

AiCNNs (Artificially-integrated Convolutional Neural Networks) for Brain Tumor Prediction

Ansh Mittal¹, Deepika Kumar^{2,*}

^{1,2}Bharati Vidyapeeth's College of Engineering, New Delhi, India

Abstract

INTRODUCTION: Accurate analysis of brain MRI images is vital for diagnosing brain tumor in its nascent stages. Automated classification of brain tumor is an important step for accurate diagnosis.

OBJECTIVES: This paper propose a model named Artificially-integrated Convolutional Neural Networks (AiCNNs) that accurately classifies brain MRI scans to 3 classes of brain tumor and negative diagnosis results.

METHODS: AiCNNs model integrates 5 already trained models including simple convolutional neural networks (one uses a simple CNN while the other utilizes data augmentation) and three pre-trained networks whose weights are transferred from ImageNet dataset.

RESULTS: AiCNNs model was trained on 3501 augmented T1-weighted contrast enhanced MRI (CE-MRI) brain images. Validation results of 99.49% (loss=0.0303) had been achieved by AiCNNs on a set of 1167 images, which outperform its contemporaries which have got results upto 97.81% (loss=0.1794) and 97.79% (loss=0.1787).

CONCLUSION: AiCNNs has been shown to obtained a test accuracy of 98.89 % on a set of 1167 images.

Keywords: MRI, Machine Learning, Deep Learning, AiCNNs, CNN, Data Augmentation, ImageNet

Received on 29 November 2018, accepted on 22 January 2019, published on 12 February 2019

Copyright © 2019 Ansh Mittal *et al.*, licensed to EAI. This is an open access article distributed under the terms of the Creative Commons Attribution licence (<http://creativecommons.org/licenses/by/3.0/>), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi: 10.4108/eai.12-2-2019.161976

1. Introduction

The brain is the central nervous system of the body, and is responsible for all activities originating in, or from the body. When abnormal cells start forming within the brain, it leads to the condition known as a Brain Tumor. A tumor in this organ can be extremely threatening to anyone's life. Among these, there can be of two types either cancerous or non-cancerous (i.e. Malignant or Benign) [1]. Brain Tumors can further be divided into primary and secondary tumors. The primary brain tumors start within the brain while secondary brain tumors spread to the brain from somewhere else. The secondary brain tumors are one of the most complicated neurological cancers [2]. The symptoms for brain tumor may range from severe headaches and seizures to problems with vision and mental changes, depending on different parts of the body [3]. Nowadays, diagnosing brain tumors just on the basis of the given symptoms has become an arduous task. Even with the advent of CT (i.e. computed

tomography) and MRI (i.e. magnetic resonance imaging) the amount of data that has to be analysed for detection, has increased. So, many mathematical and image processing techniques have been proposed to counter the aforementioned problem. These algorithms work mostly on T1, T1-weighted, T2, FLAIR MRI scans and CT scans and may not be that accurate to detect or classify brain tumors [4]. An image of a brain tumor in a T1-weighted CE-MRI brain image has been depicted in Figure 1.

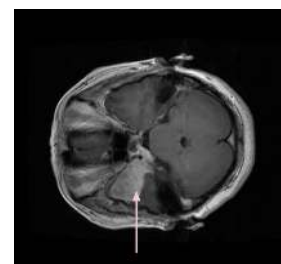


Figure 1. T1 Contrast-Enhanced MRI (CE-MRI) image depicting meningioma brain tumor (marked by arrow).

*Corresponding author. Email: Deepika.kumar@bharatividyaapeeth.edu

The discipline of Machine Learning has been proliferating in several domains while getting optimum results. These include disciplines such as Economics (for crisis detection [5]), Zoology (for animal migration) [6], psychology and psychiatry [7-8], Astronomy (for orbit transfers) [9], and even for Disaster Management (for robot that monitors and prevents disasters) [10]. Some other contributions of Machine learning also include handwritten character recognition [11], predicting chemical reactions [12], and stock price prediction [13]. Apart from this, Machine Learning has also been able to obtain significant results in domain of medicine. The discipline of Machine Learning has seen a rise in fields such as breast cancer, diabetes, heart cancer, skin cancer, and even psychological disorders [14-23]. There were attempts to create a framework for healthcare using Big Data and Machine Learning techniques [24]. The need for brain tumor classification as mentioned above hence led to many advances in mathematics and machine learning techniques being employed involving brain tumor area retrieval from images [25] and classification of brain tumors [26-28]. Hence, many other ANN models and transfer learned CN models have been proposed for the discipline of brain tumor classification such as [29- 31].

This research's contribution has been four-fold which have been summarized as follows.

1. A dataset having T1-weighted contrast-enhanced MRI (CEI-MRI) images of 3 types (i.e. Glioma, Meningioma, and Pituitary) of brain tumors samples which were previously MatLab data files, has been prepared as a set of image files (.jpg) while also augmenting some Negative brain tumor results.
2. A comparative analysis of machine learning algorithms has been conducted in Table 1. This analysis tells us that among the commonly used classifiers, the Random Forest classifier which is an ensemble of decision trees provides one of the best accuracies.
3. A brief comparison of 8 deep learning models (of which some are commonly used) has been done in Table 2 Section 3. Some trends have been analysed in the text ensuing the table.
4. AiCNNs model has been trained using 5 pre-trained models whose weights were loaded. It is a model that is formed due to integrated level-0 sub-models that are passed through a level-1 meta-model which is essentially an Artificial Neural Network. AiCNNs has been made robust to overfitting due to one model which uses data augmentation.

The rest of the paper is organized as follows. In section 2, the works related to the AiCNNs model have been studied. These are then used for comparison analysis in section 3 which compares the performance of state-of-the-art algorithms and most conventional algorithms after extracting global features from the image dataset. Section 4 delineates the proposed AiCNNs model in detail along with

the dataset used and augmentations done. Section 5 enlists the results that were obtained when the AiCNNs model had been trained and these results were compared in a table to some common deep learning models. Finally, the research is concluded by stating some inferences derived from Table 3 about the AiCNNs model and outlines a possible future scope.

2. Related Work

The earliest works in MRI image evaluation, processing, and classification date back to November 1999 [32]. This research essentially made it possible for MRI simulators to efficiently create 3D brain images. MRI scans had been used in medical imaging for a while then, from their advent in 1971 by Paul Lauterbur [33]. Research like [34] provided a direction of machine learning algorithms in the discipline of medicine. It briefly discussed the significance of decision trees, neural networks, and Naive Bayes classifier and hence became a trailblazer in the aforementioned domains. Later in 2003, Fatemzadeh proposed MGNG (modified neural-gas) algorithm for automated landmark extraction. This was a neural network-based unsupervised algorithm that split and then merged SOMs (Self Organizing Maps) [35] which were used for CT-scans. In 2006, Shang used Principal Component Analysis (PCA) and neural networks to register several computed tomography (CT) and magnetic resonance (MR) brain images automatically. This was done using the first finding principal directions using both the algorithms aforementioned.

In 2009, revolutionary research [36] shaped the application of machine learning in the discipline of Brain Tumor. It used pattern classification to differentiate between gliomas and metastases while determining the stages of gliomas in Brain MRI scans. It used many steps for classification such as region-of-interest (ROI) (which is essentially the ratio of the area of the intersection to the area of the union), feature extraction and feature selection. The classification was done using support vector machines in the support of 102 images. The results for accuracy were found out to be 85 % for binary classification of metastasis and gliomas. Here, only 4 images of meningiomas were found. But this model performed badly as has been discussed later. The research aimed at the classification of meningioma tumors was conducted in 2010. It used optimum channel amongst RGB color channels for histopathological images for optimum texture combinations. The texture features were extracted using 4 feature extractors amongst which 2 were model-based and 2 were statistics-based. These features had then been fused together in different combinations while excluding the correlated features to reduce redundancy and in turn avoiding the model from overfitting. Then a Bayesian classifier classified the images if they represented meningiomas tumor or not. Finally, a model-based Gaussian Markov field and a statistics-based run-length matrix texture were used to obtain an accuracy of 92.50 % [37]. Benign and Malignant Brain tumors or normal and abnormal images had been classified by El-

Dahshan et al [38]. Although they used just 101 images, they had been able to achieve a train and test accuracy of 99% on this set. Their approach utilized feedback pulse-coupled neural networks, discrete wavelet transforms, principal component analysis, and feed-forward back-propagation neural network for image segmentation, feature extraction, minimizing the dimensions of coefficients of wavelets and final classification, respectively. Later, SVM based approach for medical diagnosis and automated tumor detection was introduced [39]. This approach had been developed, keeping in mind that large volumes of MR images that couldn't be inspected by manual intervention. An anisotropic filter had been used here as a means of pre-processing. The connected component pixels and edge segmentation had been used together for image segmentation tasks. Further, feature extraction was conducted using global histogram features while SVR algorithm analysed and extracted these features to give ~95 % accuracy whereas feedforward backpropagation neural network (BPNN), gradient descent (GD) and levenberg-marquardt (LM) got an accuracy of ~93.15 % on average, 92.21 %, and 94.14 %, respectively. In 2015, Yan Xu et al [40] proposed a deep convolutional activation feature (CNNs) model for classification and segmentation for MICCAI 2014 brain tumor digital pathology challenge. The proposed model transferred the features from the ImageNet dataset to get 97.5 % classification accuracy and 84 % segmentation accuracy. Later, Jun Cheng et al [41] performed brain tumor classification using ROI (as described in [36]) on 3 classes (i.e. meningioma, glioma, and pituitary brain tumors). This was performed on the dataset that this model is trained on. They initiated their work by augmenting the region of the tumor with image dilation which was used as ROI instead of the original tumor region. They then split the tumor region into concentric ring-form sub-regions. Three feature extraction methods had been used - intensity histogram, bag-of-words (BoW) and gray level co-occurrence matrix which gave them an accuracy of 87.54 %, 89.72 %, and 91.28 %. These work will hold vital significance to our research as would be seen in section 3 and later on when the results would be discussed.

Authors proposed deep convolutional neural networks with significant dropouts, leaky rectified linear units, and small kernels for convolutions for the 2015 Multimodal Brain Tumor Segmentation (BraTS) [42]. They aimed at distinguishing LGG (Low-Grade Gliomas) against HGG (High-Grade Gliomas). In [43], Jocelyn Barker et al. introduced a method to overcome inefficiencies in computerized analysis of whole slide pathology and mislead diagnosis algorithms (happening due to diverse tissue regions in whole slide). This has been done using analysing regions that were coarse in slide images, then extracting localized features of the tiled region of the slide images, and then reducing their dimensions, they achieved an accuracy of 93.1% on binary classification among gliomas and glioblastoma multiforme. Later, Liya Zhao [44] introduced a multiscale CNN for the BraTS challenge that was designed to combine information from different sizes of the region

around the ROI. This model had also combined features from T1, T1- enhanced, T2 and FLAIR MRI images and it achieved an accuracy of about 90 % or 0.9. The research done in [45-49] should be referred for a detailed analysis of other segmentation-related research which utilized deep learning models. Kaur B et al. proposed a Beaming Edge SAlient (BE-SAL) approach for the segmentation of images [50]. In 2017, R. Anjali and S. Priya [51] proposed a model having different stages such as image pre-processing (or noise removal), enhancing texture features, feature segmentation, feature selection, and classification. They utilized a hybrid of CART (Classification and Regression Trees) and an ensemble of SVM (Support Vector Machine) for classifying if an MRI scan has a tumor or not. This hybrid achieved an accuracy of 92.31 %. Later, Ali Ari and Davut Hanbay [52] proposed a model having 3 stages. The first stage cleaned the data of noise, while, in the second stage cranial MRI were classified using extreme learning machine local receptive fields (ELM-LRF), as benign and malignant. Later, segmentation of the tumors was done.

3. Methodology

Various Machine learning algorithms have been used for Brain tumor classification as had been studied in [53] (that used Random Forest Approach) which gave an accuracy of 79.67 %, [54] (which used Support Vector Classification) which resulted in an accuracy of 94 %, and [55] (which used Naïve Bayes) that had a detection rate of upto 90.63 %. These algorithms were tested on a different dataset which consisted of only 50 brain MRI scans. A comparative analysis of all the existing machine learning algorithms has been done in this section for brain tumor classification for the dataset that we used. It has been done for comparing accuracies that utilized dataset combined using 2 different datasets. The first dataset used here is a combination of three types of tumors, namely Glioma, Meningioma, and Pituitary [56]. A fourth class depicting negative results has also been augmented to the dataset from [57]. Figure 2 represent a sample of 5 instances from each class discussed above.

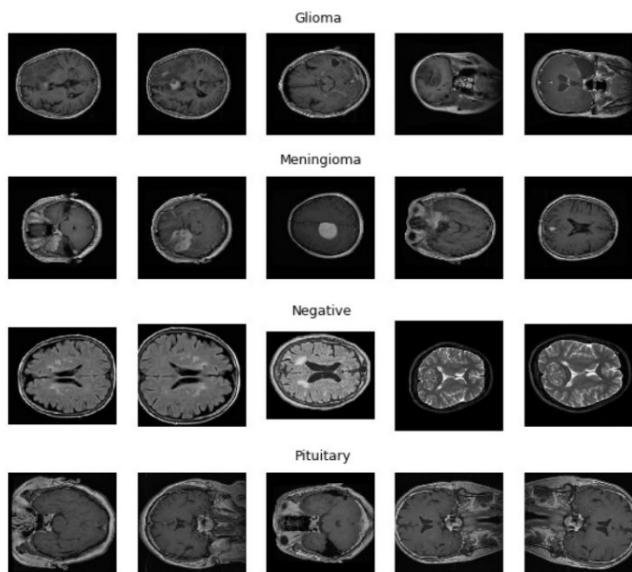


Figure 2. Sample of T1 Contrast-Enhanced MRI (CE-MRI) class images.

Then Global features from datasets such as Moments, Haralick Texture Features and color histograms have been extracted from it. A comparative analysis of various machine learning algorithms like Logistic Regression, LDA, K-nearest neighbours, Classification and Regression Trees (CART), Random Forest, Gaussian Naive Bayes and Support Vector Machine (SVM) have been done and results for the same have been depicted in Table 1. The aim of doing this had been to determine which algorithm performed well on the aforementioned dataset.

Table 1. Comparative Analysis of various machine learning algorithms

ALGORITHMS USED	MEAN ACCURACY ACHIEVED
LOGISTIC REGRESSION	63.56 % (± 1.41 %)
LDA	73.84 % (± 1.02 %)
K-NEAREST NEIGHBOURS	61.23 % (± 1.80 %)
CLASSIFICATION & REGRESSION TREES	78.41 % (± 1 %)
RANDOM FOREST CLASSIFIER	85.31 % (± 0.77 %)
GAUSSIAN NAÏVE BAYES	57.20 % (± 1.93 %)
SUPPORT VECTOR CLASSIFICATION	48.91 % (± 1.66 %)

A further extrapolation of Table 1 has been done in figure 3 given below. It shows that the Random Forest Classifier has achieved maximum accuracy of up to ~86.08%. On the other hand, SVC (or Support Vector Classification) got a maximum accuracy of 50.57 %, which has just been a little

over a simple random classifier which may give an accuracy of 50 %. Figure 3 represents a comparative analysis of algorithms using the Box Plot which depicts the maximum and the minimum accuracy that they had achieved.

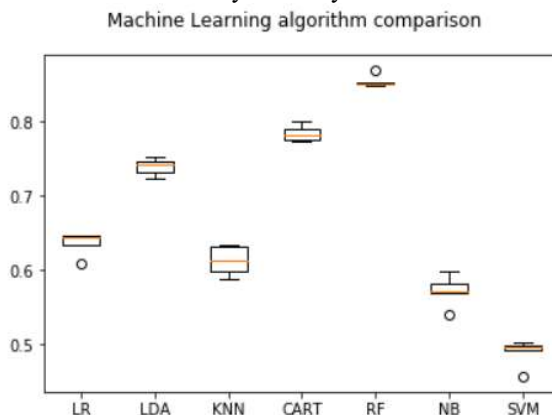


Figure 3. Comparative Analysis (using Boxplot) of different machine learning algorithms for classification.

This analysis leads to the conclusion that instead of using global features, a better classification can be done using the pixel values of MRI scans of the brain. Hence, deep learning-based approach utilizing data augmentation and simple convolutions had been used to classify tumors. The algorithms comprise of CNNs, different transfer learning methods like VGG16, VGG19, Xception, VGG19 (with ELU). The results for the same are shown in Table 2.

Table 2. Comparative Analysis of different CNN models.

MODEL NAME	CNN ALGORITHMS	DA USED	TRAIN (acc _{train})	VALIDATION (acc _{validation})	TEST (acc _{test})
MODEL (1)	SIMPLE CNN	×	97.00 %	94.00 %	92.63 %
MODEL (2)	SIMPLE CNN	✓	90.02 %	94.09 %	94.60 %
MODEL (3)	VGG16	✓	86.62 %	91.95 %	89.97 %
MODEL (4)	VGG19	✓	81.74 %	91.26 %	88.95 %
MODEL (5)	XCEPTION	×	99.03 %	68.55 %	66.32 %
MODEL (6)	VGG19 (WITH ELU)	×	97.29 %	94.26 %	97.43 %
MODEL (7)	VGG16	×	99.52 %	97.79 %	97.43 %
MODEL (8)	VGG19	×	99.64 %	97.81 %	97.51 %

This analysis suggests that most of the models that were trained had been subjected to overfitting of some kind, except those that used data augmentation (that can be referred from “DA used” column). But, the models that had used data augmentation have given a low accuracy. Hence,

an inference can be derived from Table 2 that the models with data augmentation suffer from low accuracy while avoiding overfitting. Examples of this can be seen in the case of simple CNN, VGG16 or VGG19 which obtain a training accuracy of 90.02 %, 86.62 %, and 81.74 %, while a validation and test accuracy of 94.09 % and 94.60 %, 91.95 % and 89.97 % and 91.26 % and 88.95 % had been achieved for these models respectively. On the other hand, all other models have been shown to suffer from overfitting while they usually achieve high training accuracy. The best example of this can be seen in the case of the Xception transfer-learned model which achieves a training accuracy of 99.03 %, validation accuracy of 68.55 % and test accuracy of 66.32 %. We observe that model (6) has been an exception to the rule ($acc_{train} > acc_{validation} > acc_{test}$) among the models not using data augmentation. It may be due to the exponential linear unit (or elu) activation function (described later in eq. (5) and figure 7(e)). A final comparison of all these algorithms has been done with the AiCNNs (whose framework has been proposed in the next section) in Section 5.

4. Proposed Methodology

4.1 Dataset Description

The dataset consists of 3064 T1-weighted contrast-enhanced MRI (CEI-MRI) images of 3 types (i.e. Glioma, Meningioma, and Pituitary) of brain tumor samples. The images were in MatLab data file format (.mat) which had to be converted to Image File format (.jpg) This had been done using the steps mentioned below.

- Step i. The Matlab data files had been read through h5py [58] object.
- Step ii. That object had been used to get its Image array file.
- Step iii. The Image array file had been converted to a NumPy [59] array.
- Step iv. Label had been extracted to a *Label* object from *cjdata/label*.
- Step v. *Label* object had been used to check the type of tumor and save it to Meningioma, Glioma or Pituitary folders using SciPy [60] (i.e. *misc.imsave()* function)

The second dataset consisted of 266 Brain MRI images in (.jpg) format. The negative folder contained about 111 images while the positive contained 155 images [57]. A brief summary of images taken for each tumor has been described below along with a class distribution in Figure 4. A set of 802 images for Meningioma Brain Tumor had been extracted from the dataset and placed under the class of Meningioma. A set of 1486 images for Glioma Brain Tumor has been extracted from the dataset and placed under the class of Meningioma. A set of 936 images for Pituitary Brain tumors had been extracted and placed under the

Pituitary class. Finally, a set of 111 images for Negative tumors has been extracted from the dataset.

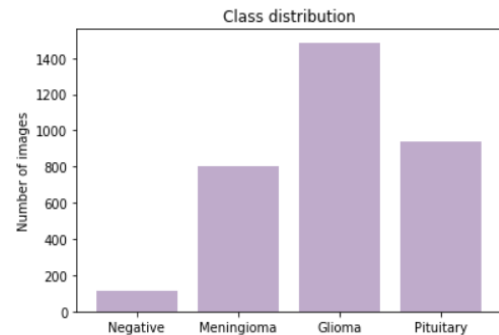


Figure 4. Initial Class Distribution of Brain Tumor Images

4.2 Data Augmentation

To overcome imbalances in the datasets of images, an augmenter module had been implemented. The number of image files in the dataset had been artificially increased by using transformations which didn't change the vectors of pixel intensities for specific classes that the images belonged to. Each image had been transformed according to the steps mentioned below.

- Step i. PIL (or Python Imaging Library) [61] had been used to open the files in the class folders and saved as PNG files (as JPG format doesn't support RGBA channels while PNG does)
- Step ii. A pipeline object had been defined using Augmentor [62] which conducted the following augmentations to the images.
 - a. Rotation: rotating an image with 0.9 probability with maximum left rotation as 10° and maximum right rotation as 15° .
 - b. Flipping: flipping of image horizontally and vertically with a probability of 0.5 and 0.5, respectively.
 - c. Resize: resizing at a probability of 35% to 512×512 .
- Step iii. This had been done till the following number of images had been obtained.
 - a. 1402 images for Meningioma Brain Tumor (after adding 600 images from augmmentor module);
 - b. 1486 original images for Glioma Tumor;
 - c. 1536 images for Pituitary Brain Tumor (after adding 600 images from augmmentor module); and
 - d. 1411 images for Negative (after adding 1300 images from augmmentor module).

The class distribution for different classes (as have been mentioned in figure 4) after passing the images through augmmentor pipeline has been depicted in figure 5, given below. The total no. of images after data augmentation had been 5835 whose class distribution has been aforementioned. In addition to artificially increase the size of the data set, data augmentation can make the resulting

model more invariant to rotation, reflection, translation, and small noise in the pixel values. After this, each image has been resized to 250×250 pixel which has then been given as an input to the input layers Conv2d_16_input, Conv2d_24_input, xception_input, vgg16_input and vgg_19 input of models (1), (2), (5), (7) and (8) from Table 2, respectively.

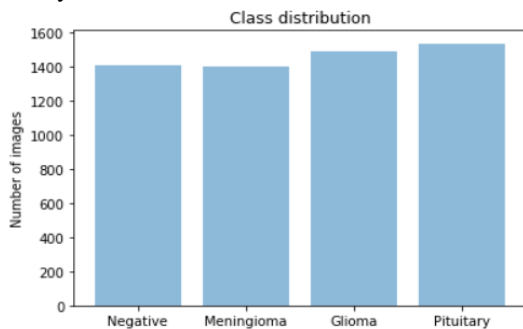


Figure 5. Uniform Class Distribution of Brain Tumor Images after dataset has been pipelined to the Augmentor module.

4.3 Dropouts

To reduce overfitting of the individual models (1-8), regularization technique called dropout had been employed at the fully connected layers, after three input layers, as can be seen from figure 6. Dropout had been set to 0.25 for most of the models, which depicts that the weights of the one-

quarter of the neurons are randomly set to zero while remaining give their original inputs multiplied by the weights. This ensures that the neurons in convolution functions and ANN layers are mostly independent of each other, in the same layer in which they are introduce.

4.4 One hot encoding

It's important to mention that encoding of the brain tumor as 0 for negative, 1 for Meningioma, 2 for Glioma and 3 for Pituitary Brain Tumor has been done using one-hot encoding. This can be depicted as below:

$[1., 0., 0., 0.] = 0$ for Negative;
 $[0., 1., 0., 0.] = 1$ for Meningioma Brain Tumor;
 $[0., 0., 1., 0.] = 2$ for Glioma Brain Tumor; and
 $[0., 0., 0., 1.] = 3$ for Pituitary Brain Tumor

4.5 Model Architecture

The final architecture for AiCNNs has been shown in Figure 8. AiCNNs has utilized models (1), (2), (5), (7), (8) from Table 2 which are described below along with Figures 6(a-e). These models are cascaded and concatenated together as an input to the Artificial Neural Network layer of AiCNNs. The kernel size taken has been 3×3 , uniformly with an input image size of 250×250 and batch size of 32. Figures 6 and 8 have been generated using Matplotlib [63] and Keras [64].

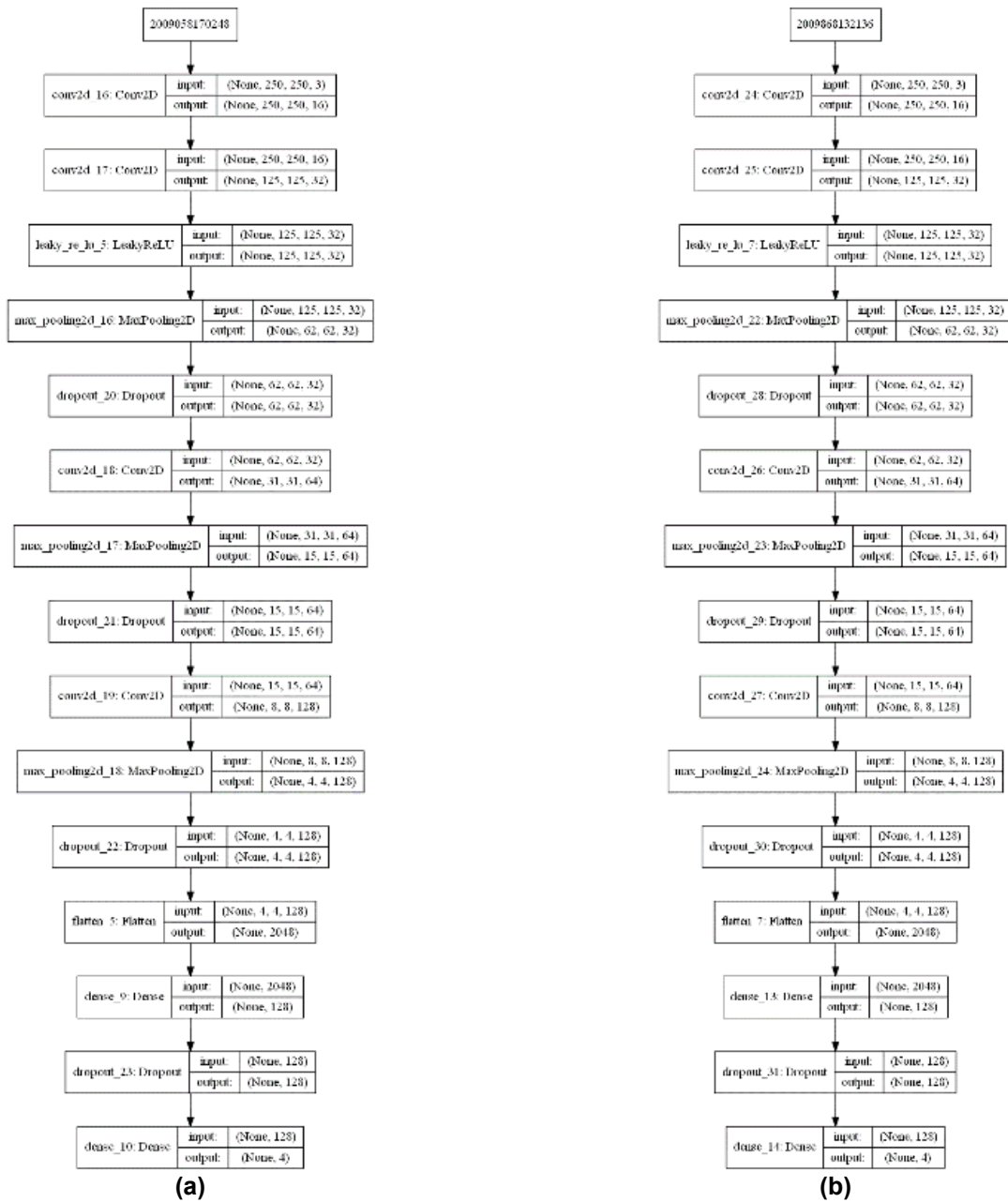


Figure 6(a) The Architecture of model (1) of AiCNNs from Table 2. This is a normal CNN model that doesn't utilize Data Augmentation. **(b)** The Architecture of model (2) of AiCNNs from Table 2. This is a normal CNN model that utilize Data Augmentation.

Model (1) has been depicted in figure 6(a). This model has its first convolutional layer (i.e. Conv2d_16) as an input with a kernel size of (3,3), along with $\tanh(s)$ (or hyperbolic tangent) (as per equation (1)) as an activation function for introducing non-linearity. The equation for the hyperbolic tangent activation function as used in [65] has been given below in equation (1).

$$\tanh(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}} \quad (1)$$

where s is considered as the output values of pixels from the previous layer of the Convolutional Neural Network for individual images. This function creates a S-shaped curve (as shown in figure 7(a)) due to its exponential (i.e. e^s or e^{-s}) terms. This is because the function defined above asymptotes at -1 and +1 as the value of s increases or decreases, respectively. This further helps map negative inputs to be strongly negative while zero inputs to be mapped near zero. Furthermore, this function is differentiable, monotonic, while its derivative is non-monotonic. Here, this function plays an integral role to

classify the images into positive and negative classes of brain tumor.

The layer (discussed earlier i.e. Conv2d_16) gives 16 feature maps in all. In the second convolutional layer (i.e. Conv2d_17), the same kernel size has been used and it gives 32 feature maps of 125×125 when it's passed through the leaky rectified linear unit as an activation function. The equations for Leaky Rectified Linear Unit (LReLU) activation function [66] has been described by the equation (2) given below.

$$\text{Leaky ReLU} = \alpha * x, \quad \text{for } x < 0, \\ = x, \quad \text{for } x \geq 0 \quad (2)$$

where α had been kept at a value of 0.08 for optimality purposes. This function is used to augment to the range of the ReLU function (discussed for next layer). This function is used to counter the dying ReLU problem in which the model becomes unfit to be trained on the data due to loss of important features in negative pixels. It and its derivative are both monotonic in nature.

Subsequently, the first max-pooling layer (i.e. max_poolingd_16) of 2×2 kernel has been utilized to get 32 feature maps, each of 62×62 pixels. Then, a dropout of 0.25 (in dropout_20) i.e. 25 % (i.e. 25% values are randomly set to zero). Similarly, the third convolutional layer (i.e. conv2d_18) gives 64 feature maps with 31×31 pixels each and second max-pooling layer (i.e. max_pooling2d_17) of 2×2 kernel size has been utilized to get 64 feature maps with 15×15 pixels. The convolutional layer described here utilizes a ReLU (i.e. Rectified Linear Unit) activation function. The equation for the ReLU activation function has been described in equation (3) given below.

$$\text{ReLU} = 0, \quad \text{for } x < 0 \\ = x, \quad \text{for } x \geq 0 \quad (3)$$

As can be seen ReLU, has been used to introduce nonlinearity in the form of half rectified (from bottom). ReLU gives a zero output for any value of vectors that is less than 0, while it behaves as an identity function for any vector values that is equal to 0 or greater than 0. It and its derivatives both are monotonic function. Despite this fact, the fact that all negative values become 0, reduce its functionality for the models to train from data. Hence, here it has been used in conjunction with other activation functions.

Further, a dropout of 0.25 has been introduced after this layer again (in dropout_21). Another convolutional-maxpool-dropout block (conv2d_19, max_poolingd_18, dropout_22, respectively) with the same configuration as the previous block has been utilized to obtain 128 feature maps of 4×4 pixels each. These are then flattened (in flatten_5) to get 2048 values which are passed through a fully connected layer of 128 neurons (in dense_9). The last dropout of 0.25 has been introduced in this model which has then been passed through a fully connected or dense layer (dense_10) to get the output as per one hot encoding defined earlier using softmax function. The equation for softmax activation

function which has been mathematically described in equation (4).

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (4)$$

where $i = 1, 2, \dots, K$ and $z = (z_1, z_2, \dots, z_k) \in \mathbb{R}^k$. The main reason to use softmax function is because it gives a probability for each class in multiclass-classification. This probability is calculated using equation described above; where value for specific neuron in the last layers is used as numerator and the sum of all the values in neurons of the last layer is added to get denominator. This provides a probability for image to be classified as a class when a corresponding neuron (hence, it corresponds to that specific class) has the highest value. This function is monotonic and differentiable but its derivative is not monotonic.

Model (2) introduced using figure 6(b) has the same layout as that of model (1) except that it utilizes Keras [64] API's *ImageDataGenerator* which has a shear range, width shift range, height shift range, and a zoom range of 0.1 and a rotation range of 20°. It horizontally flips the image as well. Due to the aforementioned augmentations, test and validation images had been rescaled as well. Figure 6 (c – e) are the transfer learning models (corresponding to model (5), (7) and (8)) that have been utilized in the AiCNNs model. They have briefly been described below.

The model (5) has been depicted in Figure 6(c). Here, the xception model [67] has been loaded from *keras.application* package with model pre-trained on the ImageNet dataset [68-69]. Now, the weights obtained from Xception (xception_input in figure 8) for this dataset (2048 feature maps of 8×8) have been flattened (flatten_11) to get 131072 features which are then passed through a fully connected or dense layer (dense_32) of 128 neurons through a dropout of 0.25 (in dropout_22). This configuration of the dense-dropout layer has been repeated again (as dense_33 and dropout_23, respectively). This model only utilizes rectified linear unit (eq. (3)) as its activation function in 2 blocks mentioned before. Finally, the output from 64 neurons has been passed through the final fully connected layer (i.e. dense_34) by a softmax activation function (eq. (4)).

The model (7) has been depicted in Figure 6(d). Here, the VGG16 model has been loaded from *keras.application* package with model pre-trained on the ImageNet dataset. The weights obtained from VGG16 (vgg16_input in figure 8) for this dataset (512 feature maps of 7×7) have been flattened (flatten_9) to get 25088 features which are then passed through a fully connected or dense layer (dense_25) of 128 neurons through a dropout of 0.4 (in dropout_16). This model only utilizes a rectified linear unit (eq. (3)) as its activation function in the block mentioned earlier. Finally, the output from 128 neurons has been passed through the final fully connected layer (i.e. dense_26) by a softmax activation function (eq. (4)).

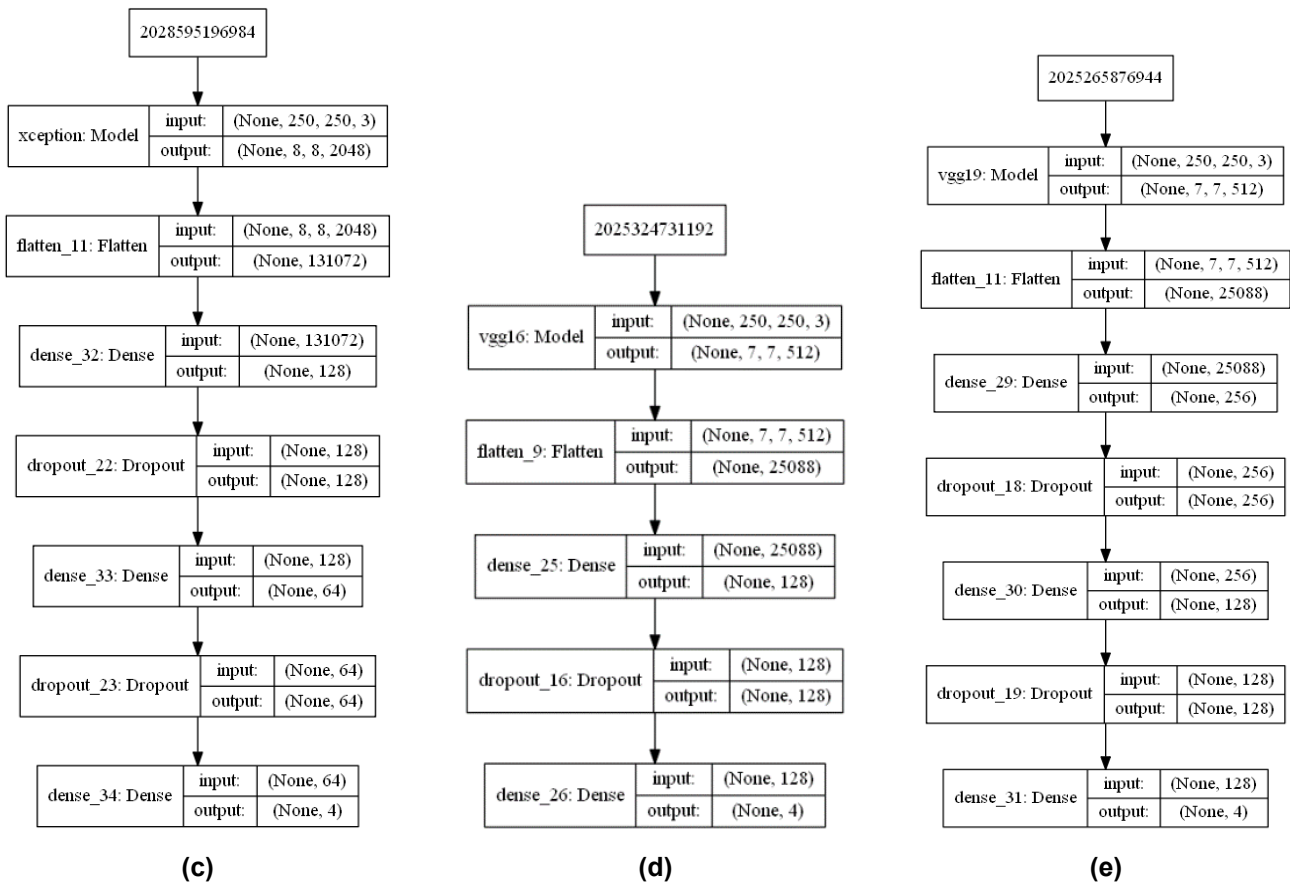


Figure 6(c) The Architecture of model (5) of AiCNNs (i.e. Artificially integrated Convolution Neural Networks) from Table 2. This is a transfer-learned Xception model. **(d)** The Architecture of model (7) of AiCNNs (i.e. Artificially integrated Convolution Neural Networks) from Table 2. This is a transfer-learned VGG16 model. **(e)** The Architecture of model (8) of AiCNNs (i.e. Artificially integrated Convolution Neural Networks) from Table 2. This is a transfer-learned VGG19 model. None of these models use data augmentation as it tends to increase the overfitting which will be discussed later.

The model (8) has been depicted in Figure 6(e). Here, the vgg19 model has been loaded from *keras.application* package with model pre-trained on the ImageNet dataset. The weights that are obtained from VGG19 model (vgg19_input in figure 8) for this dataset (512 feature maps of 7×7) have to be flattened (flatten_11) to get 25088 features which are then passed through a fully connected or dense layer (dense_29) of 256 neurons through a dropout of 0.25 (in dropout_18). This configuration of the dense-dropout layer has been repeated again (as dense_30 and dropout_19, respectively). This model only utilizes a rectified linear unit (eq. (3)) as its activation function in 2 blocks mentioned before. Finally, the output from 64 neurons has been passed through the final fully connected layer (i.e. dense_31) by a softmax activation function (eq. (4)).

Finally, the model (6) utilized the exponential linear unit (ELU) as an activation function in all its dense layers, the formula for which has been described in equation (3).

$$R(z) = \begin{cases} z & z > 0 \\ \alpha \cdot (e^z - 1) & z \leq 0 \end{cases} \quad (5)$$

where α had been kept at a value of 1 as per Keras [64] documentation of Tensorflow. ELU is very similar to ReLU and it slowly smoothens until its output equals $-\alpha$ whereas ReLU smoothens sharply. Finally, ELU tends to account for negative results and hence can be comparatively better than ReLU. Its monotonic and differentiable.

The graphs for all 5 of the activation functions that have been mentioned from eq. (1) – (5) have been given below in figure 7(a)-(e), respectively.

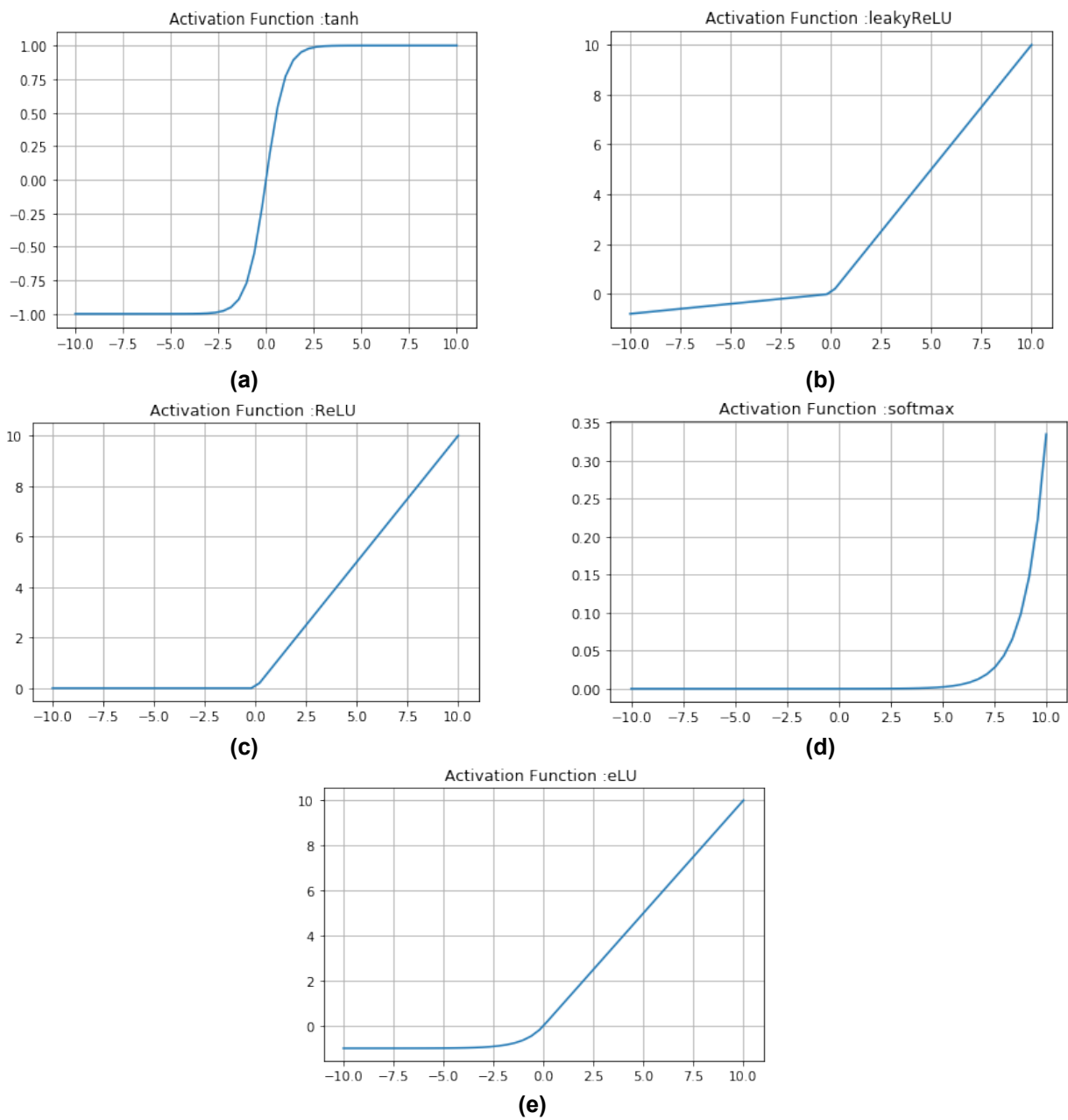


Figure 7(a). A representation of hyperbolic tangent activation function; **(b).** Leaky Rectified Linear Unit activation function; **(c).** A representation of Exponential Linear Unit activation function; **(d).** A representation of Rectified Linear Unit activation function; and **(e).** A representation of Softmax function.

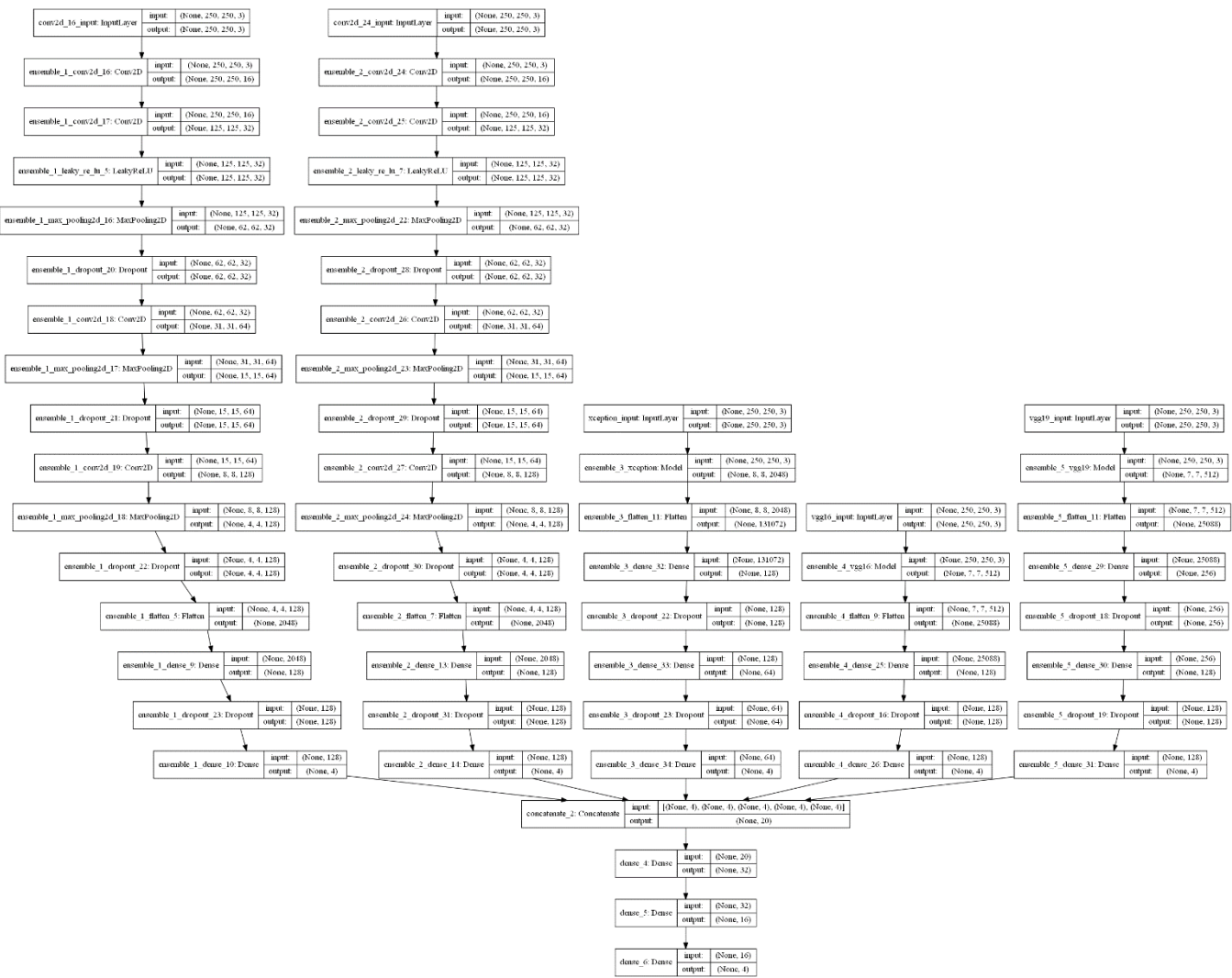


Figure 8. The Architecture of AiCNNs which is cascade of 5 classifiers (models (1), (2), (5), (7), (8)) from Table 2.

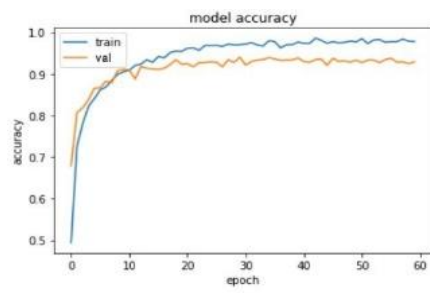
Now, the AiCNNs model has been depicted in figure 8 and has been cascaded and concatenated from models (1), (2), (5), (7) and (8) of Table 2. Here, all the 5 models discussed above had been loaded using the *load_model()* function of *keras.models* package. The inputs that had been fed to all models remained the same except that of the model (2) whose inputs had been scaled by a factor of 255.0 to normalize the pixel values around 1. The outputs of these models had then been concatenated together using *concatenate()* function of *keras.layers.merge* package. This concatenated layer of outputs had been passed through 2 hidden layers, first having 32 hidden layer neurons (*dense_1*), and the second having 16 hidden layer neurons (*dense_2*). Both these hidden layers used ReLU (eq. (3)) activation layer. Finally, the output layer has 4 output layer neurons corresponding to the 4 classes discussed earlier. This model also calculates categorical cross-entropy loss with AdaDelta optimizer. The complete architecture of the AiCNNs model has been depicted in Figure 8. AiCNNs have been inspired by the integrated stacking model [70]

which utilizes a similar cascade of classifiers where ANN layer is used as a meta-learner model or level 1 learner (or model) while CNNs are used as sub-models or level 0 learners (or models).

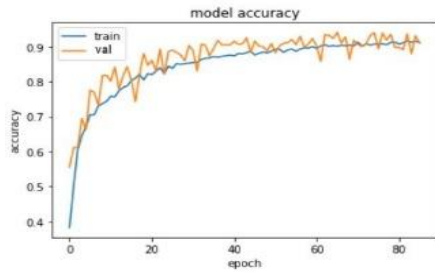
5. Results & Analysis

Initially, the 5 models (i.e. model (1), model (2), model (5), model (7), and model (8) of Table 2) have been trained on the same dataset, with only model (2) taking the input as scaled values (on 255.0 for proper normalization). These were trained on 3501 images, validated on 1167 images and tested on 1167 images. The training curve for these models has been given below. A detailed comparison of these models along with models (3), (4), and (6) has been done in Table 3 while their training history has been depicted in figure 9 (a – e) and figure 10 (a – e), which respectively represent their training history for model accuracy and model loss.

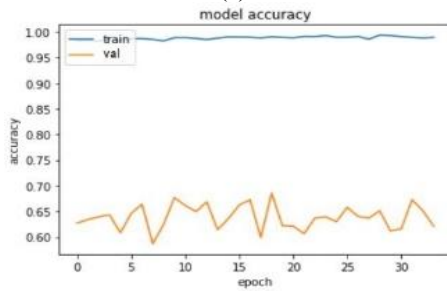




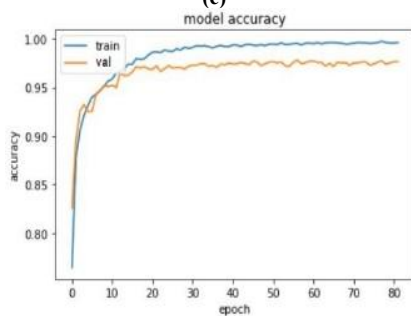
(a)



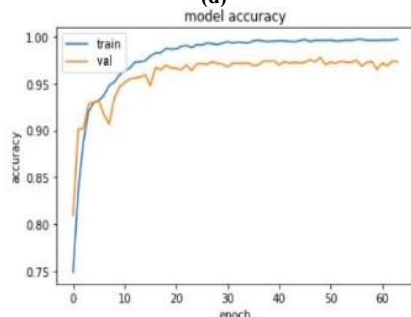
(b)



(c)

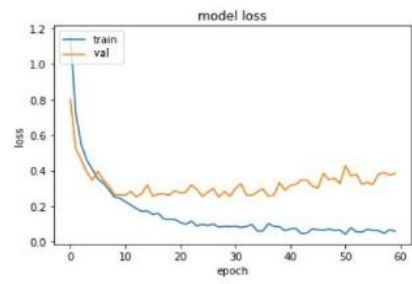


(d)

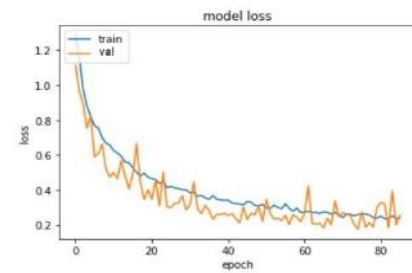


(e)

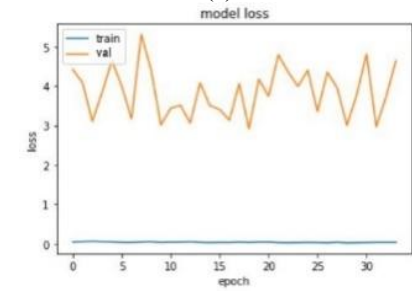
Figure 9 (a) – (e). History of Validation (in orange) & Training Accuracy (in blue) during the training of models (1), (2), (5), (7) and (8), resp. for the classification of MRI scans.



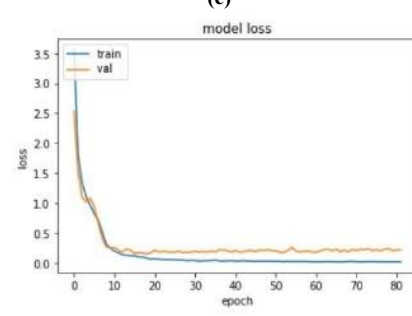
(a)



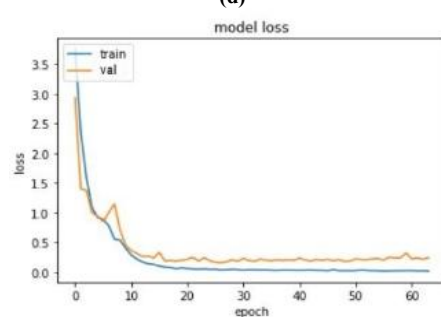
(b)



(c)



(d)



(e)

Figure 10 (a) – (e). History of Validation (in orange) & Training Loss (in blue) during the training of models (1), (2), (5), (7) and (8), resp. for the classification of MRI scans.

The model had been run on 100 epochs with batch size 32. This resulted in a training accuracy of 99.20% and training loss (categorical cross-entropy) of 0.0306. While the validation and testing accuracy had considerably been reduced to approximately 99.486 % and 98.886 % respectively. Also, the validation loss (categorical cross-entropy) for the model has been noted to be 0.0303 which

depicts that this model doesn't overfit the dataset that had been taken.

AiCNNs had been trained on a set of 3501 images, validated on a set of 1167 images and tested on a set of 1167 images. The curves for validation and training history has been displayed in figure 11 and figure 12. These curves represent the history of training of the AiCNNs (not including the 5 models that have been trained earlier).

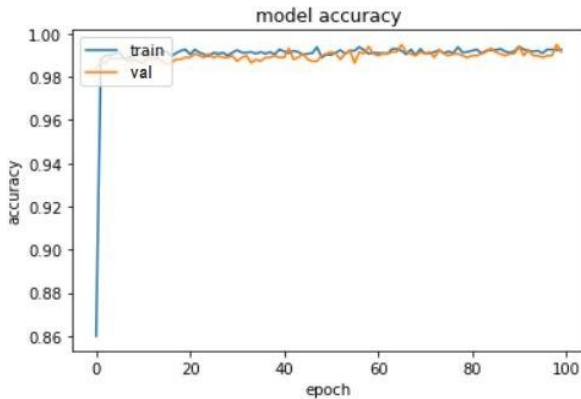


Figure 11. Validation (in orange) & Training Accuracy (in blue) during the training of AiCNNs for the classification of MRI scans.

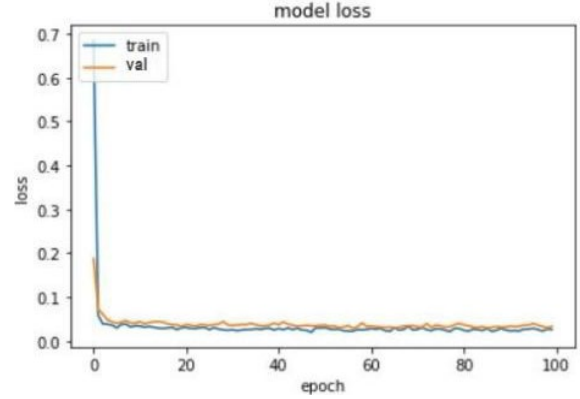
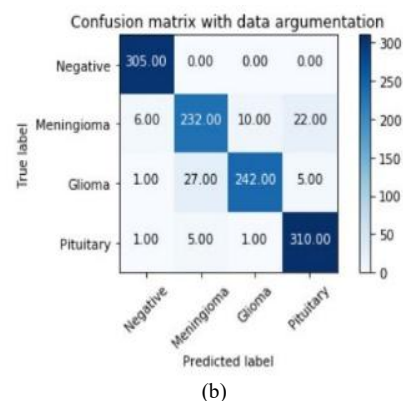
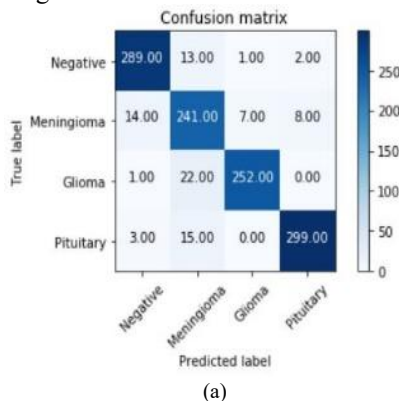


Figure 12: Validation (in orange) & Training Loss (in blue) during the training of AiCNNs for classification of MRI scans.

The confusion matrix for all the models whose training history has been mentioned earlier in figure 9 (a – e) and figure 10 (a – e), has been given in figure 13 (a – e). We can observe that the simple CNN model (1) inaccurately classifies 14 instances of Meningioma brain tumor as negative, 1 instance of Glioma and 3 instances of Pituitary brain tumor as Negative. On the other hand, the model (2) which utilizes data augmentation has misclassified only 8 instances of brain tumors as negative which had been a much more desirable result as compared to the earlier models. All these wrongly classified instances are known as False Negatives (FNs). In real-world scenarios, less false negatives are requirement for a medical system, as a system detecting FNs may give a negative result to those suffering from brain tumor, which can prove fatal.

Model (5), whose confusion matrix has been plotted alongside model (1-2) (7-8), performs the worst out of the 5 models. This has been subjected to its low accuracy and high overfitting of the data to the model. This model also has a low validation accuracy and loss as can be seen in figure 9(c) and figure 10(c). The value and performance of model (7) and (8) in real-life scenario lies in-between model (1) and (2) as these classify 9 and 11 MRI (20 in total) scans as FNs. So, these confusion matrices have been used for cross-checking the previous test accuracies (acc_{test}) that had been described in Table 2 and have finally been stated in Table 3. Figure 14 depicts the confusion matrix corresponding to AiCNNs (architecture for which has been defined in figure 8 and whose history of training (with accuracy and categorical cross-entropy loss) had been depicted in figure 11 and figure 12). AiCNNs has reduced the FNs (for brain tumor (all 3 types)) instances to 3 which has been the best achievable amongst all the models discussed and worked upon, so far. It also has less misclassified points, better validation accuracy, and validation loss compared to models (1), (2), (5), (7) and (8) (discussed earlier in Table 2) as has been discussed with reference to Table 3.

Model (5), whose confusion matrix has been plotted alongside model (1-2) (7-8), performs the worst out of the 5 models. This has been subjected to its low accuracy and high overfitting of the data to the model. This model also



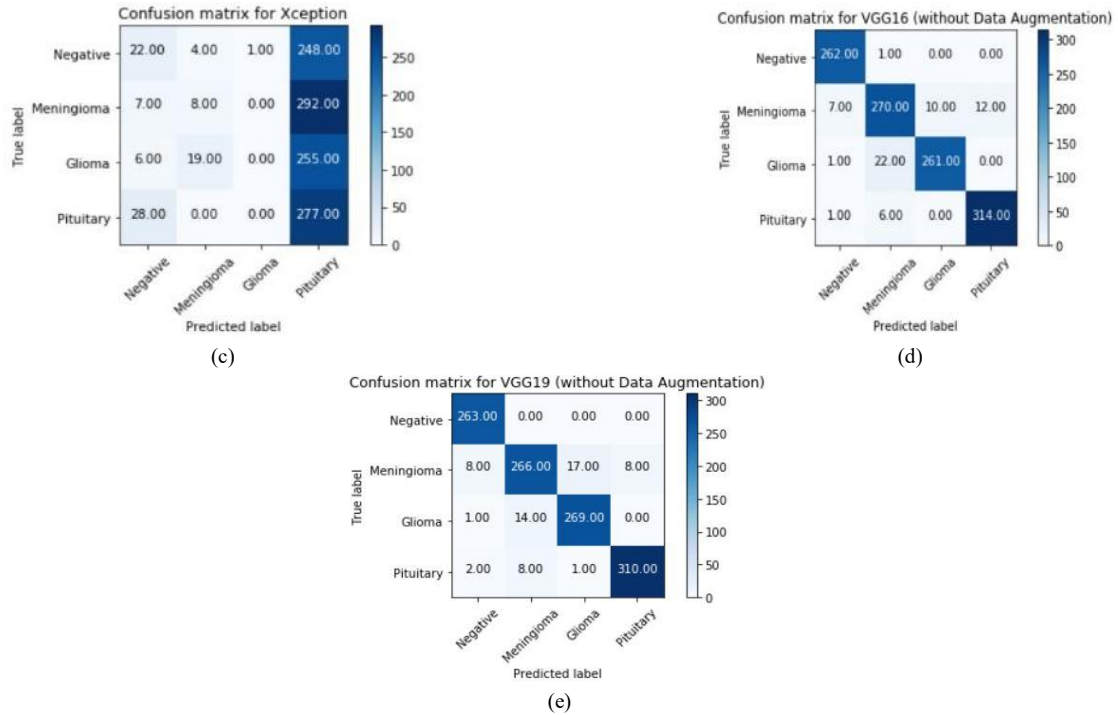


Figure 13 (a – e). Confusion Matrix for 4 classes namely, Negative, Meningioma, Glioma, and Pituitary when using Model (1), Model (2), Model (5), Model (7) and Model (8).

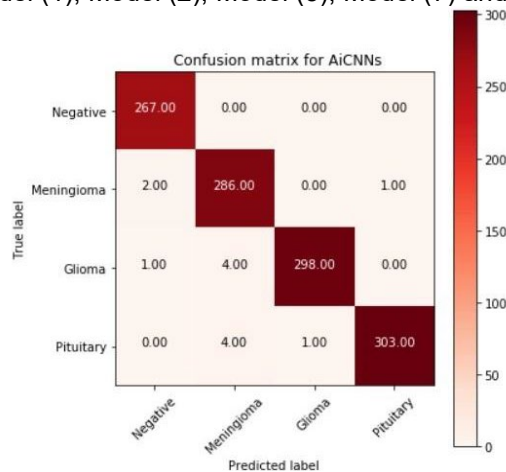


Figure 14. Confusion Matrix for 4 classes, namely, Negative, Meningioma, Glioma, and Pituitary when using AiCNNs (i.e. Artificially integrated Convolutional Neural Networks) Model.

A detailed comparison of all the models that have been implemented has been done in Table 3, which has been given below. Note that AiCNNs has been one of the best models among all the models mentioned here. This has been due to the fact that 5 models compute the weights to be given to the concatenation layer which is a vital part for

AiCNNs, discussed in section 4. The calculation of features from the 5 models (namely model (1), (2), (5), (7), and (8)) make AiCNNs more robust. If the model had to be fine-tuned it may work even better than it currently does.

Table 3. A holistic comparison of different CNN models for the classification of different types of tumors. These algorithms are compared on basis if data augmentation had been used or not, training accuracy (ACC_{TRAIN}), training loss ($LOSS_{TRAIN}$), validation accuracy (ACC_{VALID}), validation loss ($LOSS_{VALID}$), and test accuracy (ACC_{TEST}).

MODEL NAME	DA USED	ACC_{TRAIN}	$LOSS_{TRAIN}$	ACC_{VALID}	$LOSS_{VALID}$	ACC_{TEST}
MODEL (1)	No	97.00 %	0.0847	94.00 %	0.2547	92.63 %

MODEL (2)	Yes	90.02 %	0.2765	94.09 %	0.1832	94.60 %
MODEL (3)	Yes	86.62 %	0.3310	91.95 %	0.2199	89.97 %
MODEL (4)	Yes	81.74 %	0.4286	91.26 %	0.2960	88.95 %
MODEL (5)	No	99.03 %	0.0329	68.55 %	2.9137	66.32 %
MODEL (6)	No	97.29 %	0.1924	94.26 %	0.4437	97.43 %
MODEL (7)	No	99.52 %	0.0228	97.79 %	0.1787	97.43 %
MODEL (8)	No	99.64 %	0.0239	97.81 %	0.1794	97.51 %
AiCNNs	Partially [‡]	99.20 %	0.0306	99.49 %	0.0303	98.89 %

[‡] It partially uses data augmentation as Model (2) (which also utilizes data augmentation) has been cascaded into it.

An expeditious investigation of Table 3 depicts that Model (5) (i.e. Xception model) converges the fastest for the dataset described in section 4 but this model provides the least validation and test accuracy. It can also be observed that only AiCNNs utilizes data augmentation partially due to its nature of cascading models, in which model (2) (that uses data augmentation) has been used with model (1), model (5), model (7) and model (8) (which don't use data augmentation). Model (8) (i.e. VGG19 transfer-learned model) has achieved the maximum training accuracy (ACC_{TRAIN}). While the minimum training loss ($LOSS_{TRAIN}$) had been observed with Model (7) (i.e. VGG16 transfer-learned model). Now, AiCNNs have been observed to achieve the best ACC_{VALID} , $LOSS_{VALID}$, and ACC_{TEST} . The four models – namely, model (2), model (3), model (4), and AiCNNs are the only models that don't follow the general trend of ($ACC_{TRAIN} > ACC_{VALID}$) and ($LOSS_{VALID} > LOSS_{TRAIN}$). This depicts that these models have been more robust to overfitting when being trained on mentioned combined dataset. It's important to note that the introduction of only a single model robust to overfitting (i.e. model (2)) has made the AiCNNs robust to it as well.

6. Discussions & Conclusion

The model introduced in this research, AiCNNs achieved a training accuracy (ACC_{TRAIN}) of 99.20 % and training loss (i.e. categorical cross-entropy loss) ($LOSS_{TRAIN}$) of 0.0306 on a set of 3501 images. It had been robust to overfitting and achieved a validation accuracy (ACC_{VALID}) of 99.49 % and validation loss (i.e. categorical cross-entropy loss) ($LOSS_{VALID}$) of 0.0303 on a set of 1167 images. This model has been tested on yet another 1167 images and achieved an accuracy (ACC_{TEST}) of 98.89 % as can be seen from figure 14. It used a cascaded ensemble of 5 models (i.e. model (1) (a simple CNN), model (2) (CNN with data augmentation), model (5) (Xception model), model (7) (VGG16 model) and model (8) (VGG19 model)) as discussed earlier in section 4 and depicted in figure 6 and 8. These all models had been trained on the ratio of 3:1:1 for train, validation, a test set

of images (with 5835 being the total number of images) which is similar to AiCNNs as discussed above. It used 4 types of activation function which have been best described by eq. (1 – 4) and figure 7 (a – d). And, it has been observed that the introduction of only one data augmented model (i.e. model (2)) resulted in the model becoming robust to overfitting.

Although this model achieves the highest testing (ACC_{TEST}) and validation accuracy (ACC_{VALID}) and least validation loss ($LOSS_{VALID}$) compared to the models mentioned earlier. This has been ascribed due to the fact that the 5 models first have to calculate 4 vectors by a 5-way cascaded process and then these were then used to train the hidden layer (ANN layer) weights.

This work can further be extended through the utilization of all the models. Some more areas where this work can be extended have been defined below. AiCNNs used convolution functions that were made through manual trial-and-error method which is not an efficient means for defining a good classification model. It's important to mention that genetic algorithms may be utilized which can help to design an optimal CNN for classification purposes [71-72]. Along with this, generation of ANN layers can also be conducted through genetic algorithm [73-74] and evolutionary strategies [75] such as CMA-ES (Covariance Matrix Adaptation Evolution Strategy) [76-77] and PEPG (Parameter-Exploring Policy Gradients) [78-79], for which a further perusal is needed. AiCNNs model can further be extended to include more types of brain tumors such as Neurofibroma and Osteoma. It can also be extended to certain subtypes of the tumors for Meningioma, Glioma, and Pituitary such as Convexity meningioma and Skull base meningioma, Astrocytoma and Brain stem glioma, and Craniopharyngioma and Pituitary adenoma [80], respectively.

References

- [1] Mustaf, M., Sali, A. F., Illzam, E. M., Sharifa, A. M., & Nang M. K. (2018). Brain cancer: current concepts, diagnosis and prognosis. *IOSR Journal of Dental and Medical Sciences* 17(3), 41-46.

- [2] Nayak, L., Lee, E. Q., & Wen, P. Y. (2012). Epidemiology of brain metastases. *Current oncology reports*, 14(1), 48-54.
- [3] Stewart, B. W. K. P., & Wild, C. P. (2014). World cancer report 2014.
- [4] Schellinger, P. D., Meinck, H. M., & Thron, A. (1999). Diagnostic accuracy of MRI compared to CCT in patients with brain metastases. *Journal of neuro-oncology*, 44(3), 275-281.
- [5] Mittal, M., Goyal, L. M., Sethi, J. K., & Hemanth, D. J. (2019). Monitoring the Impact of Economic Crisis on Crime in India Using Machine Learning. *Computational Economics*, 53(4), 1467-1485.
- [6] Chiclana, F., Kumar, R., Mittal, M., Khari, M., Chatterjee, J. M., & Baik, S. W. (2018). ARM-AMO: an efficient association rule mining algorithm based on animal migration optimization. *Knowledge-Based Systems*, 154, 68-80.
- [7] Walsh, C. G., Ribeiro, J. D., & Franklin, J. C. (2017). Predicting risk of suicide attempts over time through machine learning. *Clinical Psychological Science*, 5(3), 457-469.
- [8] Dwyer, D. B., Falkai, P., & Koutsouleris, N. (2018). Machine learning approaches for clinical psychology and psychiatry. *Annual review of clinical psychology*, 14, 91-118.
- [9] Peng, H., & Bai, X. (2017). Limits of Machine Learning Approach on Improving Orbit Prediction Accuracy using Support Vector Machine. In *Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*.
- [10] Singh, R., Gahlot, A., & Mittal, M. (2019). IoT based intelligent robot for various disasters monitoring and prevention with visual data manipulating. *Int. J. Tomogr. Simul.*, 32(1), 90-99.
- [11] Yadav, M., Purwar, R. K., & Mittal, M. (2018). Handwritten Hindi character recognition: a review. *IET Image Processing*, 12(11), 1919-1933.
- [12] Ahneman, D. T., Estrada, J. G., Lin, S., Dreher, S. D., & Doyle, A. G. (2018). Predicting reaction performance in C-N cross-coupling using machine learning. *Science*, 360(6385), 186-190.
- [13] Shastri, M., Roy, S., & Mittal, M. (2019). Stock Price Prediction using Artificial Neural Model: An Application of Big Data. *EAI Endorsed Trans. Scalable Information Systems*, 6(20), e1.
- [14] Mangasarian, O. L., Street, W. N., & Wolberg, W. H. (1995). Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, 43(4), 570-577.
- [15] Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., & Fotiadis, D. I. (2015). Machine learning applications in cancer prognosis and prediction. *Computational and structural biotechnology journal*, 13, 8-17.
- [16] Hemanth, D. J., Anitha, J., & Mittal, M. (2018). Diabetic retinopathy diagnosis from retinal images using modified hopfield neural network. *Journal of medical systems*, 42(12), 247.
- [17] Mani, S., Chen, Y., Elasy, T., Clayton, W., & Denny, J. (2012). Type 2 diabetes risk forecasting from EMR data using machine learning. In *AMIA annual symposium proceedings* (Vol. 2012, p. 606). American Medical Informatics Association.
- [18] Kaur, P., & Sharma, M. (2018). Analysis of data mining and soft computing techniques in prospecting diabetes disorder in human beings: a review. *Int. J. Pharm. Sci. Res*, 9, 2700-2719.
- [19] Sharma, M., Singh, G., & Singh, R. (2019). An Advanced Conceptual Diagnostic Healthcare Framework for Diabetes and Cardiovascular Disorders. *arXiv preprint arXiv:1901.10530*.
- [20] Gonsalves, A. H., Thabtah, F., Mohammad, R. M. A., & Singh, G. (2019, July). Prediction of Coronary Heart Disease using Machine Learning: An Experimental Analysis. In *Proceedings of the 2019 3rd International Conference on Deep Learning Technologies* (pp. 51-56). ACM.
- [21] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639), 115.
- [22] Kaur, P., & Sharma, M. (2019). Diagnosis of Human Psychological Disorders using Supervised Learning and Nature-Inspired Computing Techniques: A Meta-Analysis. *Journal of medical systems*, 43(7), 204.
- [23] Sharma, M., Singh, G., & Singh, R. (2017). Stark assessment of lifestyle based human disorders using data mining based learning techniques. *IRBM*, 38(6), 305-324.
- [24] Kaur, P., Sharma, M., & Mittal, M. (2018). Big data and machine learning based secure healthcare framework. *Procedia computer science*, 132, 1049-1059.
- [25] Cheng, J., Yang, W., Huang, M., Huang, W., Jiang, J., Zhou, Y., ... & Chen, W. (2016). Retrieval of brain tumors by adaptive spatial pooling and Fisher vector representation. *PloS one*, 11(6), e0157112.
- [26] Shubhangi, D. C., & Hiremath, P. S. (2009, January). Support vector machine (SVM) classifier for brain tumor detection. In *Proceedings of the International Conference on Advances in Computing, Communication and Control* (pp. 444-448). ACM.
- [27] Joshi, D. M., Rana, N. K., & Misra, V. (2010, May). Classification of brain cancer using artificial neural network. In *2010 2nd International Conference on Electronic Computer Technology* (pp. 112-116). IEEE.
- [28] Othman, M. F., & Basri, M. A. M. (2011, January). Probabilistic neural network for brain tumor classification. In *2011 Second International Conference on Intelligent Systems, Modelling and Simulation* (pp. 136-138). IEEE.
- [29] Deepa, S. N., & Devi, B. A. (2012, January). Artificial neural networks design for classification of brain tumour. In *2012 International Conference on Computer Communication and Informatics* (pp. 1-6). IEEE.
- [30] Cruz-Roa, A., Arévalo, J., Judkins, A., Madabhushi, A., & González, F. (2015, December). A method for medulloblastoma tumor differentiation based on convolutional neural networks and transfer learning. In *11th International Symposium on Medical Information Processing and Analysis* (Vol. 9681, p. 968103). International Society for Optics and Photonics.
- [31] Talo, M., Baloglu, U. B., Yıldırım, Ö., & Acharya, U. R. (2019). Application of deep transfer learning for automated brain abnormality classification using MR images. *Cognitive Systems Research*, 54, 176-188.
- [32] Kwan, R. S., Evans, A. C., & Pike, G. B. (1999). MRI simulation-based evaluation of image-processing and classification methods. *IEEE transactions on medical imaging*, 18(11), 1085-1097.
- [33] Rinck, P. A. (2008). A short history of magnetic resonance imaging. *Spectroscopy Europe*, 20(1), 7-10.
- [34] Kononenko, I. (2001). Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in medicine*, 23(1), 89-109.

- [35] Jiang, J., Trundle, P., & Ren, J. (2010). Medical image analysis with artificial neural networks. *Computerized Medical Imaging and Graphics*, 34(8), 617-631.
- [36] Zacharaki, E. I., Wang, S., Chawla, S., Soo Yoo, D., Wolf, R., Melhem, E. R., & Davatzikos, C. (2009). Classification of brain tumor type and grade using MRI texture and shape in a machine learning scheme. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 62(6), 1609-1618.
- [37] Al-Kadi, O. S. (2010). Texture measures combination for improved meningioma classification of histopathological images. *Pattern recognition*, 43(6), 2043-2053.
- [38] El-Dahshan, E. S. A., Mohsen, H. M., Revett, K., & Salem, A. B. M. (2014). Computer-aided diagnosis of human brain tumor through MRI: A survey and a new algorithm. *Expert systems with Applications*, 41(11), 5526-5545.
- [39] Subash, N., & Rajeesh, J. (2015). Brain tumor classification Using machine learning. *Int. J. Comput. Technol. Appl.*, 8(5), 2335-2341.
- [40] Xu, Y., Jia, Z., Ai, Y., Zhang, F., Lai, M., Eric, I., & Chang, C. (2015, April). Deep convolutional activation features for large scale brain tumor histopathology image classification and segmentation. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 947-951). IEEE.
- [41] Cheng, J., Huang, W., Cao, S., Yang, R., Yang, W., Yun, Z., ... & Feng, Q. (2015). Enhanced performance of brain tumor classification via tumor region augmentation and partition. *PLoS one*, 10(10), e0140381.
- [42] Pereira, S., Pinto, A., Alves, V., & Silva, C. A. (2015, October). Deep convolutional neural networks for the segmentation of gliomas in multi-sequence MRI. In *BrainLes 2015* (pp. 131-143). Springer, Cham.
- [43] Barker, J., Hoogi, A., Depeursinge, A., & Rubin, D. L. (2016). Automated classification of brain tumor type in whole-slide digital pathology images using local representative tiles. *Medical image analysis*, 30, 60-71.
- [44] Zhao, L., & Jia, K. (2016). Multiscale CNNs for brain tumor segmentation and diagnosis. *Computational and mathematical methods in medicine*, 2016.
- [45] Işın, A., Direkoğlu, C., & Şah, M. (2016). Review of MRI-based brain tumor image segmentation using deep learning methods. *Procedia Computer Science*, 102, 317-324.
- [46] Chang, P. D. (2016, October). Fully convolutional deep residual neural networks for brain tumor segmentation. In *International Workshop on Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries* (pp. 108-118). Springer, Cham.
- [47] Akkus, Z., Galimzianova, A., Hoogi, A., Rubin, D. L., & Erickson, B. J. (2017). Deep learning for brain MRI segmentation: state of the art and future directions. *Journal of digital imaging*, 30(4), 449-459.
- [48] Wang, G., Li, W., Ourselin, S., & Vercauteren, T. (2017, September). Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks. In *International MICCAI Brainlesion Workshop* (pp. 178-190). Springer, Cham.
- [49] Mittal, M., Goyal, L. M., Kaur, S., Kaur, I., Verma, A., & Hemanth, D. J. (2019). Deep learning based enhanced tumor segmentation approach for MR brain images. *Applied Soft Computing*, 78, 346-354.
- [50] Kaur, B., Sharma, M., Mittal, M., Verma, A., Goyal, L. M., & Hemanth, D. J. (2018). An improved salient object detection algorithm combining background and foreground connectivity for brain image analysis. *Computers & Electrical Engineering*, 71, 692-703.
- [51] Anjali, R., & Priya, S. (2017). AN EFFICIENT CLASSIFIER FOR BRAIN TUMOR CLASSIFICATION.
- [52] Ari, A., & Hanbay, D. (2018). Deep learning based brain tumor classification and detection system. *Turkish Journal of Electrical Engineering & Computer Sciences*, 26(5), 2275-2286.
- [53] Soltaninejad, M., Zhang, L., Lambrou, T., Allinson, N., & Ye, X. (2017). Multimodal MRI brain tumor segmentation using random forests with features learned from fully convolutional neural network. *arXiv preprint arXiv:1704.08134*.
- [54] Kumar, P., & Vijayakumar, B. (2015). Brain tumour Mr image segmentation and classification using by PCA and RBF kernel based support vector machine. *Middle-East Journal of Scientific Research*, 23(9), 2106-2116.
- [55] Zaw, H. T., Maneerat, N., & Win, K. Y. (2019, July). Brain tumor detection based on Naïve Bayes Classification. In *2019 5th International Conference on Engineering, Applied Sciences and Technology (ICEAST)* (pp. 1-4). IEEE.
- [56] Cheng, J. (2017, April 2). brain tumor dataset. figshare. Retrieved 23 September 2019, from https://figshare.com/articles/brain_tumor_dataset/1512427/5.
- [57] Brain MRI Images for Brain Tumor Detection. (2019). Retrieved 23 September 2019, from <https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection>
- [58] Collette, A. (2013). Python and HDF5. Retrieved 23 September 2019, from <https://www.h5py.org/>
- [59] Oliphant, T. E. (2006). *A guide to NumPy* (Vol. 1). Trelgol Publishing USA.
- [60] Jones, E., Oliphant, T., Peterson, P., & others. (2001). *SciPy: Open source scientific tools for Python*. Retrieved from <http://www.scipy.org/>
- [61] Python Imaging Library (PIL). (2013). Retrieved 23 September 2019, from <http://www.pythonware.com/products/pil/>.
- [62] Bloice, M. D., Stocker, C., & Holzinger, A. (2017). Augmentor: an image augmentation library for machine learning. *arXiv preprint arXiv:1708.04680*.
- [63] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in science & engineering*, 9(3), 90.
- [64] François, C. (2015, March 27). keras. Retrieved from <https://github.com/fchollet/keras>
- [65] Mathias, A. C., & Rech, P. C. (2012). Hopfield neural network: the hyperbolic tangent and the piecewise-linear activation functions. *Neural Networks*, 34, 42-45.
- [66] Pedamonti, D. (2018). Comparison of non-linear activation functions for deep neural networks on MNIST classification task. *arXiv preprint arXiv:1804.02763*.
- [67] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1251-1258).
- [68] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [69] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248-255). Ieee.
- [70] Brownlee, J. (2018, December). How to Develop a Stacking Ensemble for Deep Learning Neural Networks in Python with Keras. Retrieved 23 September 2019, from

- <https://machinelearningmastery.com/stacking-ensemble-for-deep-learning-neural-networks/>.
- [71] Sun, Y., Xue, B., Zhang, M., & Yen, G. G. (2018). Automatically designing CNN architectures using genetic algorithm for image classification. *arXiv preprint arXiv:1808.03818*.
- [72] Sun, Y., Xue, B., Zhang, M., & Yen, G. G. (2019). Evolving deep convolutional neural networks for image classification. *IEEE Transactions on Evolutionary Computation*.
- [73] Castillo, P. A., Arenas, M. G., Castillo-Valdivieso, J. J., Merelo, J. J., Prieto, A., & Romero, G. (2003). Artificial neural networks design using evolutionary algorithms. In *Advances in Soft Computing* (pp. 43-52). Springer, London.
- [74] Ding, S., Li, H., Su, C., Yu, J., & Jin, F. (2013). Evolutionary artificial neural networks: a review. *Artificial Intelligence Review*, 39(3), 251-260.
- [75] Hansen, N., & Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2), 159-195.
- [76] Hansen, N., Müller, S. D., & Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary computation*, 11(1), 1-18.
- [77] Hansen, N. (2006). The CMA evolution strategy: a comparing review. In *Towards a new evolutionary computation* (pp. 75-102). Springer, Berlin, Heidelberg.
- [78] Sehne, F., Osendorfer, C., Rückstieß, T., Graves, A., Peters, J., & Schmidhuber, J. (2010). Parameter-exploring policy gradients. *Neural Networks*, 23(4), 551-559.
- [79] Sehne, F., Graves, A., Osendorfer, C., & Schmidhuber, J. (2010, December). Multimodal parameter-exploring policy gradients. In *2010 Ninth International Conference on Machine Learning and Applications* (pp. 113-118). IEEE.
- [80] Kieffer, S. (2019). Types of Brain Tumors – Glioma, Meningioma, Pituitary Tumors, Metastatic, Skull Base Tumors | Johns Hopkins Medicine. Retrieved 24 September 2019, from https://www.hopkinsmedicine.org/neurology_neurosurgery/centers_clinics/brain_tumor/about-brain-tumors/types.html