.121453.

(https://www.sciencedirect.com/science/article/pii/S0957417423019553)

[5] https://www.hopkinsmedicine.org/health/conditions-and-diseases/brain-tumor/brain-tumor-types

[6] **Dataset:**

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

[7] **New Dataset:**

https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri

[8] Title= EfficientNetV2: Smaller Models and Faster Training, author = Mingxing Tan and Quoc V. Le, year = 2021

https://arxiv.org/abs/2104.00298