Manipal University
Jaipur

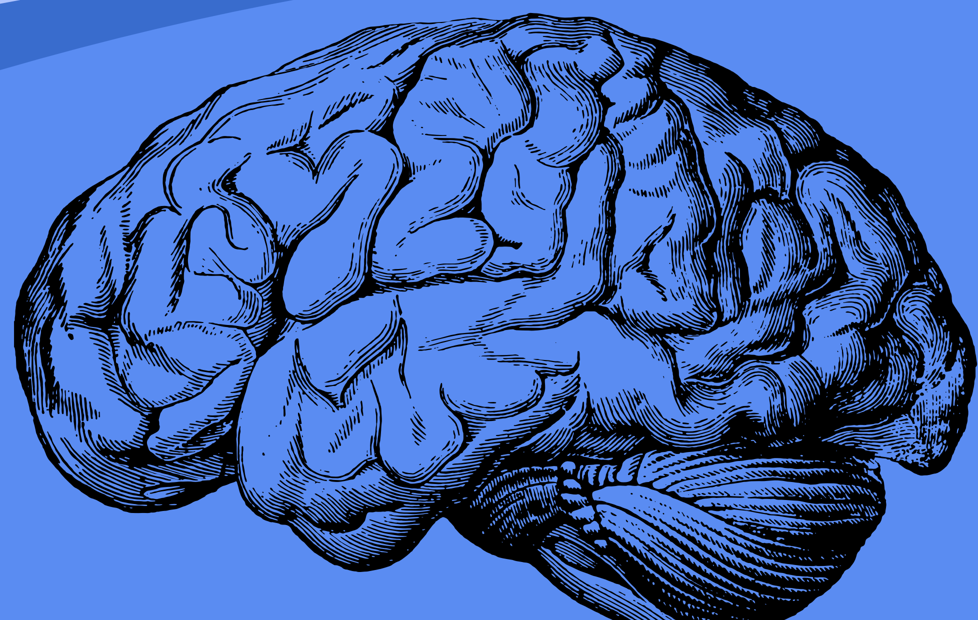
MINOR PROJECT

Brain Tumor Detection

Under the Guidance of Dr. Satyabrata Roy

Sonam Chhikara - 209301273

Ayush Kumar - 209301457



CONTENT

WHY WE CHOSE THIS PROBLEM STATEMENT

- Brain tumors are a prevalent health issue with a high mortality rate, making early detection critical.
- Medical imaging is crucial for brain tumor detection but can be challenging to interpret.
- Machine learning and deep learning techniques, such as CNNs, have shown great promise in medical image analysis, including brain tumor detection.
- Developing a CNN model for brain tumor detection aligns with the growing trend of AI-assisted diagnosis and treatment in healthcare.
- The project aims to build an efficient AI assistant for doctors for early screening of brain tumor.

PROJECT OBJECTIVE

- Collecting and preprocessing medical images of the brain that contain tumors or are tumor-free.
- Developing and training customized deep learning models to classify brain images as either tumor or non-tumor.
- Evaluating the performance of the model in terms of accuracy, f1 score, precision, recall value etc.
- Developing a user-friendly interface to interact with the model and visualize the results.

EXISTING WORKS AND TECHNIQUES

Method	Pros	Cons	Example
Machine Learning	• Machine learning can detect brain tumors by analyzing medical images and identifying patterns characteristic of tumors.	• Limited reliability • Dependent on large, high-quality data sets	• Support Vector Machines (SVMs) • Decision Tree
Deep Learning	• Deep learning is an advanced type of machine learning that can analyze complex medical images and identify subtle patterns or features that are characteristic of tumors. • It has a high accuracy and efficiency in detecting brain tumors, as it can learn from large datasets of medical images and tumor characteristics, improving its performance over time.	• Dependent on large, high-quality data sets • Limited reliability	• Convolutional Neural Networks (CNNs) • Recurrent Neural Networks (RNNs)

PROJECT REQUIREMENTS

SOFTWARE:

- Image processing and analysis software, such as MATLAB, Keras or Python libraries (e.g., OpenCV, scikit-image)
- Machine learning libraries, such as scikit-learn, TensorFlow.
- Deep learning libraries, such as TensorFlow.
- Data visualization and analysis software, such as MATLAB, Python libraries (e.g., Matplotlib, Seaborn).

HARDWARE:

- High-performance computing system, such as a desktop computer or a GPU-accelerated server, for training and evaluating ML and DL models.
- Large-capacity storage system for storing medical images and results of the analysis.

WORKING OF OUR PROJECT

STEP-1 (Dataset Used)

- We used dataset of brain MRI images from Kaggle.
- The dataset contains 2 folders i.e. yes and no which contains 253 Brain MRI Images. The folder yes contains 155 Brain MRI Images that are tumorous and the folder no contains 98 Brain MRI Images that are non-tumorous.

STEP-2 (Data Augmentation)

- Since this is a small dataset, we used data augmentation in order to create more images. This was also useful in tackling the data imbalance issue. Dataset used consisted 61% of the data (155 images) as tumorous and 39% of the data (98 images) as non-tumorous. So, in order to balance the data we generated 9 new images for every image that belongs to 'no' class and 6 images for every image that belongs the 'yes' class.

WORKING OF OUR PROJECT

STEP-3 (Data Preprocessing)

- Crop the part of the image that contains only the brain (which is the most important part of the image).
- To ensure that the images in our dataset are compatible with our neural network, we need to resize them to a consistent shape of (240, 240, 3). This is important because the images in the dataset come in different sizes and formats, which can cause issues when feeding them as inputs to our neural network.
- Apply normalization: to scale pixel values to the range 0-1.

STEP-4 (Data split)

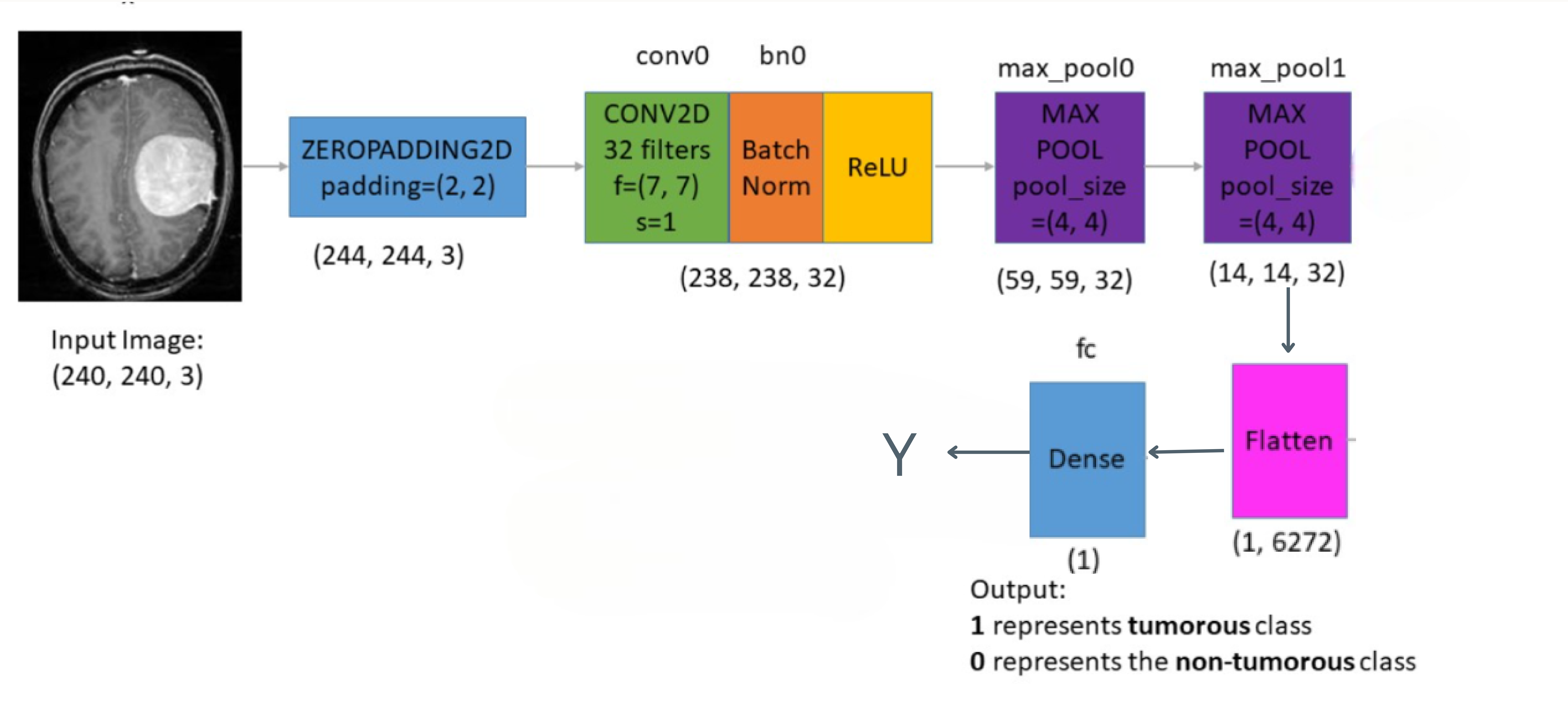
The data was split in the following way:

- 70% of the data for training.
- 15% of the data for validation.
- 15% of the data for testing.

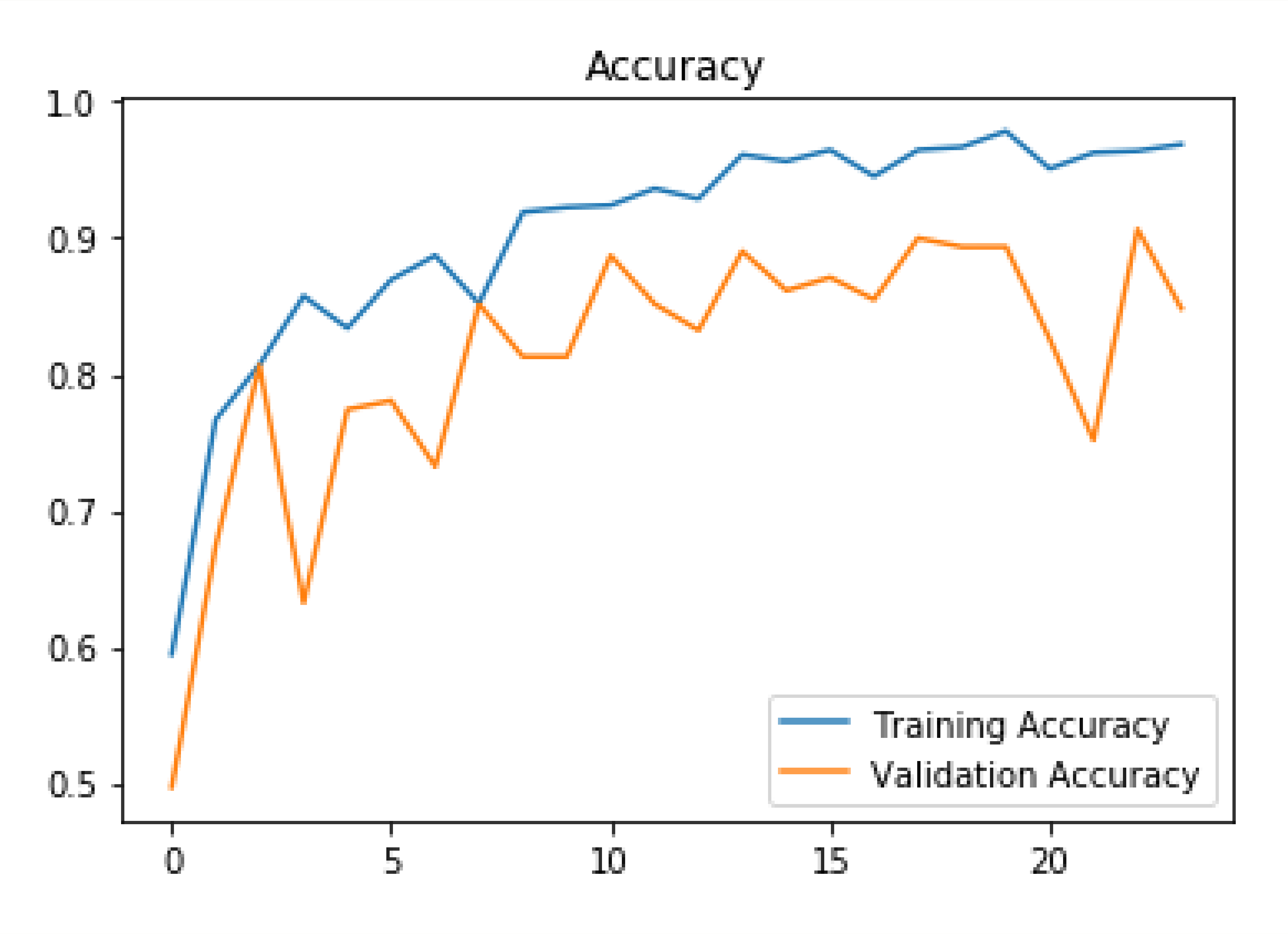
STEP-5 (Applying CNN model)

- The architecture is a Convolutional Neural Network (CNN) for binary classification.
- The first layer is a Zero Padding layer with a pool size of (2,2) to prevent image size reduction during convolution.
- The second layer is a Convolutional layer with 32 filters of size (7,7) and stride of 1, producing a 32-channel feature map.
- The third layer is a Batch Normalization layer to normalize the feature map.
- The fourth layer is a ReLU Activation layer to introduce non-linearity into the network.
- The fifth and sixth layers are Max Pooling layers with pool size of (4,4) and stride of 4 to reduce feature map size.
- The seventh layer is a Flatten layer to convert the 3D feature map into a 1D vector.
- The eighth and final layer is a Dense output layer with one neuron and a sigmoid activation function, producing the binary classification prediction.
- The network uses a combination of convolutional, activation, pooling, and fully connected layers to extract features and make a prediction.

NEURAL NETWORK ARCHITECTURE



ACCURACY


Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0
5
10
15
20
Training Accuracy
Validation Accuracy

RESULTS AND EVALUATION

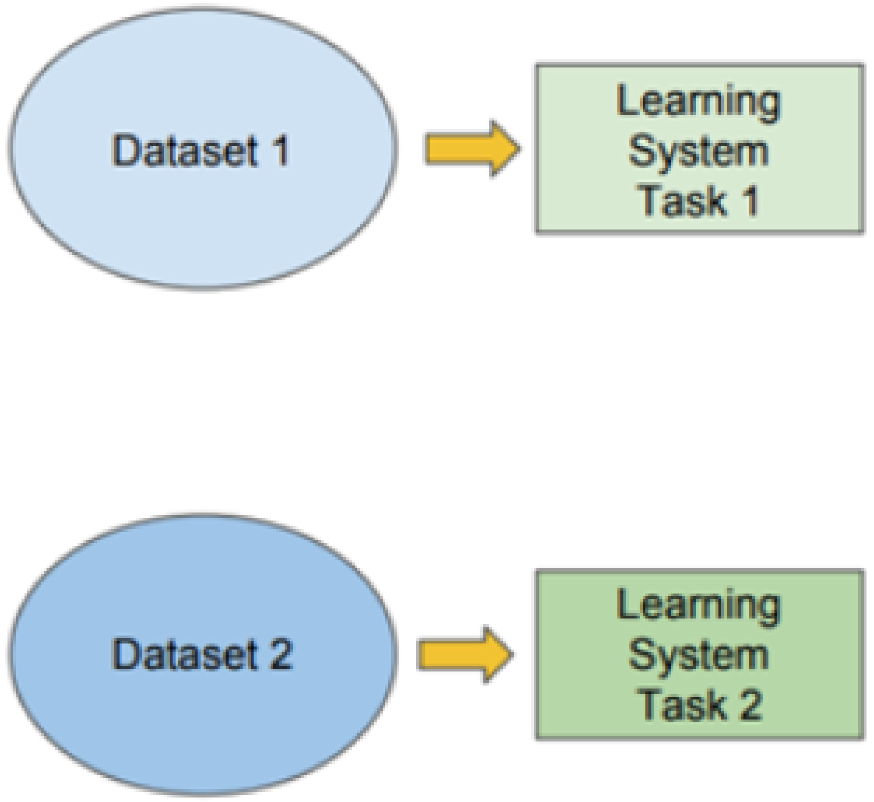
- The best-performing model achieved 90% accuracy on the test set, correctly identifying brain tumors with high accuracy.
- The model achieved an f1 score of 0.90 on the test set, indicating both precision and recall were high.
- The overall performance of the model was good in detecting brain tumors.
- Further improvements can be made by training the model on more epochs.

Performance table of the best model:

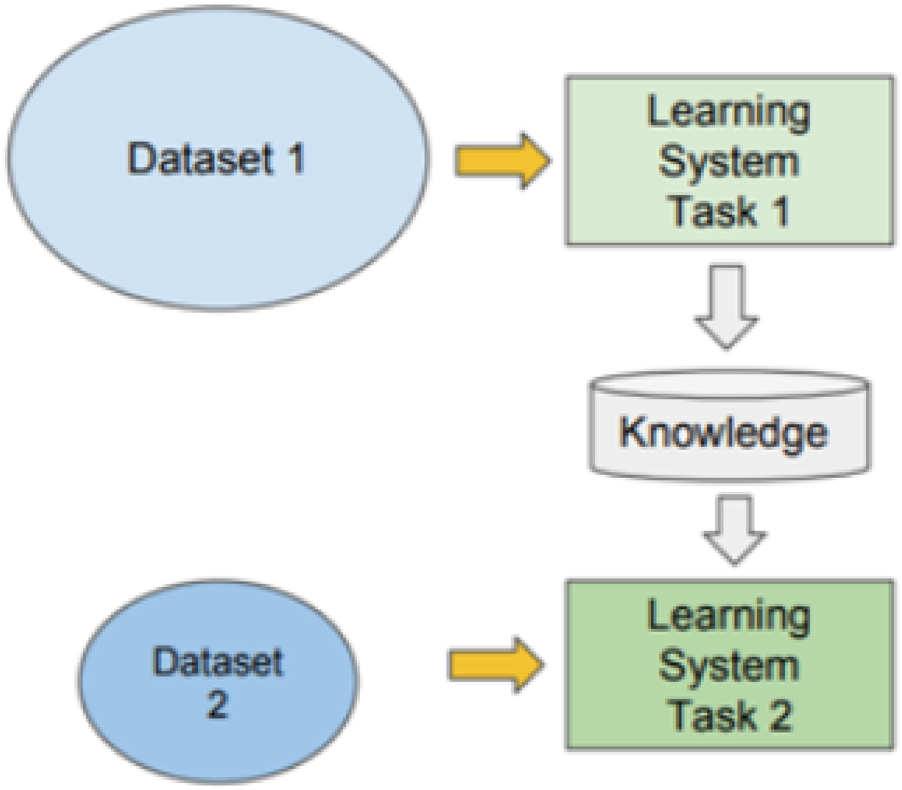
	Validation Set	Test Set
Accuracy	**91%**	**90%**
F1 Score	**0.91**	**0.90**

Traditional ML vs Transfer Learning

- Isolated, single task learning:
  - Knowledge is not retained or accumulated. Learning is performed w.o. considering past learned knowledge in other tasks

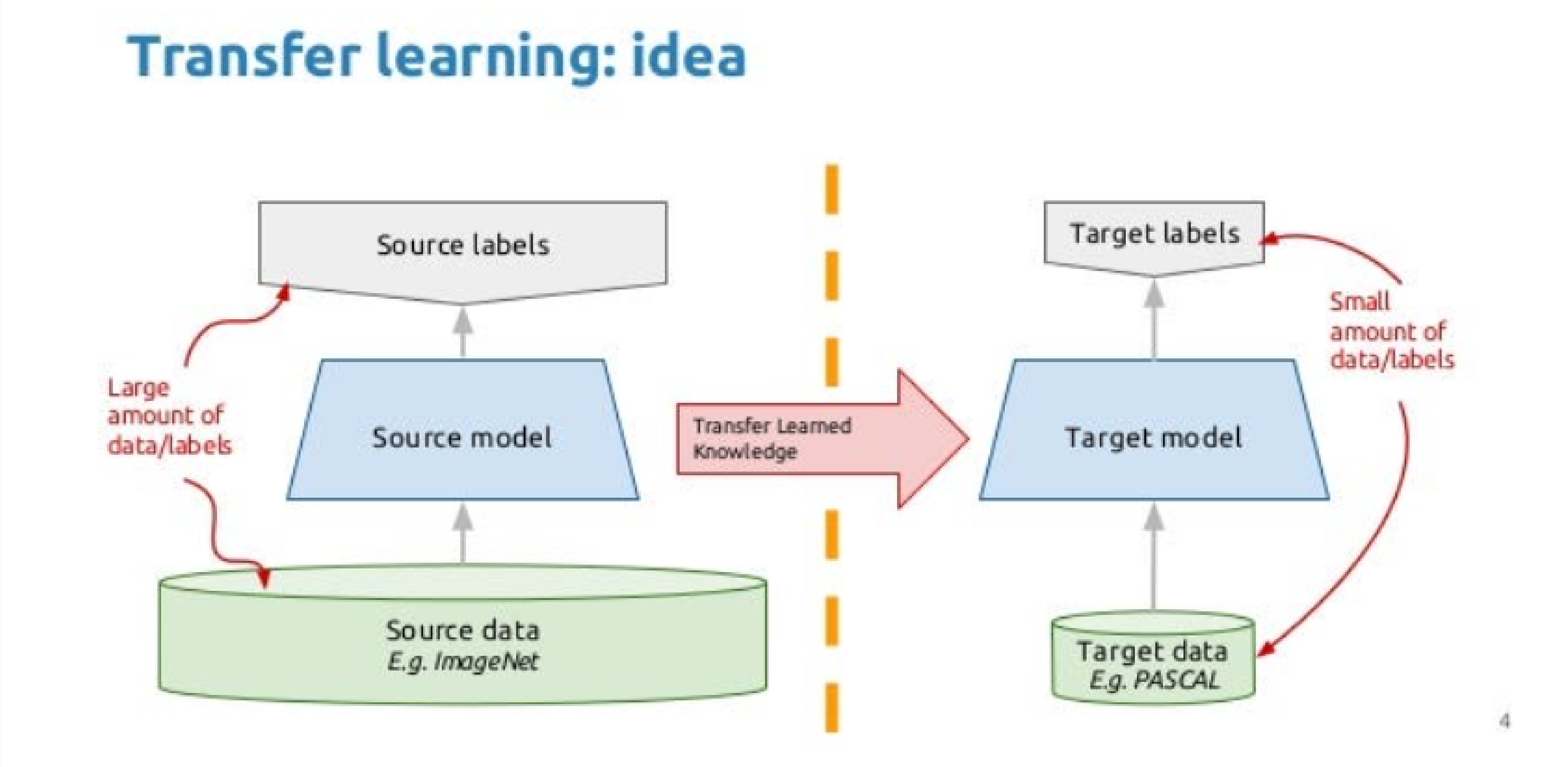


- Learning of a new tasks relies on the previous learned tasks:
  - Learning process can be faster, more accurate and/or need less training data



Transfer learning is a machine learning technique that involves using a pre-trained model as a starting point for training a new model on a different dataset. Here's how it works:

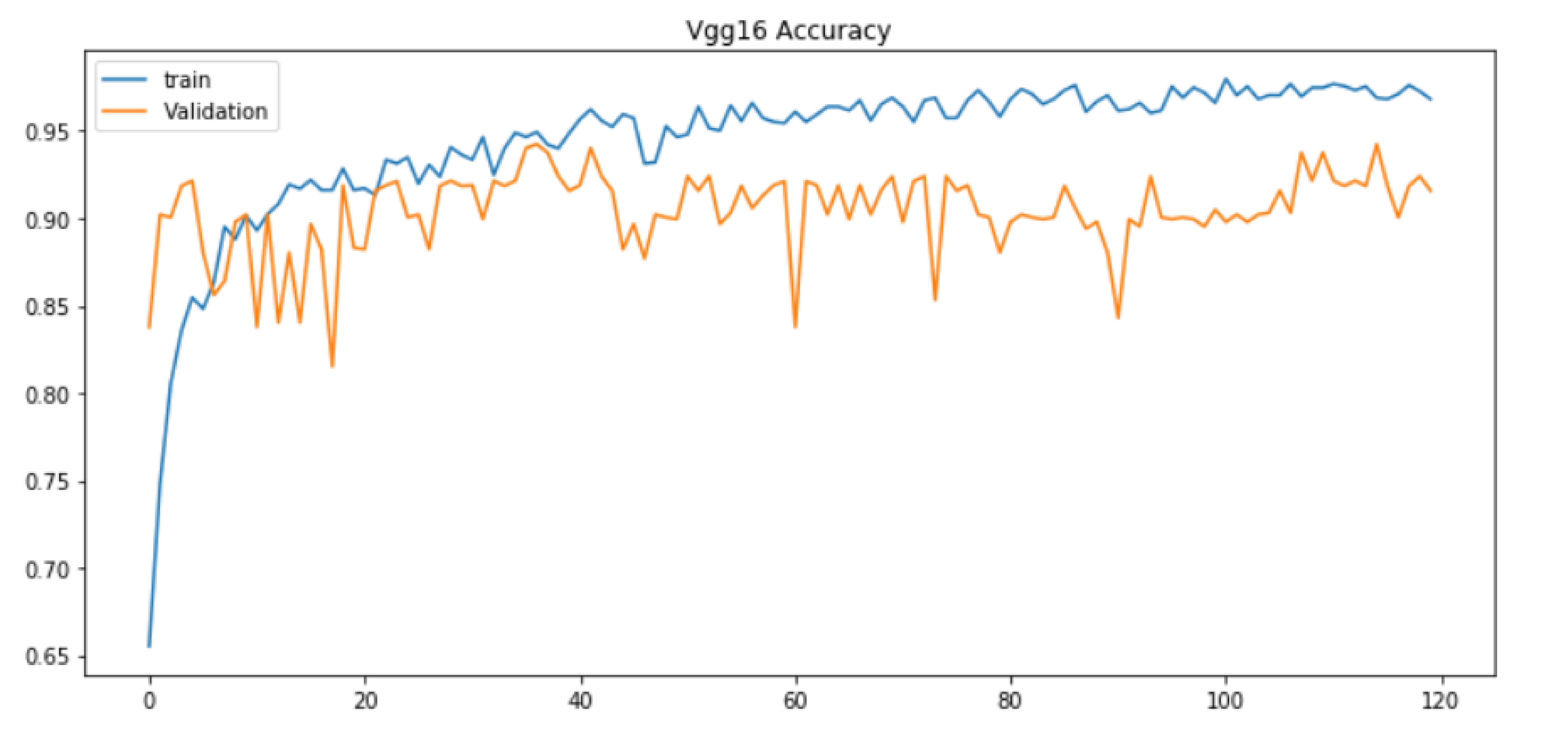
- Source data: The original dataset used to train the pre-trained model.
- Source model: A pre-trained model that has already been trained on a large dataset, such as CNN, to recognize a wide range of features.
- Source labels: The classification labels assigned to the data in the source dataset.
- Target data: A new dataset that you want to train a model on, which may be different from the source data.
- Target model: A new model that is trained on the target data using the pre-trained model as a starting point.
- Target labels: The classification labels assigned to the data in the target dataset.



STEP-6 (Applying VGG-16)

- Input size: The VGG-16 network takes as input images of size 224x224 pixels.
- Layers: The network consists of 16 layers, including 13 convolutional layers and 3 fully connected layers.
- Convolutional layers: The convolutional layers use 3x3 filters with a stride of 1 and padding of 1. The first convolutional layer has 64 filters, and the subsequent layers have either 128 or 256 filters.
- Max pooling: After every two convolutional layers, the network applies max pooling with a 2x2 filter and a stride of 2.
- Activation function: The network uses the Rectified Linear Unit (ReLU) activation function after each convolutional and fully connected layer.
- Fully connected layers: The three fully connected layers have 4096 units each, followed by a final output layer with two units for binary classification (tumor present or not).
- Training: The network is trained using backpropagation with stochastic gradient descent (SGD) and dropout regularization.
- Pretrained model: A pretrained VGG-16 model is often used as a starting point for transfer learning on new datasets. This can help to improve performance and reduce training time.

ACCURACY

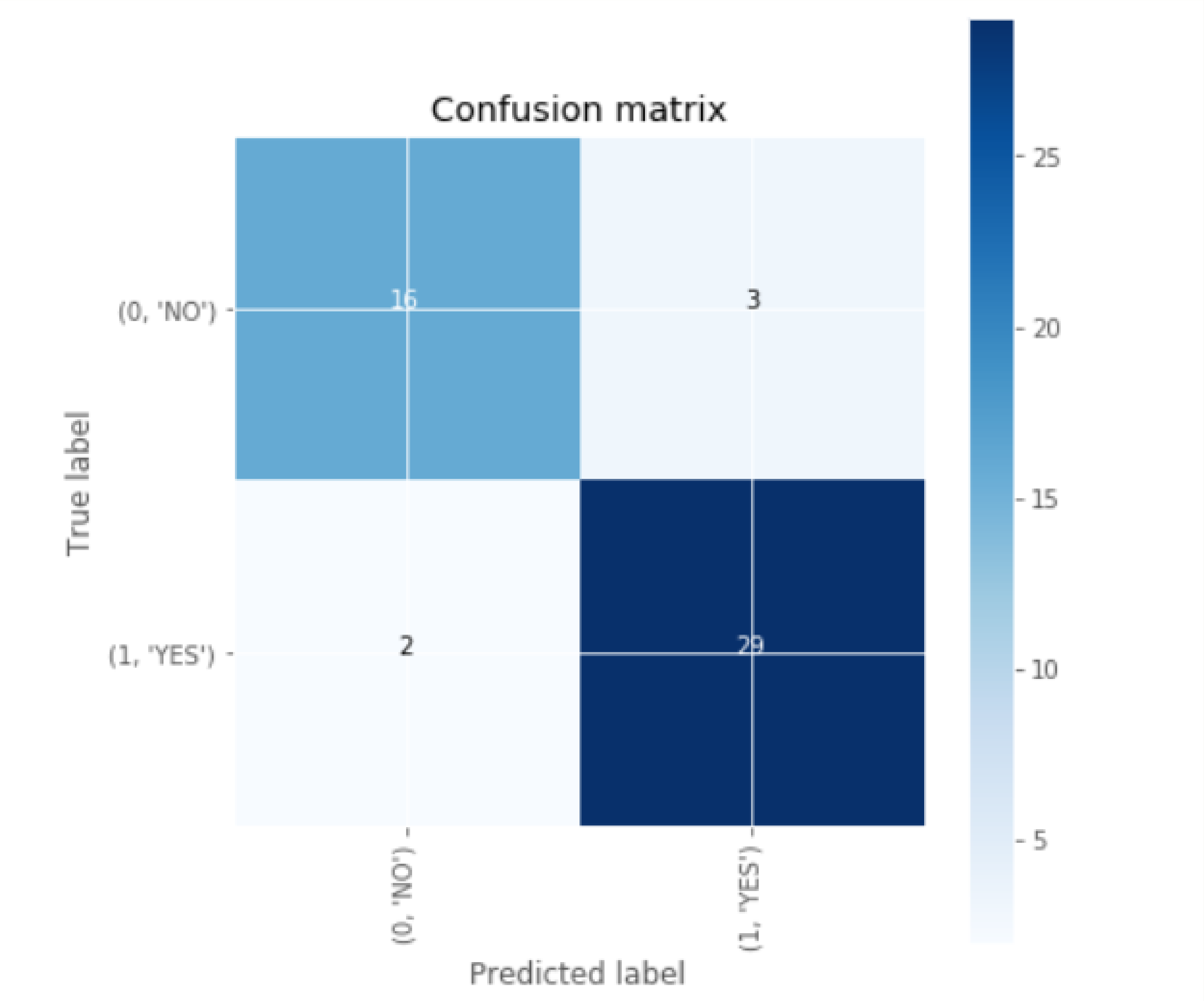


Vgg16 Accuracy
train
Validation
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0
20
40
60
80
100
120

RESULTS AND EVALUATION

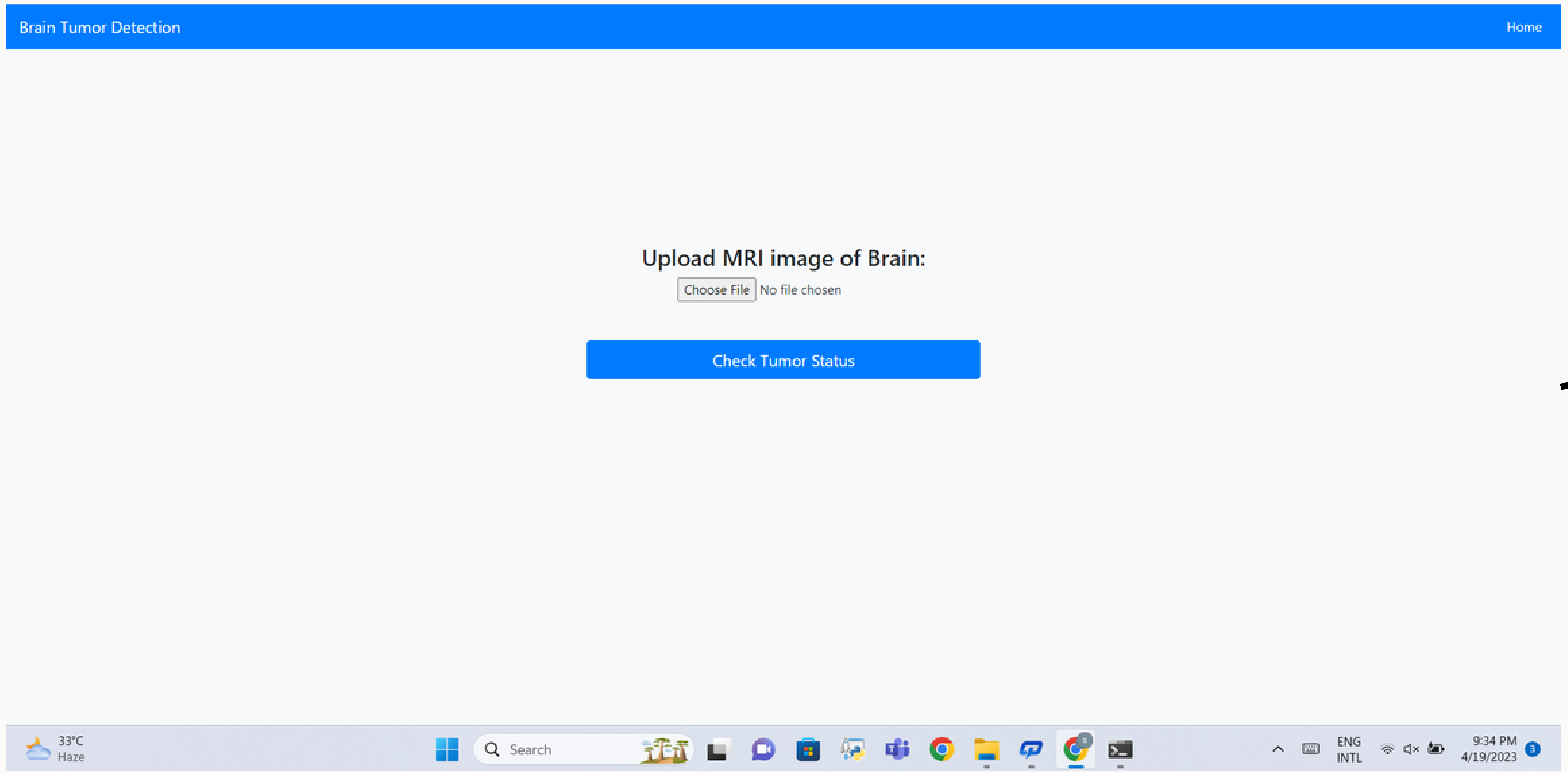
- Accuracy: 0.950000
- Precision: 0.954286
- Recall: 1.000000
- F1 score: 0.974333

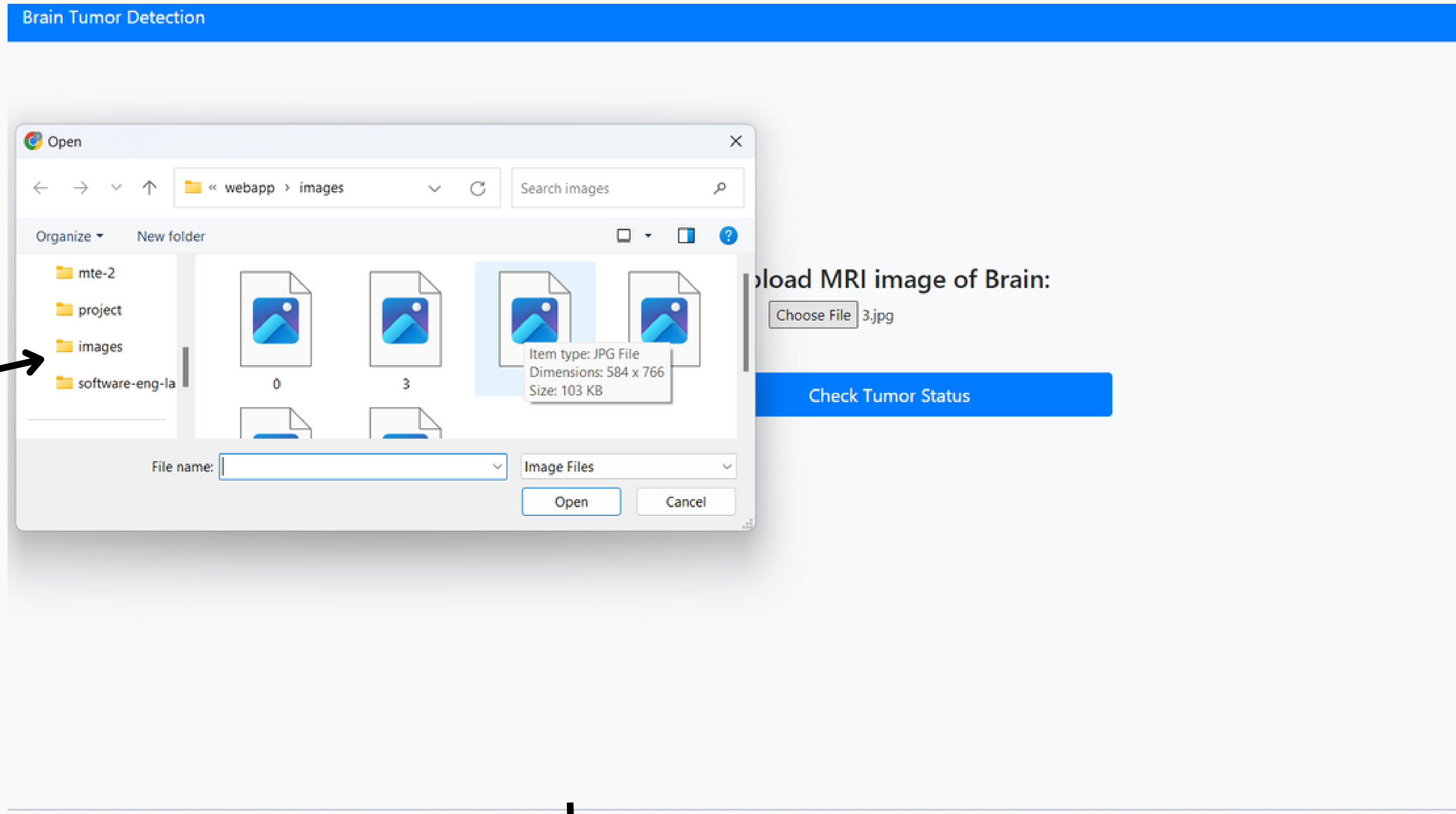
Confusion Matrix



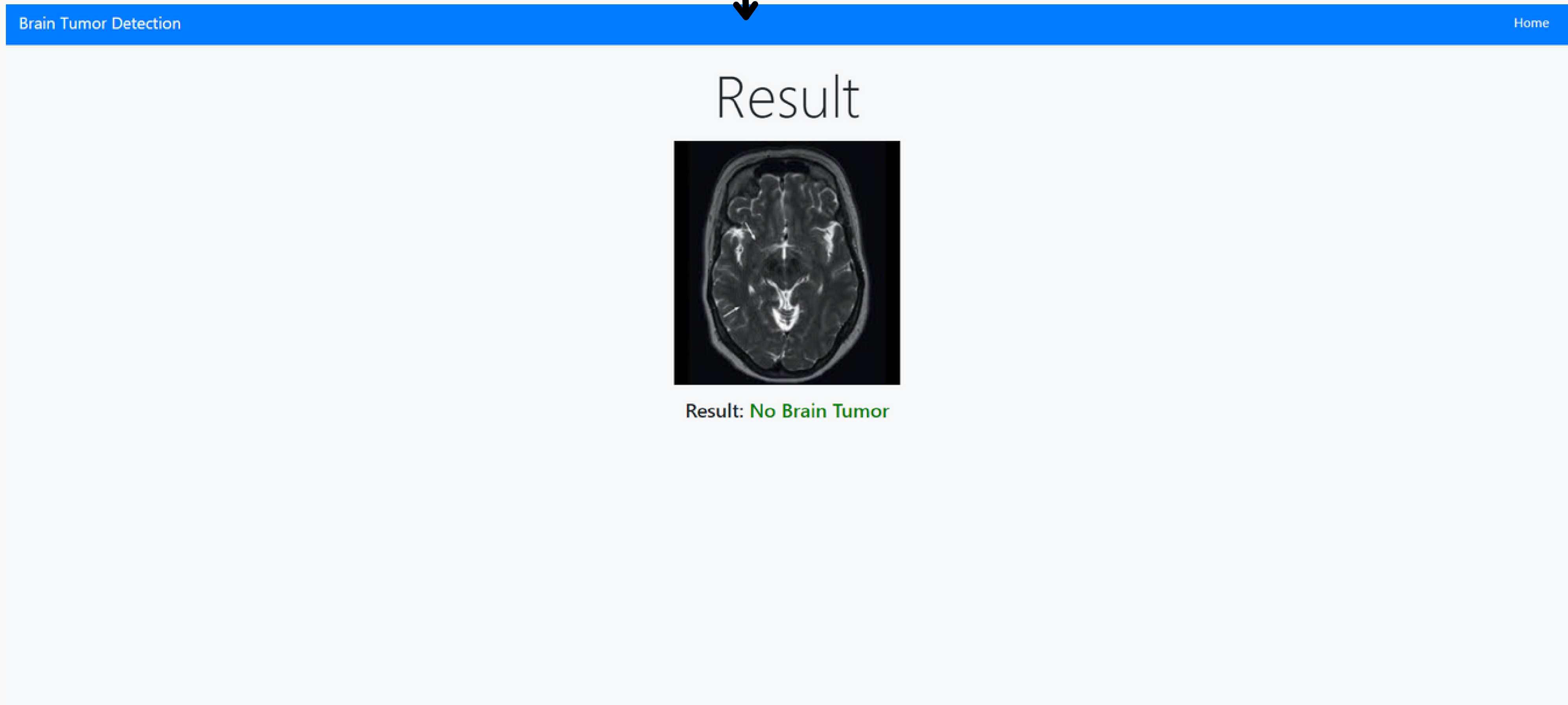
STEP-7 (Frontend)

- A web application has been developed using Flask framework, which can determine whether a given medical image contains a brain tumor or not.
- At the backend, a Convolutional Neural Network (CNN) model has been used, which has been trained on a large dataset of medical images to accurately classify them as tumor or non-tumor.
- When a user uploads a medical image through the web application, the image is passed through the CNN model, which analyzes the image and predicts whether it contains a brain tumor or not.
- The result is then displayed to the user on the web page. This can help doctors and patients to quickly identify the presence of brain tumors and take appropriate action.


Brain Tumor Detection
Home
Upload MRI image of Brain:
Choose File No file chosen
Check Tumor Status
33°C
Haze
Search
ENG
INTL
9:34 PM
4/19/2023


Brain Tumor Detection
Open
webapp › images
Search images
Organize
New folder
mte-2
project
images
software-eng-la
0
3
Item type: JPG File
Dimensions: 584 x 766
Size: 103 KB
File name:
Image Files
Open
Cancel
load MRI image of Brain:
Choose File 3.jpg
Check Tumor Status

Frontend


Brain Tumor Detection
Home
Result
Result: No Brain Tumor

REFERENCES

- L. Guo, L. Zhao, Y. Wu, Y. Li, G. Xu, and Q. Yan, “Tumor detection in MR images using one-class immune feature weighted SVMs,” IEEE Transactions on Magnetics, vol. 47, no. 10, pp. 3849–3852, 2011.
- Jin Liu, Min Li, J. Wang, Fangxiang Wu, T. Liu Yi. Pan. A survey of MRI-based brain tumor segmentation methods. TSINGHUA SCI TECHNOL 2014;19(6): 578-595.
- Michael R. Kaus, “Automated Segmentation of MR Images Of Brain Tumors”, Radiology 2001; 218:586–591, 2014
- Dipali M. Joshi, N. K. Rana, V. M. Misra, " Classification of Brain Cancer Using Artificial Neural Network", IEEE International Conference on Electronic Computer Technology, ICECT ,2010.
- Abiwinanda N, Hanif M, Hesaputra ST, Handayani A, Mengko TR (2019) Brain tumor classification using convolutional neural network. IFMBE Proc 68(1):183–189.
- G. S. Tandel et al., "A review on a deep learning perspective in brain cancer classification", Cancers, vol. 11, no. 1, pp. 1-32, 2019.