Brain Tumor Classification Using Convolution Neural Network

N Saranya¹, D KarthikaRenuka², J Nikkesh kanthan³
¹Assistant Professor- Department of CSE, Sri Krishna College of Engg.& Technology, Coimbatore
²Associate Professor- Department of IT, PSG College of Technology, Coimbatore.
³Department of CSE, Sri Krishna College of Engg. & Technology, Coimbatore
saranyan@skcet.ac.in

Abstract. Brain tumour is a very serious brain cancer. It is present or become due to the separation of the brain cells. In the recent field of this study, tells us that deep learning will help in health industry of medical diseases imaging in the Medical Diagnostic of all the diseases. CNN is mostly used in this Machine learning algorithm. Likewise, in this paper also, we bring out the Convolution Neural Network algorithm, image processing and data augmentation to say the brain images are cancerous and which are not cancerous. This project will require less computational power due to the transfer learning compared to the old CNN model. This has good accuracy results than the old pre-trained models.

Keywords: MRI; CNN; transfer learning; inception; resnet brain tumor; VGG deep learning

1. Introduction

In the past few years because of AI and Deep learning, significant advancement has been made in the medical science like Medical Image processing technique which helps doctors to the diagnose disease early and easily, before that, it was tedious and time-consuming. So to resolve such kind of limitations computer-aided technology is much needed because Medical Field needs efficient and reliable techniques to diagnose life-threatening diseases like cancer, which is the leading cause of mortality globally for patients [1-5].

So in our study with the help of Brain MRI Images as shown in Figure 1, we provide a method for classification of brain tumors into cancerous and non-cancerous using data augmentation technique and convolutional neural network.
1.1 MRI IMAGE OF TUMOR

![MRI images of tumour]

Figure 1. MRI image of tumour

1.2 BLOCK DIAGRAM

![Block diagram]

Figure 2. Block diagram

1.3 METHODOLOGY

Data Set Description

The data, Brain MRI Images for Brain Tumor Detection, was done from Kaggle repository. It is classified into two scan types: If there is a brain tumor—YES; If there isn’t a brain tumor—NO.
2. Data Pre-processing
The dataset considered has very limited data. For a classification model to perform with high accuracy, it needs to be trained on a humongous amount of data. To provide data at such a scale, Imagedatagenrator is being used from the keras library. New types of data will be generated from the existing images, by changing the factors like shear range, scale, zoom range, and flip orientation. It generates new data from the existing images, by altering the factors such as scale, shear range, zoom range, and flip orientation[6-10]. Figure 2 shows the block diagram.

Architecture
The proposed convolutional neural network model performs the following 4 operations:

- Convolution operation
- Max pooling
- Flattening
- Full Connection

2.1 Convolution Operation: To reduce the size of the image, Convolution operation creates a feature map, that further helps in easy and faster processing on the image as shown in Figure 3. Every image has its own features so to detect the particular feature of an image convolution operation detects features of the image using feature detector [11].

2.2 Max Pooling: A complicated image will be large in size so it has to be reduced, while retaining the important features or patterns as shown in Figure 4. Max Pooling is one such technique, where the features with highest normalized value are preserved [12].

2.3 Flattening: For the features of the image to be fed into a neural network, they must be represented in the form of a feature vector as shown in Figure 5. Flattening is the operation which performs this transformation of 2D feature matrix into a vector [13]
2.4 Full-Connection: A full connection establishes a link between the flattened features and the actual neural network as shown in Figure 6 [14]

3. MACHINE LEARNING

Machine learning as a field of research is somewhat hard to elucidate succinctly, because of the breadth of the world, the number of ongoing researches, the interdisciplinary aspects of the world and a multitude of other factors. For the requirements of this report the next definition are getting to be used as a start line, because it's sufficiently precise: "A bug is claimed to seek out from this point mathematically the matter are formulated, applied to the machine learning of the algorithms to hunt for a function which is not known from the domain ,to the input and also a co-domain is known, that of the output. In other terms, this program should hunt out maths connection with the data and therefore, the input target. This connection will depend on the task at hand. The similar example of the already mentioned classes of the tasks for is to hunt the connection between the representation (vectorized) of the input to the consisting gaggle of indiscrete categories. Sample of these tasks contain the image classification for developing a program and object detection, finding out the faces in the images and finding the bound box for them. Other tasks are different like credit fraud spotting, MT, and example and translation software. The implementation of the machine learning algorithms performance will either does with accuracy in this implementation. The percentage category classified correctly calculated into ratio to properly be classified into 5 different examples. The hard part is that isn’t such tons the measured choice rather than the aspects of the output will be needed to measure in the primary area. The other complications is due to the practical insecurities chosen quality will be hard to be measured so the measured should be chosen in these cases. Supervised learning and unsupervised learning are the basic experience or learning or machine learning algorithm. In supervised learning, the goal is to find out the label, the value associated. That's, programs that perform also or almost also on unobserved inputs as those observed during training, the foremost thing that separates machine learning from optimization is that the goal consists of minimizing both the training error and thus the test error. The aforementioned
Datasets are split in two predetermined sets to be utilized within the training process, one for training and one for testing [15].

In attempting to understand the aforementioned goal two more challenges appear; underfitting and overfitting on the training set. Underfitting occurs when the model cannot reach a sufficiently low training error and overfitting occurs when the gap between the training error and thus the test error is simply too large. The previous problem is caused by a model with low representational capacity that can't fit the training set properly, the latter are often caused by a model with high capacity that memorizes properties from the training set too well to perform sufficiently on the test set. Overfitting can also be caused by a training set that's too small to generalize properly from. Underfitting are often fixed rather easily with a model with sufficient capacity, but overfitting are often harder to repair because of the actual fact that acquiring greater amounts of data isn't feasible. A prominent example of how which can be introduced during the training is regularization, which is used to limit the space of potential functions since the goal is to hunt out the only one. There's shortly more room to travel into detail here, and thus the issues which can occur and their corresponding solutions are getting to be discussed in later sections.

The parameters of the machine learning algorithm that are not adapted or changed by the algorithm itself during training are called hyperparameters, parameters which can be tuned to vary the behaviour of the training algorithm. It's fitting to introduce an additional partition of the dataset here, that of the primary training data into a training set and a validation set. The training set is used purely for adjusting the inside parameters, e.g. weights and biases during the training and thus the validation set is used to measure this generalization error and to manage the hyperparameters accordingly. The validation set isn't used to tune the inside parameters. The important distinction between the validation set and thus the test set is that the latter isn't used during the training process within the least and doesn't give any input to the training process - it's simply used to measure performance. As a final observation during this chapter; there are plenty of machine learning algorithms to choose from, far too many to elucidate intimately here or even to supply a cursory oversight.

4. DEEP LEARNING

Deep Learning and Neural Networks Although the very initiative towards neural networks was taken in 1943 with the work of Warren McCulloch and Walter Pitts, the first application using artificial neurons came with Frank Rosenblatt’s invention, the perceptron. A perceptron is that the only possible version of a man-made neuron, and it is the essential attributes as follows: One or more numeric inputs with corresponding weights, positive or negative, for every input. A bias which may be either positive or negative. Can informally be described because the neurons resistance to “firing off”. An activation function (in the case of the perceptron, the unit step function). One output value, the activation function applied to the sum of the weighted inputs and thus the bias. More informally stated the perceptron outputs 1 if the sum of the weighted inputs and thus the bias is bigger than 0, and 0 if not. Albeit perceptron’s aren’t utilized in practice they led to subsequent logical step, the multilayer perceptron (MLP) or the feedforward neural network. A feedforward neural network is simply artificial neurons in layers, with all the outputs from each neuron within the preceding layer fed forward, not backwards, into each neuron within the subsequent layer, the exceptions being the input layer (consisting of passive neurons that do not transform the input) and thus the output layer. The layers between the first and last are called the hidden layers, which provides the network depth and thus leads to the first a fully connected 9 layers.

All neurons within the network have a singular set of neighbourhoods of the name of this chapter, deep learning, which is additionally a typical name for the use of deep neural networks as a whole and associated techniques. Since each hidden layer, and thus the output layer, contains neurons who
individually are connected with the output from each neuron within the previous layer, those layers during a feedforward network are called weights and thus the activation functions are non-linear functions. That latter part is getting to be expounded upon later within the chapter. a man-made neuron.

Author: Michael Nielsen Source: http://neuralnetworksanddeeplearning.com/chap1.html#perceptrons License: cbn

Before the more technical details of feedforward neural networks are addressed, a more fundamental property of feedforward neural networks must be introduced first; that feedforward neural networks work as universal function approximators. An single perceptron or the opposite artificial neuron isn't of much use, but a network with a minimum of 1 hidden layer can approximate any continuous function, which in practice means any discontinuous function are often approximated also. an overview of the formal proof is outside of the scope of this report, but conceptually a man-made neuron are often compared to a NAND or NOR gate therein it works as an universal building block. the foremost difference is that a man-made neuron has parameters which can be tuned and should therefore be trained. While speaking of parameters it's fitting to introduce the activation functions that are becoming utilized in practice. The unit step function mentioned above isn't utilized in practice because of the actual fact that a little change in input can cause a huge change in output (0 to 1), an unwanted property since endless change within the output is preferable. The sigmoid function and thus the hyperbolic tangent function, smoother versions of the unit step function, are utilized in practice but the activation function of choice today is that the rectified long measure function (ReLU), which is defined by returning only positive values, and variants of that function. The output layer, or the classification layer, uses the softmax function, which outputs a probability distribution over all the categories. More informally the softmax function outputs the foremost likely category, the classification that the network found most probable. to teach a neural network we'd like another quite measure of how big this error is outside of the training error, some quite measure of how rich the weights and biases within the network are as whole. to unravel this a price or objective function is introduced, a function that measures the whole current error. the two important properties such an objective function must have is that it's non-negative for all inputs, which it's zero or on the brink of zero if the error is small. The direct goal of the training is thus to attenuate the target function.

5. Conclusion

The objective of this work is to uprise a model for classifying the brain tumor mri images. hence, convolution neural network supported classification is used. This concept is able to detect the images using Keras, by building an artificial convolutional neural network. Pre-processing image is done by this is splitting segmenting and extracting the brain tumor from MR images. In this document, a new approach was presented to classify brain tumors. First, using the image edge detection technique, we find the region of interest in MRI images and cropped them then, we used the data augmentation technique for increasing the size of our training data. Second, we provide an efficient methodology for brain tumor classification by proposing a simple CNN network. For sophisticated and accurate results neural network requires a large amount of data to train on, but our experimental result shows that even on such a small dataset, we can attain full accuracy and our accuracy rate is very fine as compared to VGG-16, ResNet-50, and Inception-v3 model. Our model average training time per epoch is 205 sec while the VGG-16 takes 456 sec, ResNet-50 takes 606 sec and Inception-v3 takes 375 sec average training time per epoch. So, our model needs less computational specifications as it takes less execution time. Moreover, our model and Inception-v3. Our proposed system can play a prognostic significance in the detection of tumors in brain tumor patients. Our proposed system is for binary classification problems, however, in future work, the proposed method can be extended for categorical classification problems such as identification of brain tumor types such as Glioma, Meningioma, and Pituitary or may
be used to detect other brain abnormalities. Also, our proposed system can play an effective role in the early diagnosis of dangerous disease in other clinical domains related to medical imaging, particularly lung cancer and breast cancer whose mortality rate is very high globally. We can prolong this approach in other scientific areas as well where there is a problem in the availability of large data or we can use the different transfer learning methods with the same proposed technique.

The major drawback will be the computational time while working with larger data. However, working on a larger data set may improve the accuracy of the training model. As an extension to this work, the model can be modified to become compatible with 3 dimensional brain scans, in order to perform more efficient image segmentation and also to identify the stage of the tumor.

References

[1] Mayfield Certified Health Info, https://mayfieldclinic.com/pe-brain tumor.htm abann American Association of Neurological Surgeons, https://www.aans.org/en/ Patients/ Neuro surgical Conditions-and Treatments/ Brain Tumors
[2] Md. Rezwanul Islam , Md. Rizwan Imtiaz, Marium-E-Jannat, Detection and analysis of brain tumor from MRI by Integrated Thresholding and Morphological Process with Histogram based method , IEEE 2018
[3] Harshini Badisa, Madhavi Polireddy, Aslam Mohammed, CNN Based Brain Tumor Detection , ISSN: 2249 – 8958, 8 (4), April 2019
[4] Bashayer Fouad Marghalani, Muhammad Arif, Automatic Classification of Brain Tumor and Alzheimer's Disease in MRI ,2019
[5] J. Seetha and S. Selvakumar Raja, Brain Tumor Classification Using Convolutional Neural Networks, 11(3), p. 1457-1461, 2018
[6] Selvaraj Damodharan and Dhanasekaran Raghavan, Combining Tissue Segmentation and Neural Network for Brain Tumor Detection, 12(1), 2015
[7] Sourabh Hanwat, Chandra J, Convolutional Neural Network for Brain Tumor Analysis Using MRI Images, 2019
[8] Tonmoy Hossain, Fairuz Shadmani Shishir, Mohsena Ashraf, MD Abdullah Al Nasim, Faisal Muhammad Shah, Brain Tumor Detection Using Convolutional Neural Network, ICASERT 2019
[9] Fatih Özyurt, Eser Sert, Engin Avci, Esin Dogantekin, Brain tumor detection based on Convolutional Neural Network with neutrosophic expert maximum fuzzy sure entropy ,2019
[10] Burhan ERGEN, Zafer COMERT, BrainMRNet: Brain Tumor Detection using Magnetic Resonance Images with a Novel Convolutional Neural Network Model, 2019
[11] M. Suganya and H. Anandakumar, Handover based spectrum allocation in cognitive radio networks, 2013 International Conference on Green Computing, Communication and Conservation of Energy (ICGCE), Dec. 2013. doi:10.1109/icgce.2013.6823431. doi:10.4018/978-1-5225-5246-8.ch012
[12] Haldorai and A. Ramu, An Intelligent-Based Wavelet Classifier for Accurate Prediction of Breast Cancer, Intelligent Multidimensional Data and Image Processing, pp. 306–319 Moi Hoon Yap, Gerard Pons, Joan Martí, Sergi Ganau, Melcior Sentís, Reyer Zwiggelaar, Adrian K. Davison, Robert Mart, Automated Breast Ultrasound Lesions Detection Using Convolutional Neural Network, 2018 © 2010 Kaggle Inc, https://www.kaggle.com/
[13] Rachapudi, V., Lavanya Devi, G. Improved convolutional neural network based histopathological image classification. Evol. Intel. (2020). https://doi.org/10.1007/s12065-02000367-y
[14] Krishnaraj, V., Jaisharma, K., & Deepa, N. (2019). Real time brain tumor detection using deep reinforcement learning algorithm. Test Engineering and Management, 81(11-12), 5370-5379. Retrieved from www.scopus.com.