

Automated Intracranial Neoplasm Detection Using Convolutional Neural Networks

Mr. P. Veeresh Kumar¹, N. Geetha Gayathri², G. Harika³, Ch. Sai Poojitha⁴, G. Vyshnavi⁵
T. Lakshmi⁶

¹ Associate Professor Department of Information Technology, KKR & KSR Institute Of Technology And Sciences(A), Guntur, India

^{2,3,4,5,6} Undergraduate Students, Department of Information Technology, KKR & KSR Institute Of Technology And Sciences(A), Guntur, India

Abstract - A brain tumor is nothing but an abnormal collection of cells in the intracranial. The skull that surrounds the brain is very hard, so when a tumor grows inside the brain, it puts pressure on the skull and can cause serious damage. It is regarded as one of the most dangerous diseases in both children and adults. About 11,700 people are diagnosed with brain tumors each year. It accounts for 85–90% of all central nervous system tumors. The 5-year survival rate for malignant brain tumors is approximately 34% for men and 36% for women. Accurate diagnosis, detection of brain tumors, and early initiation of appropriate treatment are essential to prolonging a patient's life expectancy. Brain tumor segmentation is one of the most important and laborious tasks in the field of medical imaging. Manual classification with human assistance can lead to inaccurate predictions and diagnoses. Additionally, this is a daunting task when there is a large amount of data that needs to be supported. Brain tumors are highly variable in appearance, and there are similarities between tumor tissue and normal tissue, thus making the extraction of tumor regions from images difficult. There are different types of imaging tests, such as X-rays, MRIs, and CT scans, used to detect brain tumors. In this study, computed tomography (CT) scan images are used to identify brain tumors. Deep learning (DL) is the latest technology that provides more efficient results in detection and classification. In this paper, a model is developed by detecting tumors using a convolutional neural network. In our study, CNN achieved 97.87% accuracy.

Key Words: Brain Tumor, Magnetic Resonance Image (MRI), Convolutional Neural Networks (CNN), Deep Learning, Tensor Flow, Keras.

1. INTRODUCTION

We use medical imaging technology to get a glimpse inside the human body. The most difficult task is to diagnose and classify tumors or cancers. According to statistics, brain tumors are one of the most important types of cancer in terms of mortality. The International Agency for Research on Cancer (IARC) reports that worldwide, more than one million people are diagnosed with 4,444 brain tumors each year, and the mortality rate is steadily increasing year by

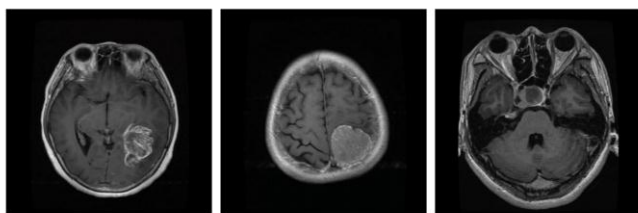
year. Those under the age of 34 To date, doctors have implemented over 4,444 advanced methods to identify more painful tumors in patients. Using CT (computed tomography) and MRI (magnetic resonance imaging) scans, doctors can now dissect abnormalities in different parts of the body. Brain tumors have recently gained momentum due to the increased demand for labour savings and the clinical evaluation of large amounts of medical data. This type of image processing and analysis involves complex calculations of data as well as visualizing and understanding that data. Brain tumors can originate from abnormal, precancerous, cancerous, harmful, harmless, or non-cancerous cells that form in the brain. He divides malignant tumors into two types: those that originate in the brain and subsequent cancerous growths that can spread to different areas of the human body. Under these conditions, cancer metastasizes within the patient's body and is said to have a very low survival rate and a high mortality rate.

1.1 Brain Tumor

The brain is a soft tissue mass that resembles a sponge that is covered in bone. The brain is lined with a delicate layer of tissue, including cerebrospinal fluid. Between the brain and the ventricles, there is cerebrospinal fluid. The cerebrum, cerebellum, and brainstem are the three components that make up the brain. The brain reacts to external stimuli. The cerebellum maintains the body's equilibrium. The brain and spinal cord are linked by the brainstem. A brain tumor is recognized as an overgrowth of the numerous cells that make up the brain. Primary and secondary cells are both present. While secondary cells are spared by metastasis from other parts of the body to the brain, primary cells come from the brain itself. Some metastases spread rapidly. Some metastases spread fairly quickly and exhibit symptoms like seizures, speech difficulties, nausea, and blurred vision. It can be fatal if not treated in a timely manner. This is due to EDA, which is brought on by decreased blood flow and the subsequent deterioration of healthy tissue as a result of the skull's constrained interior and growing intracranial pressure. Electronic healthcare systems and magnetic field information technology aid clinical professionals in giving patients better care. A brain tumor can affect other nearby

structures and is distinguished by its position, image intensity, shape, and size. If the tumor is accompanied by emphysema, the intensity of other nearby areas is also changed. Benign tumor and malignant tumor are the two tumors. Due to their fixed size and slow growth, benign tumors are not regarded as harmful. The tissues around benign tumors are unaffected by them. A birthmark is an illustration of a benign tumor. Cancer can be treated when it is in the premalignant stage, whereas ignorance can cause cancer. Cancer is the name for a cancerous tumor. They begin to grow around nearby tissue after some time has passed since their generation. Malignant tumors can be fatal and grow quickly. For patients, early detection of malignant tumors is very beneficial. To achieve the highest level of accuracy, various techniques have been investigated for the automated detection and classification of malignancies. Furthermore, hospitals are able to achieve extremely low false-negative rates, which raises the demand for computer support. Most observers can identify the presence of specific characteristics that identify malignancy. MRI is frequently used to recognize and highlight specific internal body structures. represents magnetism. imaging using resonance. A strong magnetic field and body radio waves are used in an MRI to produce precise images. Due to its ability to achieve high spatial resolution and better soft-tissue contrast, it is a crucial and useful tool for the accurate diagnosis of disease as well as for the treatment and monitoring of disease using imaging modalities.

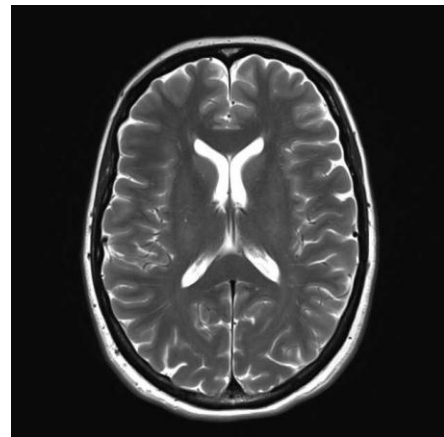
Segmentation of magnetic resonance (MR) images of the brain is key to diagnosing malignancies and other neurological abnormalities. Image segmentation is a preprocessing step that helps divide an image into smaller segments. Images are divided by classes or subsets. This is useful because the segmentation allows us to focus on regions of interest. Segmentation of MRI images can also be performed manually, but it is time-consuming and less accurate.



Glioma

Meningioma

Pituitary



No Tumor

2. LITERATURE REVIEW

[1] K. B. Vaishnav and K. Amshakala proposed an Automated MRI brain image segmentation and tumor detection using SOM-clustering and Proximal Support Vector Machine classifier. Photograph segmentation is a critical approach noticeably used to extract suspicious parts from clinical photographs, which include MRI, CT scan, mammography, and so forth. With this motivation, SOM clustering is proposed in this paper for image segmentation of brain MRI scans. This proposed work proposes effective segmentation and classification using HFS-SOM and PSVM. After segmentation, the resulting image is given as input to his PSVM classifier, followed by feature extraction and selection using GLCM-PCA. Experimental results show that the proposed system has a high accuracy rate and a low error rate. In the future, the system could be improved to support other types of cancer images with a few changes at both the segmentation and classification stages. We need to support a large number of inputs, and we need to improve the accuracy rate. To achieve this, swarm-based feature selection can add more features to improve tumor detection and classification results.

[2] G. Birare and V. A. Chakkarwar developed an Automated Detection of Brain Tumor Cells Using Support Vector Machine. Thousands of people receive a brain tumor diagnosis each year. While it is of significant importance to separate, find, and extract contaminated tumor regions from magnetic resonance imaging for diagnostic evaluation, it is time-consuming and prone to human error. To get around these restrictions, segmentation and classification techniques that are automated and semi-automatic are now being deployed. As a result, we created automatic segmentation using the K-means technique and a brain tumor identification module that uses a support vector machine classifier to identify tumors while also extracting Otsu-threshold characteristics from the output segmented image using a grayscale occurrence matrix. The accuracy of the experimental results was 98.51%. Overall, this approach produces excellent outcomes. Future work on medical

diagnostics may involve the development of additional algorithms, like multiclass SVM.

[3]A. Kumar, A. Ashok and M. A. Ansari introduced Brain Tumor Classification Using Hybrid Model Of PSO And SVM Classifier. Medical imaging is a very important process for the diagnosis of any disease today. MRI is usually used to detect the presence and type of tumor. The process of classifying brain tumors is very complicated. There are various procedures in medical image processing, such as image segmentation, image extraction, and image classification. Different types of features are extracted from segmented data. MRI images, including intensity, shape, and texture-based features. In this paper, machine learning was used to create an online database of MRI images of brain tumors. The particle swarm algorithm was used to create the model optimization (PSO) algorithm for feature selection, and then a support vector machine (SVM) classifier was used. used to classify tumor types on current brain MRI images .Using the PSO-SVM classification model gives 95.23% accuracy when all 14 features are included, compared to 86.82% for the naive SVM classifier. Similarly, the specificity is 94.8% and the sensitivity is 100%. In the future, we hope to apply this concept to bio-inspired algorithms and compare them all to different medical images.

[4]R. Ahmmed, A. S. Swakshar, M. F. Hossain and M. A. Rafiq developed Classification of tumors and it stages in brain MRI using support vector machine and artificial neural network. Cells are the smallest unit of tissue, and their abnormal growth causes tumors in the brain. In this study, support vector machine (SVM) and artificial neural network (ANN)-based classification of tumors and stage in brain MRI images is presented. An integration of temperature-based K-Means and modified fuzzy C-Means (TKFCM) clustering algorithms is used to segment MRI images based on the grayscale intensities of small portions of brain images. The K value of the temperature-based K-Means algorithm is higher than that of conventional K-Means, and the automatically updated membership of FCM eliminates the contour problem of tumor area detection. There is a second method used to classify tumors into benign and four malignant stages using ANN. The accuracy of classification of normal and tumor brains by this proposed method is up to 97.37% with a bit error rate (BER) of 0.0294, which is superior to other methods.

[5]C. Saha and M. F. Hossain designed MRI brain tumor images classification using K-means clustering, NSCT and SVM. Magnetic resonance imaging (MRI) has justified its potential as a novel imaging tool for the anatomical study of brain tissue. Manual classification of a large number of patient MRI brain images is a very time-consuming process and prone to error; therefore, there is a need to develop a fast and accurate automatic scheme for classifying MRI brain images. In this article, we use K-means clustering, the non-

subsampled contourlet transform (NSCT), and support vector machines (SVM) to automatically classify MRI brain images as normal if no tumor is present and abnormal if a tumor is present. We present a scheme for categorizing The preprocessing stage of this proposed scheme uses a median filter to remove noise and improve the resolution of MRI brain images. Then, using K-means clustering, segment the MRI image of the brain. Due to the salient properties of NSCT, such as multi-scale, multi-orientation, and shift invariance, NSCT is applied to segmented images. Seven features are then extracted from the NSCT sub-band coefficients, and these features are applied to a support vector machine for the classification of MRI brain images. The effectiveness of this method is evaluated in terms of sensitivity, specificity, and accuracy, and this method achieves 98.86% classification accuracy on GRB kernel SVM. This scheme has been compared with other classification schemes for MRI brain images and provides satisfactory results, demonstrating the efficacy of our scheme. However, further validation of the proposed hybrid scheme is needed with further testing and training on MRI brain images from different databases and consideration of other performance evaluation parameters to be addressed in future work.

3. 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

3.2.1 Hardware Configuration

- Processor: Intel core i5 or above. 64-bit, quad-core, 2.5 GHz minimum per core .
- Ram: 4 GB or more .
- Hard disk: 10 GB of available space or more.
- Display: Dual XGA (1024 x 768) or higher resolution monitors .
- Operating system: Windows

3.2 Functional Requirements

- Python 3.6.2 or later, PIP, and NumPy 1.13.1 for Windows.
- Pip
- NumPy
- Pandas

- Anaconda
- Jupyter Notebook
- Tensor Flow
- Keras

4. METHODOLOGY

The proposed procedure is divided into various stages and each stage is explained in detail.

- Image input
- Image Preprocessing
- Splitting Data for Training and Testing
- Classifying Brain Tumor Images Using CNNs
- Obtaining Outputs and Validating Models

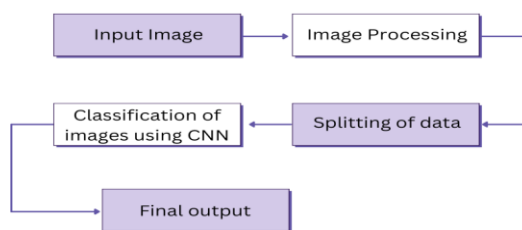


Fig - 1 : Flowchart of proposed method

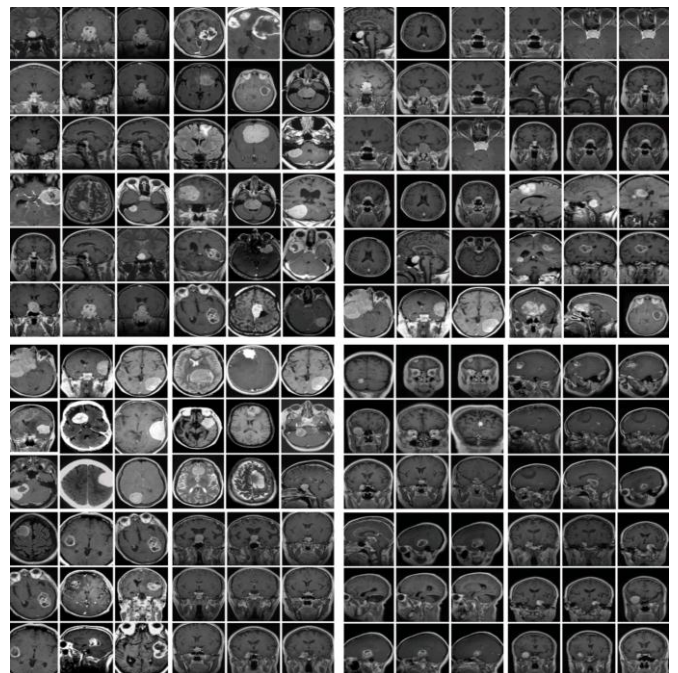
Image Input

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 both datasets contains images with and without tumors. Later 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 chances of getting inaccurate results so we test on testing dataset

The Dataset of brain with tumor:



The Dataset of brain without tumor:

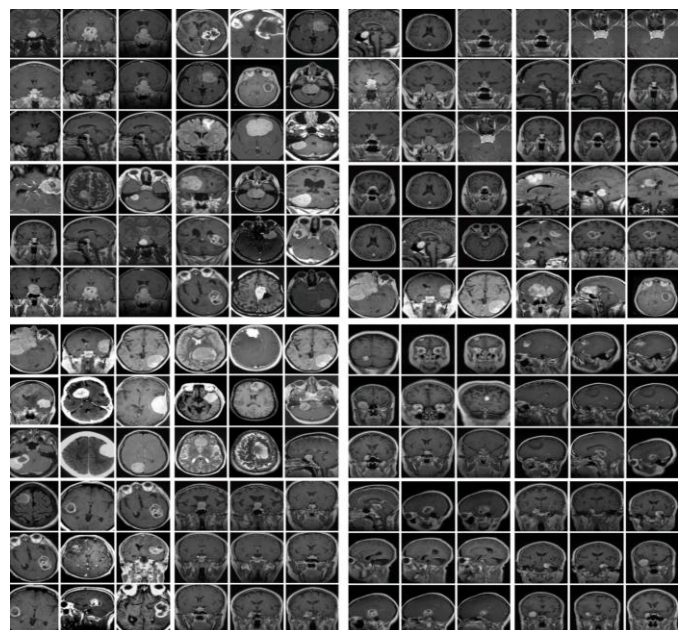


Image Preprocessing

MRI images acquired from a public source cannot be directly supplied for processing. These images contain noise, such as external noise and noise generated by the patient's movement during the MRI scan. For efficient brain tumor detection, they must be removed and images enhanced.

- Grayscale conversion: imported images in jpg format are first converted from RGB models to grayscale images.

- Image resizing: Grayscale images are resized from their current size to a convenient 200 x 200 size for consistency.
- Median Filtering: The resized grayscale image is median filtered to remove existing noise and reduce edge blurring effects.
- High-pass filtering: The median filtered image is high-pass filtered and enhanced by adding a resized image to the resulting image.

Splitting Data for Training and Testing

Testing the model on the same data it was trained on will result in overfitting and poor performance in real-world scenarios. To avoid this, split the data into two parts: a training set and a test set. Normal practice is an 80/20 split.

In this, we split the dataset into training and testing sets using the `train_test_split()` function from the scikit-learn package. The sample size chosen is 0.2 (i.e., 20% for testing and 80% for training). Classification, like any kind of medical image, is the best way to identify an image. All classification algorithms are based on predictions of images in which one or more features are present, and each of these features belongs to one of several classes. An automatic and reliable convolutional neural network (CNN) classification method is used. This makes the structure robust and helps identify details. Convolutional Neural Network (ConvNet or CNN) is a deep learning algorithm that takes in an input image and assigns importance to different aspects or objects in the image to distinguish them from each other. ConvNet requires less preprocessing when compared to other classification algorithms. While the primitive method requires manual development of filters, ConvNet has the ability to learn these filters and properties with sufficient training. ConvNet can successfully capture the spatial and temporal dependencies of images by applying relevant filters. This architecture improves matching with image datasets because the number of relevant parameters is reduced and the weights are reusable. In other words, the network can be trained to better understand image sophistication. The role of the ConvNet is to reduce the image to a manageable form without losing important features necessary to get good predictions. In this step, we need to import Keras and other packages that we will use when building CNN.

Sequential

- To initialize the neural network, we create an object of the Sequential class.
- `classifier = Sequential ()`

Convolution

- To add a convolutional layer, call the Add function on the classifier object and pass the parameters to

Convolution2D. The first argument for a feature detector is the number of feature detectors to create. The second and third parameters are the dimensions of the feature detector matrix. We used 256 feature detectors for the CNN. The next parameter is the input shape, which is the shape of the input image. Images are converted to this format during preprocessing.

- If the image is black and white, it is converted to a 2D array; if the image is colored, it is converted to a 3D array. In this case, we assume that you are working with color. The input shape is a tuple that contains the number of channels (three for colour images) as well as the dimensions of each channel's 2D array.
- If you are not using a GPU, it is recommended to use lower dimensionality to reduce computation time. The last parameter is the activation function. The classification of the problem of images is nonlinear. Therefore, we use a commutation function to ensure that we do not have negative pixel values during computation. This is how the nonlinearity is achieved.

Pooling

Pooling Layer: This is another layer where the matrix starts. This layer is added to reduce the number of parameters in the matrix. This prevents overfitting the model. There are two types of pooling.

MAX_POOLING: Get the maximum value of the patch here.
AVG_POOLING: Here we get the average value of the patches.

Flattening

- All pooled feature maps are collected and combined in this step and placed in a single vector for input to the next layer.
- The flatten function flattens all feature maps into one long column.
- `classifier.add(Flatten())`

Dropout Layer

This is an optional layer that can be added to the model. This layer prevents overfitting by omitting some units.

Fully Connection

- The next step is to use the vectors obtained above as inputs to the neural network using the Keras Dense function. The first parameter is printed. This is the number of nodes in the hidden layer. Through experimentation, we can determine what is the

most appropriate number. The higher the number of dimensions, the more computational resources are required to fit the model. A common method is to choose the number of nodes in powers of two.

- classifier.add(Dense(output=64))
- The next layer we need to add is the output layer. In this case, we expect a binary result, so we use the sigmoid activation function. If you expect more than one result, use the SoftMax function.
- The output here is 1. This is because we only expect the predicted probabilities of the classes.
- classifier.add (high density (output = 1, activation = 'sigmoid'))

Obtaining outputs and validating models

This will give you the output when the image is given as input, along with a percentage of confidence and information about whether a tumor is present or not.

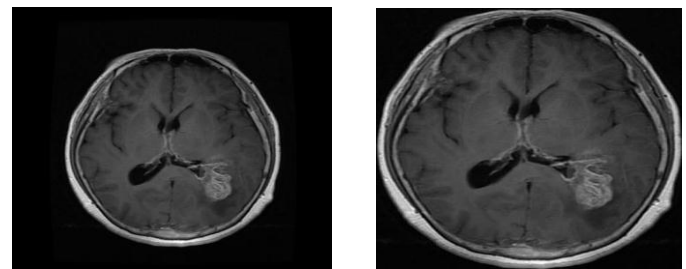
5. RESULTS

In methodology, we are having various stages in this section, and the output of each stage will be described.

In first stage we successfully converted the input image (an OpenCV image) to grayscale using edge detection after applying pre-processing. Edge-based segmentation on an image classifies pixels as edge or non-edge depending on the filter output. In region-based segmentation, the image was subjected to regional maxima in order to group adjacent pixels with similar values. The CNN model was trained and tested before the K-means technique was used to identify the region of interest (ROI), which was then examined for the presence of a tumor.



Fig - 2 : Image Preprocessing



Original Image

Cropped Image

```
+ 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 (MaxPooling2D)	(None, 32, 32, 32)	0
dropout (Dropout)	(None, 32, 32, 32)	0
flatten (Flatten)	(None, 32768)	0
dense (Dense)	(None, 64)	2097216
dense_1 (Dense)	(None, 32)	2880
dense_2 (Dense)	(None, 10)	330
dense_3 (Dense)	(None, 4)	44

Fig - 3: CNN model architecture

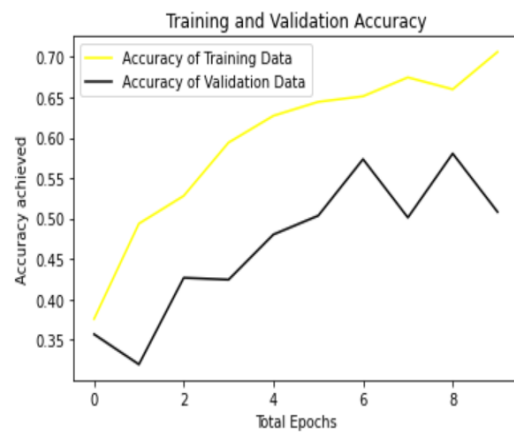
```
+ Code + Text
```

```
[ ] 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 [=====] - 785s 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: 1.0846 - 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: 0.4465
Epoch 4/10
153/153 [=====] - 55s 361ms/step - loss: 0.9461 - accuracy: 0.5809 - val_loss: 1.0786 - 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: 0.4882
Epoch 8/10
153/153 [=====] - 56s 362ms/step - loss: 0.8187 - accuracy: 0.6489 - val_loss: 1.0821 - val_accuracy: 0.5168
Epoch 9/10
153/153 [=====] - 55s 358ms/step - loss: 0.7936 - accuracy: 0.6768 - val_loss: 1.0834 - val_accuracy: 0.5035
Epoch 10/10
153/153 [=====] - 55s 357ms/step - loss: 0.7968 - accuracy: 0.6612 - val_loss: 1.1018 - val_accuracy: 0.5105
```

Fig - 4: Training model



<Figure size 432x288 with 0 Axes>

Fig - 5: Training and Validation Accuracy

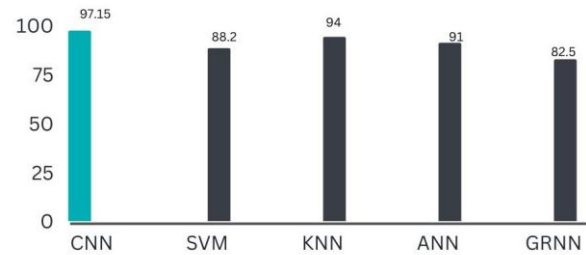


Fig - 8: Accuracy comparison with other methods

6. CONCLUSIONS

In this paper, we used CNNs to detect brain tumors, and we successfully implemented all features and tested it across all possible test cases with satisfactory results. The main goal of this project is to improve the accuracy with minimal computation time and less complexity, to achieve high validation accuracy and very low validation loss, and to reduce the time required to detect brain tumors. Compared to previously proposed algorithms it has more accuracy and detects tumor in less time. It is about 97.15% accurate, whereas SVM is only 88% accurate. Finally, the goal of this project is achieved by using CNN.

7. REFERENCES

[1] K. B. Vaishnav and K. Amshakala, "An automated MRI brain image segmentation and tumor 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. Ahmmad, 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.



Fig - 6: Output with confidence percentage for no tumor



Fig - 7: Output with confidence percentage for image having tumor

[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.