Brain Tumor Detection using Deep Learning

AI Semester Project
21BSCS08, 21BSCS014, 21BSCS022

Brain Tumor Detection with AI

Our project uses artificial intelligence to detect brain tumors in MRI scans. Early diagnosis saves lives.

We've created a system that helps doctors identify tumors faster and more accurately, especially in areas with limited medical resources.

Importance

- **Early Detection Critical**

Delays in brain tumor diagnosis can have serious consequences.

- **Increasing Prevalence**

Brain tumors are becoming more common globally

.

- **AI Assistance**

Technology can help doctors diagnose faster and better.

- **Global Access**

Our solution helps areas with few medical experts.

Aim

- Our main goal is to create a system that can automatically find brain tumors in MRI scans.

Objectives

- Improve the speed and accuracy of detecting brain tumors compared to traditional methods.
- Develop a user-friendly interface that doctors can easily use.
- Ensure the system is effective in remote areas where medical resources are limited.
- Validate the model's performance through thorough testing and evaluation.
- Explore future enhancements to expand the system's capabilities.

Benefits of our system?

- It can find brain tumors quickly and accurately.
- It reduces the amount of work doctors have to do.
- It can be used in places with limited medical resources.
- It can be part of telehealth apps, making it easier for patients to get help.
- It learns well even from a small amount of data.

Nature of dataset

We worked with a set of 253 brain MRI images. Each image was labeled as either showing a tumor ('yes') or not showing a tumor ('no').To prepare the images for our model, we resized them to a standard size and adjusted the colors so the model could understand them better.

Methodology

Data Collection

253 labeled MRI images (tumor/no tumor) from public dataset.

Preprocessing

Resized to 224x224 pixels and normalized pixel values.

Data Augmentation

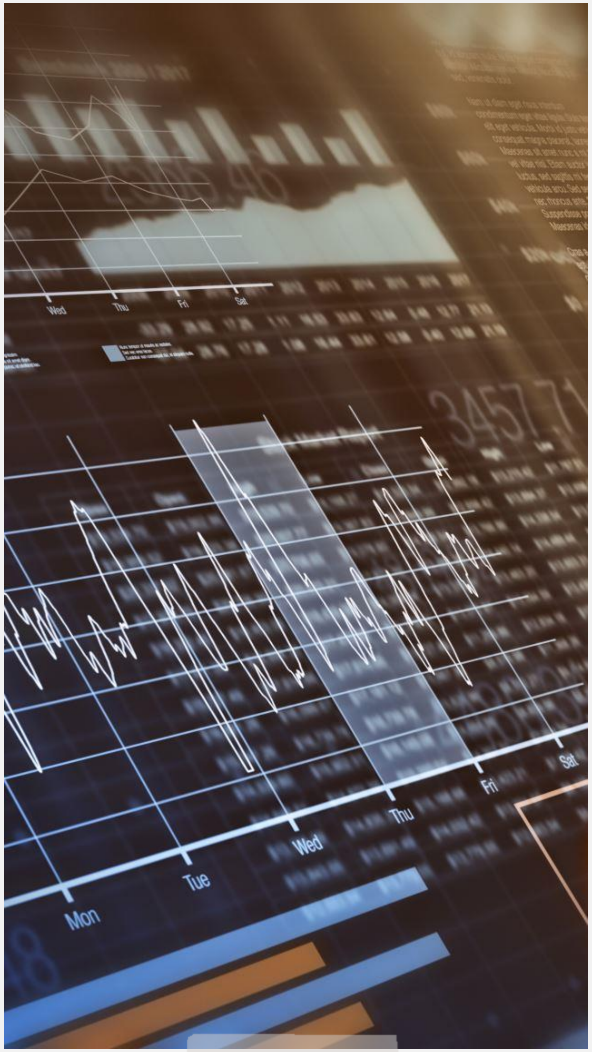
Created variations by rotating images up to 15 degrees.

Model Building

Used VGG16 with transfer learning and custom layers.

Methodology

1. **Data Loading and Labeling:** We collected MRI images from a publicly available dataset. The images were organized into two folders: one for images with tumors and one for images without tumors. We used Python’s OpenCV library to load and label these images.



Methodology

1. **Preprocessing:** To ensure the model could handle all the images consistently, we performed several preprocessing steps:
   - **Resizing:** All images were resized to 224x224 pixels to match the input size required by our model.
   - **Normalization:** We scaled the pixel values to be between 0 and 1, which helps the model learn better.

Methodology

- **Label Encoding:** We converted the labels into a binary format, making it easier for the model to understand.
- **Data Augmentation:** we used data augmentation techniques to create variations of the training images. This included randomly rotating images up to 15 degrees, which helps the model become more flexible and perform better on new images.

Methodology

3. **Model Building Using Transfer Learning:** We used the VGG16 model, which is a pre-trained convolutional neural network. This model has already learned to detect important features in images. We froze the original layers of VGG16 and added custom layers for our specific task, including:

    1. AveragePooling2DFlattenDense layer with ReLU activationDropout layer to prevent overfittingDense output layer with Softmax activation for binary classification.
    2. Model Training: We trained the model using 90% of our data and tested it on the remaining 10%. We used the Adam optimizer with a learning rate of 0.001 and trained for 10 epochs with a batch size of 8 images.
    3. Evaluation and Visualization: After training, we evaluated the model using a classification report and confusion matrix to check its performance. We also monitored accuracy and loss graphs to ensure the model was learning effectively.

Model Performance

We used 90% of our data to train the model and 10% to test it.I’m happy to say that our model achieved about 96.5% accuracy! We used tools like confusion matrices to check how well the model was learning and to make sure it wasn’t just memorizing the data.

Model Performance

96.5%

Accuracy

Our model correctly identifies tumors in most cases.

90%

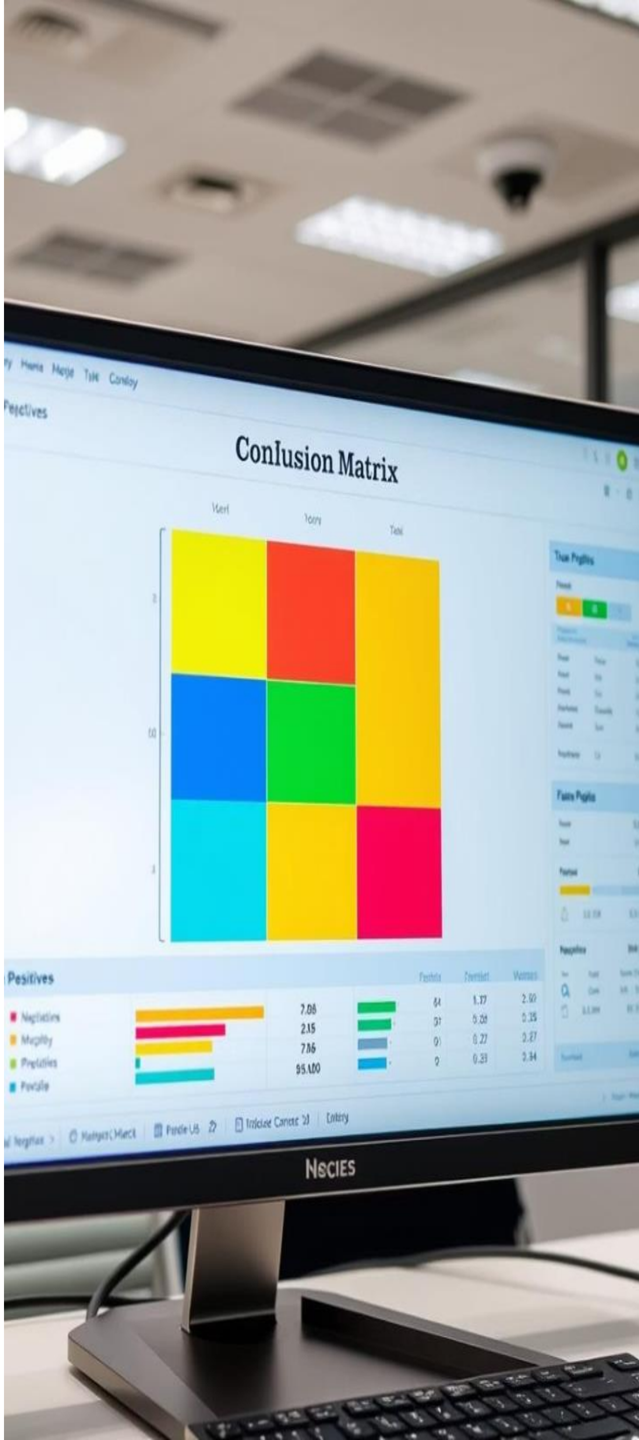
Training Data

Percentage of dataset used to train the model.

10%

Testing Data

Portion used to validate performance.



Implementation Scope

Our system can be used in many ways:

- As a helper for doctors in hospitals.
- In telemedicine apps for remote checkups.
- As a learning tool for students studying brain imaging.
- To support research on brain tumors.
- To help doctors confirm their diagnoses.

Results

Our model had a final accuracy of **96.5%**, meaning it correctly identified brain tumors in about 96 out of 100 casses.

The confusion matrix showed that the model did a great job identifying both 'tumor' and 'no tumor' cases, with very few mistakes. This means our model is reliable and can be a helpful tool in real healthcare situations."

Conclusion

our deep learning system effectively detects brain tumors from MRI images using the VGG16 model and transfer learning. Achieving high accuracy with limited data, our project showcases AI's potential to enhance medical diagnostics, alleviate doctor workloads, and improve patient care, particularly in underserved areas.

References

- [1] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv preprint arXiv:1409.1556, 2014. https://arxiv.org/abs/1409.1556
- [2] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," arXiv preprint arXiv:1603.04467, 2016. https://arxiv.org/abs/1603.04467
- [3] DataFlair, "Brain Tumor Classification using Machine Learning. https://data-flair.training/blogs/brain-tumor-classification-machine-learning/