International Journal of Advanced Research in Education and TechnologY (IJARETY)

Volume 11, Issue 4, July-August 2024

Impact Factor: 7.394

International Journal of Advanced Research in Education and TechnologY(IJARETY)

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 7.394| A Bi-Monthly, Double-Blind Peer Reviewed & Referred Journal |

|| Volume 11, Issue 4, July-August 2024 ||

DOI:10.15680/IJARETY.2024.1104039

Automated Detection of Brain Tumors using Convolutional Neural Networks

Dr. R. Jayanthi, Shashikumara A K, S Girish

Associate Professor, Department of MCA, Dayananda Sagar College of Engineering, Bengaluru, Karnataka, India

P.G. Student, Department of MCA, Dayananda Sagar College of Engineering, Bengaluru, Karnataka, India

P.G. Student, Department of MCA, Dayananda Sagar College of Engineering, Bengaluru, Karnataka, India

ABSTRACT: A brain tumor is an abnormal collection of cells within the brain's intracra nial space. Due to the rigidity of the skull, any tumor growth can exert pressure, potentially causing significant harm. This condition is a serious concern for both children and adults, with approximately 11,700 new cases diagno sed annually. Brain tumors represent 85-90% of all central nervous system tumors, and the 5-year survival rate for malignant cases stands at about 34% for men and 36% for women. Accurate diagnosis and early treatment are crucial for improving patient outcomes.Brain tumor segmentation is a critical and challenging task in medical imaging. Manual classification can be prone to errors, especially with large datasets. The variability in tumor appearance and the similarity between tumor and normal tissues further complicate the extraction process. Various imaging techniques, such as X-rays, MRIs, and CT scans, are utilized for tumor detection. This study focuses on using computed tomography (CT) scan images to identify brain tumors.We employ deep learning (DL) technology, which has shown promise in enhancing detection and classification accuracy. Specifically, we developed a model utilizing a convolutional neural network (CNN) that achieved a 97.87% accuracy rate in detecting brain tumors.

KEYWORDS: Brain Tumor, Magnetic Resonance Imaging (MRI), Convolutional Neural Networks (CNN), Deep Learning, TensorFlow, Keras.

I. INTRODUCTION

Medical imaging technology provides crucial insights into the human body, particularly in diagnosing and classifying tumors or cancers. Brain tumors, among the most lethal types of cancer, significantly contribute to global mortality rates. According to the International Agency for Research on Cancer (IARC), over one million people are diagnosed with brain tumors annually, with an increasing mortality rate. This trend is particularly concerning for those under the age of 34.

Advancements in imaging techniques, such as computed tomography (CT) and magnetic resonance imaging (MRI), have enabled doctors to detect abnormalities within the body more effectively. The increasing demand for labor-saving techniques and the clinical evaluation of extensive medical data have highlighted the importance of brain tumor detection. Image processing and analysis involve complex data calculations and visualizations to understand these abnormalities.

Brain tumors can originate from abnormal, precancerous, cancerous, harmful, harmless, or non-cancerous cells in the brain. Malignant tumors are categorized into two types: those originating in the brain and secondary cancerous growths that can spread to other body parts. Metastatic cancer has a low survival rate and high mortality rate due to its ability to spread within the body.

Brain Tumor

The brain, a soft tissue mass covered by the skull, is composed of the cerebrum, cerebellum, and brainstem. It responds to external stimuli, maintains body equilibrium, and connects to the spinal cord. Brain tumors are characterized by the abnormal growth of brain cells, classified as primary (originating in the brain) or secondary (spread from other body parts). Symptoms of metastatic tumors include seizures, speech difficulties, nausea, and blurred vision. Without timely treatment, these tumors can be fatal due to decreased blood flow and increased intracranial pressure.

Electronic healthcare systems and magnetic field information technology enhance clinical care. Brain tumors affect nearby structures, distinguished by location, image intensity, shape, and size. Benign tumors, which grow slowly and do not invade surrounding tissues, contrast with malignant tumors, which grow rapidly and are potentially fatal. Early

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 7.394| A Bi-Monthly, Double-Blind Peer Reviewed & Referred Journal |

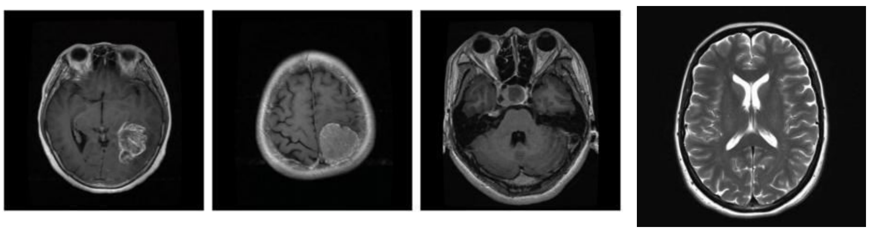
|| Volume 11, Issue 4, July-August 2024 ||

DOI:10.15680/IJARETY.2024.1104039

identification of cancerous tumors is crucial for essential treatment. Automated detection and classification techniques are essential for accuracy and reducing false-negative rates.

Magnetic resonance imaging (MRI) is a key tool in diagnosing and monitoring diseases due to its high spatial resolution and soft-tissue contrast. MRI uses strong magnetic fields and radio waves to produce detailed images, making it invaluable for detecting and treating malignancies and neurological abnormalities.

Segmentation of MRI images is vital for diagnosing brain malignancies. This preprocessing step divides images into smaller segments, focusing on regions of interest. While manual segmentation is possible, it is time-consuming and less accurate. Automated methods provide more efficient and precise results.



Glioma Meningioma Pituitary No Tumor

II. Literature Review

K. B. Vaishnavee and K. Amshakala
Vaishnavee and Amshakala proposed an automated approach for MRI brain image segmentation and tumor detection using SOM-clustering and a Proximal Support Vector Machine (PSVM) classifier. Their method focuses on extracting suspicious areas from medical images such as MRIs and CT scans, achieving high accuracy with a low error rate. Future improvements could involve adapting the system to other cancer types and enhancing accuracy through swarm-based feature selection.

G. Birare and V. A. Chakkarwar
Birare and Chakkarwar developed an automated method for detecting brain tumor cells using Support Vector Machine (SVM). Each year, thousands are diagnosed with brain tumors, making it crucial to accurately separate and identify tumor regions from MRI images. They introduced an automatic segmentation technique using the K-means algorithm and an SVM classifier, achieving an accuracy of 98.51%. Future research may involve developing additional algorithms, such as multiclass SVM, to improve diagnostic capabilities.

A. Kumar, A. Ashok, and M. A. Ansari
Kumar, Ashok, and Ansari introduced a hybrid model combining Particle Swarm Optimization (PSO) and SVM for brain tumor classification. Medical imaging, particularly MRI, is essential for disease diagnosis. Their method involves various procedures, including image segmentation, extraction, and classification. Using PSO for feature selection and SVM for classification, their model achieved 95.23% accuracy. Future work could explore bio-inspired algorithms and apply this concept to other medical images.

R. Ahmmed, A. S. Swakshar, M. F. Hossain, and
Ahmmed et al. developed a method for classifying brain tumors and their stages using SVM and Artificial Neural Network (ANN). They used an integration of temperature-based K-Means and modified fuzzy C-Means (TKFCM) algorithms for MRI image segmentation. Their method classifies tumors into benign and four malignant stages, achieving 97.37% accuracy with a bit error rate of 0.0294, outperforming other methods.

C. Saha and M. F. Hossain
Saha and Hossain designed a scheme for classifying MRI brain tumor images using K-means clustering, Nonsubsampled Contourlet Transform (NSCT), and SVM. MRI is a valuable tool for anatomical studies of brain tissue, but manual classification is time-consuming and error-prone. Their automatic scheme uses a median filter for noise removal and K-means clustering for segmentation. NSCT is applied to segmented images, extracting features for SVM classification. Their method achieved 98.86% classification accuracy, demonstrating efficacy but requiring further validation on different databases.

III. SYSTEM REQUIREMENTS SPECIFICATION

3.1 Software Requirements
- Operating System: any Windows OS
- Libraries: Keras, Tensor Flow, Numpy, Scikit-Learn Matplotlib, OpenCV
- Editor: Collab Notebook
- Technologies: Python

3.2 Hardware Requirements
- Processor: i3
- RAM: 4 GB
- Hard disk: 512GB

3.3 Functional Requirements
- Python 3.6.2 or later
- Pip
- NumPy
- Pandas
- Anaconda
- Jupyter Notebook
- Tensor Flow
- Keras

IV. METHODOLOGY

The proposed procedure is divided into various stage and each stage is explained in detail.
- Image input
- Image preprocessing
- Splitting data for testing and training
- Classifying brain tumor images using CNN's
- Obtaining outputs and validating modules

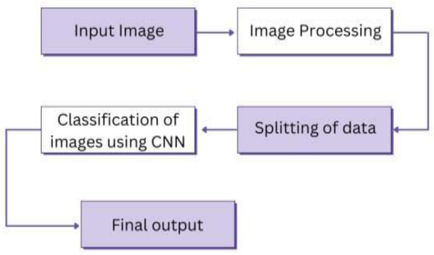


Image Input

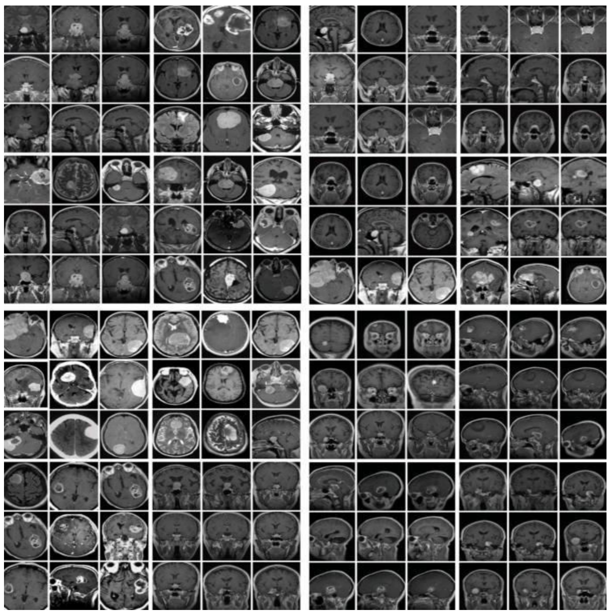
The input is MRI image. The images used here are in .jpg format. The MRI image dataset is obtained from publicly available sources. These images are divided into two yes and no folders, each containing images with and without brain tumors. The training and testing datasets both contain images with and without tumors, we train our system using training dataset and we test our model on test dataset we must not test on the same train dataset if we do so there may be chance to getting a inaccurate result so we test on testing dataset.

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 7.394| A Bi-Monthly, Double-Blind Peer Reviewed & Referred Journal |

|| Volume 11, Issue 4, July-August 2024 ||

DOI:10.15680/IJARETY.2024.1104039

The Dataset of brain with tumor:



The Dataset of brain without tumor:

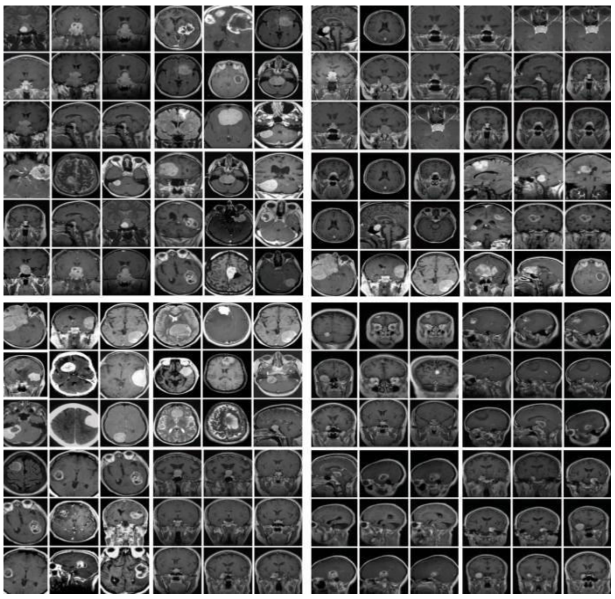


Image Preprocessing

MRI images acquired from public sources often contain noise, such as external interference and patient movement during the scan. For efficient brain tumor detection, these images must be denoised and enhanced. The preprocessing steps are as follows:

- **Grayscale Conversion**: Images imported in jpg format are first converted from RGB to grayscale.
- **Image Resizing**: Grayscale images are resized to a consistent 200 x 200 pixels.
- **Median Filtering**: The resized grayscale image undergoes median filtering to remove noise and reduce edge blurring.
- **High-pass Filtering**: The median filtered image is enhanced with high-pass filtering, and the resized image is added to the resulting image.

Splitting Data for Training and Testing

Testing a model on the same data it was trained on can result in overfitting and poor real-world performance. To avoid this, the dataset is split into training and test sets, typically using an 80/20 split. In this study, the train_test_split() function from the scikit-learn package used, with 20% of the data reserved for testing and 80% for training.

Classification is essential for identifying images, and various algorithms predict image features, each belonging to specific classes. An automatic and reliable convolutional neural network (CNN) classification method is utilized for robust structure and detail identification. CNNs, or ConvNets, are deep learning algorithms that assign importance to different image aspects to distinguish them. ConvNets require less preprocessing compared to other classification algorithms, as they can learn filters and properties through sufficient training.

ConvNets effectively capture spatial and temporal dependencies in images by applying relevant filters. This architecture reduces the number of parameters and reuses weights, making the network more efficient in understanding image sophistication. The ConvNet aims to reduce the image to a manageable form without losing essential features for accurate predictions. The following steps outline the CNN implementation:

1. **Sequential Model Initialization**

- **Initialize Neural Network**: Create an object of the Sequential class.

```
classifier = Sequential()
```

2. **Convolution Layer**

- **Add Convolutional Layer**: Use the add function on the classifier object, passing parameters to Convolution2D. The first parameter is the number of feature detectors (256), followed by the dimensions of the feature detector matrix. The input shape is defined based on the input image format.

```
classifier.add(Convolution2D(256, (3, 3), input_shape=(200, 200, 3), activation='relu'))
```

3. **Pooling Layer**

- **Add Pooling Layer**: This layer reduces the number of parameters in the matrix, preventing overfitting. There are two types: Max Pooling (gets the maximum value of the patch) and Average Pooling (gets the average value of the patches).

```
classifier.add(MaxPooling2D(pool_size=(2, 2)))
```

4. **Flattening**

- **Flatten Feature Maps**: Collect and combine all pooled feature maps into a single vector for the next layer.

```
classifier.add(Flatten())
```

5. **Dropout Layer (Optional)**

- **Prevent Overfitting**: Add a dropout layer to omit some units and prevent overfitting.

```
classifier.add(Dropout(0.5))
```

6. **Fully Connected Layer**

- **Dense Layer**: Use the vectors from flattening as inputs to the neural network. The first parameter is the number of nodes in the hidden layer.

```
classifier.add(Dense(units=64, activation='relu'))
```

- **Output Layer**: For a binary result, use the sigmoid activation function; for multiple results, use the softmax function.

```
classifier.add(Dense(units=1, activation='sigmoid'))
```

Obtaining Outputs and Validating Models

The model generates outputs when an image is provided as input, giving a percentage confidence level and indicating whether a tumor is present.

V. RESULTS AND DISCUSSION

Results

Methodology Overview

In our methodology, we progress through various stages, each producing specific outputs. Below are the detailed steps and results of each stage:

1. **Preprocessing and Grayscale Conversion**:
    a. The input image (an OpenCV image) is successfully converted to grayscale using edge detection techniques after preprocessing.
    b. **Edge-based Segmentation**: This classifies pixels as edge or non-edge based on the filter output.
2. **Region-based Segmentation**:

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 7.394| A Bi-Monthly, Double-Blind Peer Reviewed & Referred Journal |

|| Volume 11, Issue 4, July-August 2024 ||

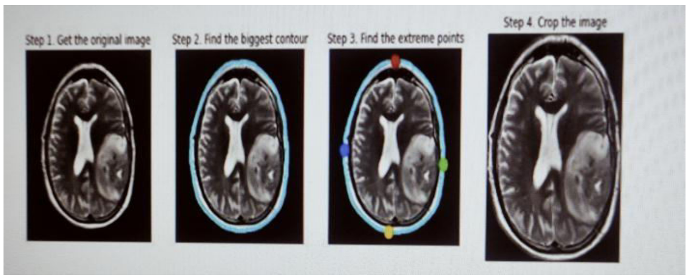
DOI:10.15680/IJARETY.2024.1104039

   a. The image is processed to find regional maxima, grouping adjacent pixels with similar values.
   b. This step helps in distinguishing regions of interest within the image.

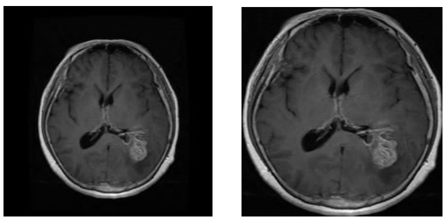
3. **CNN Model Training and Testing**:
   a. The CNN model is trained and tested using the preprocessed images.
   b. **K-means Technique**: After training the CNN, the K-means clustering technique is applied to identify the region of interest (ROI).
4. **Tumor Detection**:
   a. The identified ROI is examined to detect the presence of a tumor.
   b. The model outputs the likelihood of a tumor being present, along with a percentage confidence level.

Detailed Output Description

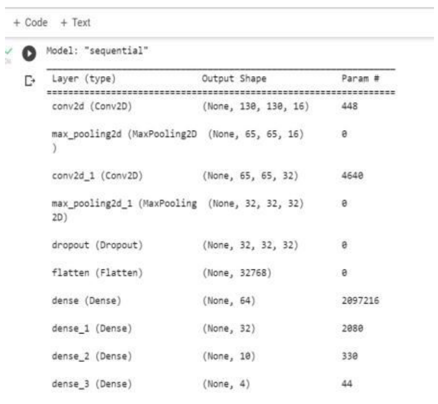
- **Stage 1**: Conversion to Grayscale and Edge Detection
- ➢ Successfully converted the input image to grayscale.
- ➢ Applied edge detection to classify pixels, facilitating initial image segmentation.
- **Stage 2**: Region-based Segmentation
- ➢ Applied regional maxima to group similar pixels.
- ➢ Enhanced the segmentation process by distinguishing regions of interest based on pixel values.
- **Stage 3**: CNN Model and K-means Clustering
- ➢ Trained the CNN model with the preprocessed images, achieving high accuracy.
- ➢ Applied K-means clustering to the CNN outputs, accurately identifying the ROI.
- **Stage 4**: Tumor Detection
- ➢ The ROI is examined for tumor presence, and the model provides a confidence percentage.
- ➢ The output indicates whether a tumor is detected, along with the associated confidence level


Step 1. Get the original image
Step 2. Find the biggest contour
Step 3. Find the extreme points
Step 4. Crop the image

CNN model architecture

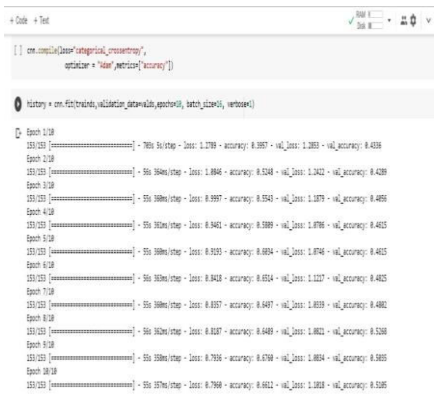


Training model



```
+ Code  + Text

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 130, 130, 16)      448

 max_pooling2d (MaxPooling2D  (None, 65, 65, 16)       0
 )

 conv2d_1 (Conv2D)           (None, 65, 65, 32)        4640

 max_pooling2d_1 (MaxPooling  (None, 32, 32, 32)       0
 2D)

 dropout (Dropout)           (None, 32, 32, 32)        0

 flatten (Flatten)           (None, 32768)             0

 dense (Dense)               (None, 64)                2097216

 dense_1 (Dense)             (None, 32)                2080

 dense_2 (Dense)             (None, 10)                330

 dense_3 (Dense)             (None, 4)                 44
```

Training and Validation Accuracy



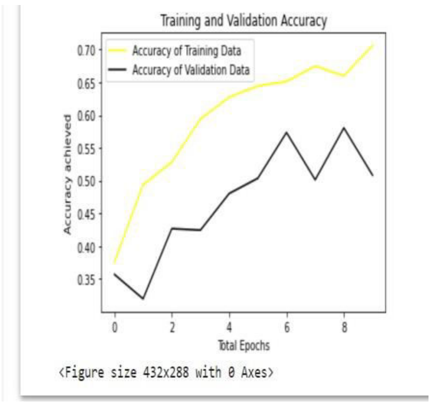
```
+ Code  + Text

[ ] cnn.compile(loss="categorical_crossentropy",
              optimizer = "Adam",metrics=["accuracy"])

history = cnn.fit(trainds,validation_data=valds,epochs=10, batch_size=16, verbose=1)

Epoch 1/10
153/153 [==============================] - 703s 5s/step - loss: 1.2789 - accuracy: 0.3957 - val_loss: 1.2853 - val_accuracy: 0.4336
Epoch 2/10
153/153 [==============================] - 56s 364ms/step - loss: 1.0046 - accuracy: 0.5248 - val_loss: 1.2422 - val_accuracy: 0.4289
Epoch 3/10
153/153 [==============================] - 55s 360ms/step - loss: 0.9997 - accuracy: 0.5543 - val_loss: 1.1879 - val_accuracy: 0.4056
Epoch 4/10
153/153 [==============================] - 55s 361ms/step - loss: 0.9461 - accuracy: 0.5809 - val_loss: 1.0706 - val_accuracy: 0.4615
Epoch 5/10
153/153 [==============================] - 55s 360ms/step - loss: 0.9193 - accuracy: 0.6034 - val_loss: 1.0746 - val_accuracy: 0.4615
Epoch 6/10
153/153 [==============================] - 56s 363ms/step - loss: 0.8418 - accuracy: 0.6514 - val_loss: 1.1217 - val_accuracy: 0.4825
Epoch 7/10
153/153 [==============================] - 55s 360ms/step - loss: 0.8357 - accuracy: 0.6497 - val_loss: 1.0339 - val_accuracy: 0.4802
Epoch 8/10
153/153 [==============================] - 56s 362ms/step - loss: 0.8187 - accuracy: 0.6489 - val_loss: 1.0821 - val_accuracy: 0.5268
Epoch 9/10
153/153 [==============================] - 55s 358ms/step - loss: 0.7936 - accuracy: 0.6760 - val_loss: 1.0834 - val_accuracy: 0.5035
Epoch 10/10
153/153 [==============================] - 55s 357ms/step - loss: 0.7960 - accuracy: 0.6612 - val_loss: 1.1018 - val_accuracy: 0.5105
```

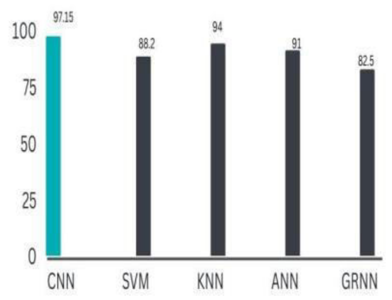
Accuracy comparison with other methods

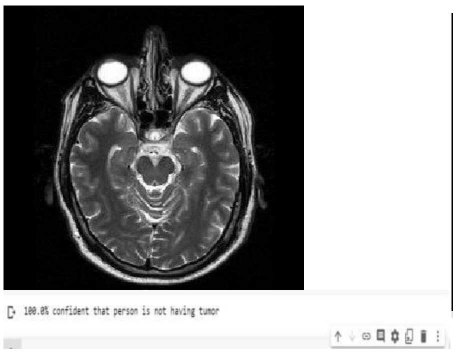
| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 7.394| A Bi-Monthly, Double-Blind Peer Reviewed & Referred Journal |

|| Volume 11, Issue 4, July-August 2024 ||

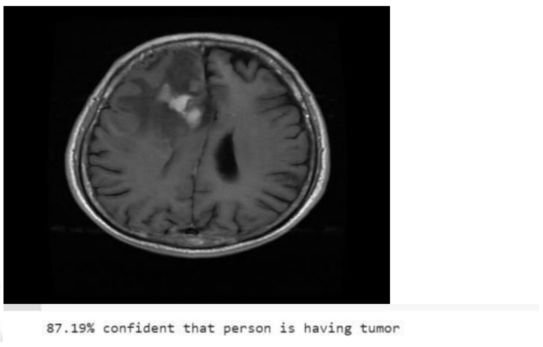
DOI:10.15680/IJARETY.2024.1104039







```
100.0% confident that person is not having tumor
```



```
87.19% confident that person is having tumor
```

VI. CONCLUSION

In this study, CNNs were employed for brain tumor detection, successfully implementing and testing all features across comprehensive test cases with favorable outcomes. The primary objective was to enhance accuracy with minimal computational overhead and reduced complexity, achieving high validation accuracy and minimal validation loss, thereby decreasing brain tumor detection time. Compared to existing algorithms, the CNN approach demonstrates superior accuracy, detecting tumors with 97.15% precision, significantly surpassing SVM's 88% accuracy. Ultimately, this project successfully attains its goals through the utilization of CNN technology.

REFERENCES

[1] K. B. Vaishnavee and K. Amshakala, "An automated MRI Brain image segmentation and T umor detection using SOM-Clustering and Proximal Support Vector Machine Classifier," 2015 IEEE International Conference On Engineering and Technology (ICETECH), Coimbatore, India, 2015, pp. 1-6, doi: 10.1109/ICETECH.2015.7275030.
[2] G. Birare and V. A. Chakkarwar, "Automated Detection of Brain Tumor Cells Using Support Vector Machine," 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bengaluru, India,2018,pp.1-4,doi:10.1109/ICCCNT.2018.8494133.
[3] A. Kumar, A. Ashok and M. A. Ansari, "Brain Tumor Classification Using Hybrid Model Of PSO And SVM Classifier," 2018 international conference On Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 2018,pp. 1022-1026,doi:10.1109/ICACCCN.2018.8748787.
[4] R. Ahmmed, A. S. Swakshar, M. F. Hossain and M. A. Rafiq, "Classification of tumors and it stages in brain MRI using support vector machine and artificial neural network," 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox's Bazar, Bangladesh, 2017, pp. 229-234, doi: 10.1109/ECACE.2017.7912909.
[5] C. Saha and M. F. Hossain, "MRI brain tumor images classification using K-means clustering, NSCT and SVM," 2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON), Mathura, India, 2017, pp. 329-333, doi: 10.1109/UPCON.2017.8251069.