**BRAIN TUMOR PREDICTION THROUGH MRI IMAGES USING CNN IN KERAS**

Shivangi Pokhriyal, Suma N, Upendra Raju A

Department of Electronics

Mount Carmel College, Autonomous, Bangalore, India

Email: [shivangipokhriyal@gmail.com](mailto:shivangipokhriyal@gmail.com)

**Abstract**

In certain fields of medicine, such as surgery and disease treatment, computer vision methods have shown to be extremely effective. Researchers are now experimenting with the identification of various diseases such as lung and kidney cancer. Various medical imaging datasets are freely accessible for researchers today, such as the Cancer Imaging Archive, where we can get free data access to massive databases. A brain tumour is one of the conditions under which a patient's brain produces abnormal cells. This abnormal growth of cells is called tumour, and there are many forms of them. The aim of this work is to create a classification model that takes MRI images of a patient and determines whether or not the patient has a brain tumour. This work uses publicly accessible datasets available on Kaggle by the name ‘Brain MRI Images for Tumour Detection’. It involves the construction of the model, and then its testing, using simple custom layers convolutional neural networks. Some predictions by the model have been observed and results are computed.

**Keywords:** Deep Learning, Keras, Tensor Flow, Brain Tumour, Prediction, Convolutional Neural Networks (CNN), Magnetic Resonance Imaging (MRI)

**Introduction**

The field of deep learning is a subset of machine learning and with the advancement in deep learning techniques clinicians and scientists are being gravitated towards its application in the medical sciences, or medical imaging to be specific. The images of the insides of the human body can be obtained from various imaging techniques like MRI, CT, X Rays and so on. The scientists take great interest in exploring the deep learning algorithms and how they perform when provided with medical image data as input. Conventionally, these medical images have been interpreted by well-trained and experienced medics or doctors, but the medical consultancy turns out to be an expensive resource and not always approachable by all people. In addition to that, the opinion of the medical image data by the experts varies over its subjectivity and complexity. With the advancement in medical sciences, the image acquisition methods have been developed and this points to the fact that image data is being collected at a faster pace than it is being utilised. With ginormous amounts of medical image data and limited human resources in the medicine, the probability for of human error may increase. Henceforth, there is a dire need to address this situation with the help of artificial intelligence and develop a classification model that will help reduce the burden of medical practitioners to some extent.

In this paper, the medical images of the human brain are collected from the MRI scans of patients. The images obtained are the raw data. The question in consideration here is to classify the image data into whether the patient has a tumor or not. In order to address the problem statement, we build a a model using convolutional neural networks (CNNs). The CNN will take the image data as input and will categorize the data into two categories 0 or 1. The output value 0 or 1 indicates the absence or presence of brain tumor respectively. The model is build using the concepts of deep learning and the primary focus while building and the training the model would be to achieve good classification metrics.

The CNNs or convolutional neural networks are the algorithms that comes under deep learning and these algorithms are responsible for working with image data. The CNNs are exemplar for extracting hidden features and representations from the images. Deep learning is a subset of machine learning. This field of study revolves around the models that draw serious inspiration from the structure of human brain. The deep learning models use the analogy from the biological neural network networks and attempts to mimic the brain’s functions of humans. Henceforth, these models are known as artificial neural networks or neural networks. The elementary structure of an artificial neural network is shown in figure 1. We will be referring to the artificial neural networks as neural networks.

Diagram

Description automatically generated

**Figure 1: 2 Layer neural network**

The simplest type of neural network has an input layer, an output layer and a hidden layer and so it can also be referred as the 2-layer neural network. The neural networks make use of these layers to identify or detect the patterns observed in the data. The deep learning technique that is used for this project is CNN or Convolutional Neural Networks. The structure of CNNs discussed in the next sections. CNNs are the incredible algorithms when it comes to image classification. A great way to implement these algorithms is to use the Python library Keras. Keras makes the building of CNNs an easy and hassle-free process. It is an open-source library scripted in the programming language called python. Keras deals with the complexities of the algorithm and provides the neat means to build various models in fewer amounts of time. It enables the user to skip the cumbersome and time-consuming mathematical computations and these with the computations in the backend. Keras is not only very user friendly, but it also offers support for multiple GPUs.

**Methodology**

The methodology adopted for building the classification model will be discussed in this section. Starting with the type of data used and training a classification model to testing it. Reiterating the aim, we are building a convolutional neural network (CNN) model which will take the MRI scans of patients as input and will classify them based on whether they have tumor or not.

1. **Dataset**

The dataset for this object was fetched from Kaggle by the name ‘Brain MRI Images for Brain Tumour Detection’. Kaggle is the data science community which provides open-source datasets to explore and build models in a web-based data-science environment. The dataset was acquired from Kaggle as a zip file, which was then unpacked on the laptop and then uploaded to Google Drive. The purpose to upload it on drive was to ensure that the model can access the dataset directly from the cloud Or google drive in our case. The MRI scans are stored in two files named 'Yes' and 'No' in the datasets. The 'Yes' category contains scans of patients who have a brain tumour, while the 'No' folder contains scans of patients who do not have a brain tumour. There were 253 photos in total in the dataset, with 155 of tumour positive patients and 98 of tumour negative patients. The downloaded dataset contained only images with a size of 240X240 pixels.

1. **Model Classification**

We used collab notebooks, online cloud-based IDE provided by google for python, for this purpose of our research. The code was written, saved, compiled and executed on the google cloud server and hence it can be accessed from anywhere. Initially, we started with enabling the GPU of collab notebooks.

1. The first step started with the commands for importing the libraries required for the implementation.

Text, letter

Description automatically generated

**Figure 2: Libraries to import**

1. Then MRI scans of patients from folders ‘Yes’ and ‘No’ were read and stored in list X and their corresponding labels were stored in list y.
2. While storing them in separates lists, images were also resized from 240X240 pixels to 28X28 pixels. This caused the distortion of MRI images.
3. Now we split the data in to two sets- training set and testing set by calling test\_train\_split, and the train and test values are stored in X\_train and X\_test respectively.
4. The labels were converted into categoricals. The categoricals were transformed into NumPy arrays.
5. After this conversion, we print the shape of train and test data, and it was conformal with previous results as shown in figure 3.

Text

Description automatically generated

**Figure 3: Shape of test and train data**

1. So, we examined and found that 169 images of the size 28X28 pixels were in the training set and remaining 84 images of the size same, i.e., 28X28 pixels were in the testing set (or validation set).
2. After we split this data, we need to define the model network, and introduce layers in our convolutional neural network.
3. The model is built and ‘adam’ was used as an optimizer whereas loss function we used was categorical entropy. We are using accuracy here, as the evaluation metrics for performance evaluation of this model.
4. Then the model was compiled and trained with the training data for 50 iterations of epochs. The training was carried out with training data and the results were compared to the validation dataset.
5. Test data (or validation data) was used to assess the model's performance after it had been trained.
6. Accuracy graphs for training and testing datasets were created. Loss function graphs were also plotted for training and testing data (or validation data).

Chart, line chart, histogram

Description automatically generated

**Figure 4: Model accuracy and model loss for train and test (validation) data**

1. The model's predictions for test data were then verified by comparing them to the actual labels.

A picture containing graphical user interface

Description automatically generated

**Figure 5**

1. Finally, the classification report was printed in order to assess model performance.

Table

Description automatically generated

**Table 1: Classification Report**

1. **Model Evaluation**
2. The model was compiled and then trained for 50 epochs. The result of the training is checked on the validation data. This step completes the model training and proceeds towards the evaluation of model’s performance. The validation accuracy of 86.90% for a loss of 0.5466 is achieved on the validation data.

Text

Description automatically generated

**Figure 6: Model Loss and Accuracy**

Chart, line chart

Description automatically generated

**Figure 7: Model accuracy for train and test (validation) data**

Chart

Description automatically generated

**Figure 8: Model accuracy for train data**

1. The model’s loss versus epochs provided is visualized in figure 9 and it depicts the loss of the model on train and validation datasets with 50 epochs. For both of these datasets, loss is approaching zero.

Chart, line chart, histogram

Description automatically generated

**Figure 9: Model loss for train and test (validation) data**

**Results and Discussion**

At this point, the CNN model is build and the results and performance of the model can be discussed. The dataset downloaded from Kaggle was split into two sets- training set and testing set. The total number of images in the dataset prior to the division were 253. 155 images out of all 253 were of the patients having no brain tumor and rest of the patients having brain tumor. Our model was trained on 169 randomly selected images from the dataset and afterwards it was tested on the other 84 images of the dataset, which were the complete set of fresh images the model hasn’t seen earlier.

1. **Input data**

The dataset for this paper was taken from Kaggle. As it has been already established that the data was the image data obtained from MRI. Originally the data set had dimensions of 240x240 pixel, which were the resized to 28X28 pixels. The resizing affected the image quality and they got distorted which can be seen in figure 10.

A collage of a clock

Description automatically generated with low confidence

**Figure 10: Input Data**

1. **Output**

The output data after passing through the classifier model is shown in figure 11. The output images have two labels, displaying predicted and actual values of the input data. It can be observed that classifier is able to correctly predict and classify majority of the images.

A picture containing text

Description automatically generated

**Figure 11: Output data**

1. **Result**

The efficacy of our CNN model can be described with the help of various metrics. Classification report of the model summarizes those performance metrics. The classification report for the model is given in table 2. On the test set, the model had an accuracy of 86.90 percent (~87 percent). For a smaller number of training dataset, the accuracy of the model was observed to be about 87% for a loss of 0.5466 at 50 epochs; the accuracy can be refined by supplementing more data to the classifier.

Table

Description automatically generated

**Table 2: Classification report**

The support column in the classification report gives the information about the number of classified input occurrences of a given class.

**Conclusion**

The objective of the work was favourably met, and the results were in alignment with this paper’s objective. It presented an algorithm which was successful in differentiating the medical images with tumour. The CNN model was executed successfully. The predicted output of the model was in agreement with the actual results. It was duly acclaimed that our model was able to classify the MRI images of the patients into 0 or 1, that is not having or having tumor. In conclusion the model builds on the foundations of CNNs gave the validation accuracy of 87%. Had we provided it with more of the data of images, there would have been chances of achieving higher accuracy percentage.

**Future Work**

The recommended developments that can be done in future for enhancing the performance of the model includes

* The dataset for training the model can be expanded. Further additional data for training will help in improving and revamping the accuracy of the model and narrowing the loss.
* There can be strategies in which more training data can be obtained, one could be collecting more data from external sources, and another could be applying data augmentation techniques on the existing data.
* The utilization of architectures that are pre-trained, for example Resnet 34 or Vgg16, can also support in realizing the better accuracy in classification of images.

**References**

1. https://analyticsindiamag.com/brain-tumor-prediction-through-mri-images-using-cnn-inkeras/, September 2020
2. Milica M. Badža and Marko C. Barjaktarovi´c, “Classification of Brain Tumors from MRI Images Using a Convolutional Neural Network”, MDPI, March 2020.
3. https://github.com/MuhammadFathy/Study-of-Detection-Brain-Tumor-with-machinelearning
4. Zheshu Jia, Deyun Chen “Brain Tumor Identification and Classification of MRI images using deep learning techniques”, IEEE Access 2017
5. http://www.it.uu.se/edu/course/homepage/projektTDB/ht17/project01/Project01a\_report. pdf
6. <https://innovatemedtec.com/digital-health/medical-imaging>
7. https://www.envrad.com/difference-between-x-ray-ct-scan-and-mri/
8. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3948928/>
9. https://medium.com/@kohlishivam5522/understanding-a-classification-report-for-yourmachine-learning-model-88815e2ce39