Available online at www.sciencedirect.com

ScienceDirect

Cognitive Systems Research 57 (2019) 147–159

Cognitive Systems RESEARCH

www.elsevier.com/locate/cogsys

Convolutional neural network based Alzheimer's disease classification from magnetic resonance brain images

Rachna Jain [a], Nikita Jain [a], Akshay Aggarwal [a], D. Jude Hemanth [b,*]

[a] Computer Science and Engineering, Bharati Vidyapeeth's College of Engineering, New Delhi, India
[b] Department of ECE, Karunya University, Coimbatore, India

Received 22 October 2018; received in revised form 14 December 2018; accepted 26 December 2018
Available online 4 January 2019

Abstract

Alzheimer's disease, the most common form of dementia is a neurodegenerative brain order that has currently no cure for it. Hence, early diagnosis of such disease using computer-aided systems is a subject of great importance and extensive research amongst researchers. Nowadays, deep learning or particularly convolutional neural network (CNN) is getting more attention due to its state-of-the-art performances in variety of computer vision tasks such as visual object classification, detection and segmentation. Several recent studies, that have used brain MRI scans and deep learning have shown promising results for diagnosis of Alzheimer's disease. However, most common issue with deep learning architectures such as CNN is that they require large amount of data for training. In this paper, a mathematical model $P_F S_E C_{TL}$ based on transfer learning is used in which a CNN architecture, VGG-16 trained on ImageNet dataset is used as a feature extractor for the classification task. Experimentation is performed on data collected from Alzheimer's Disease Neuroimaging Initiative (ADNI) database. The accuracy of the 3-way classification using the described method is 95.73% for the validation set.

Keywords: Convolutional Neural Network; Alzheimer; Brain images; Accuracy

1. Introduction

Alzheimer's disease (AD) is the most common form of dementia. Dementia is a general term that describes a set of symptoms associated with a decline in memory or other thinking skills severe enough to reduce a person's ability to perform ordinary activities. Alzheimer's disease accounts for 60–80 percent of cases of dementia. In early stages of disease, also known as mild cognitive impairment (MCI), memory loss is mild, but with late-stage Alzheimer's, the patient loses the ability to even carry on a conversation. According to a recent report (Fargo, 2014), the annual number of new cases of AD and other dementias is projected to get doubled by 2050.

AD, being one such disease for which currently there is no cure or treatment that slows or stops its progression, therefore requires robust and accurate methods for its early diagnosis (Bron et al., 2015). The diagnosis of Alzheimer's disease requires a variety of medical tests which leads to huge amounts of multivariate heterogeneous data. It can be certainly exhausting to manually compare, visualize, and analyse this data due to the heterogeneous nature of medical tests.

The success of deep learning algorithms at visual object recognition tasks and competitions such as ILSVRC

* Corresponding author.
E-mail address: judehemanth@karunya.edu (D.J. Hemanth).

https://doi.org/10.1016/j.cogsys.2018.12.015
1389-0417/

(Russakovsky, 2015) intersects with a time of dramatically increased use of electronic medical records and diagnostic imaging. Although the terms machine learning and deep learning are relatively recent, their ideas have been applied to medical imaging for decades, perhaps particularly in the area of computer aided diagnosis (CAD) and medical imaging applications such as breast tissue classification (Sahiner et al., 1996); Cerebral micro bleeds (CMBs) detection (Dou et al., 2016), Brain image segmentation (Chen, Dou, Yu, Qin, & Heng, in press; Hemanth, Anitha, & Balas, 2016) and classification (Hemanth et al., 2012, 2011), CT liver image segmentation (Zidan, Ghali, Ella Hassanien, Hefny, & Hemanth, 2012).

Deep CNNs are most popular these days amongst computer vision researchers due to their recent performances in certain tasks such as visual object recognition, detection and segmentation. Some of the advantages of CNNs over ANNs (Artificial Neural Networks) are that they operate over volumes, unlike regular neural networks where the input is a vector, here the input is a multi-channelled image (e.g. RGB Image).

CNNs have concept called as parameter sharing i.e. same weights are shared by multiple neurons in a particular feature map. Local connectivity is the concept of each neural connected only to a particular region of image unlike ANNs where all neurons are fully-connected. Basic building blocks of CNNs are discussed in Section 3.1.

Even though CNNs have performed exceptionally well for computer vision tasks in medical field recent couple of years, however training these huge architectures from scratch has few limitations (Tajbakhsh et al., 2016):

(1) Proper training of deep neural networks requires huge amount of annotated data, which can be a problem especially for medical imaging field where it is expensive and sometimes difficult to acquire sufficient data.
(2) Training of such architectures requires huge amount of computational resources.
(3) Deep learning requires careful and tedious tuning of hyper-parameters, optimal tuning, which otherwise can lead to overfitting/underfitting, resulting in overall poor performance.

To resolve these issues, researchers have come up with a successful alternative approach called transfer learning (Yosinski, Clune, Bengio, & Lipson, 2014). Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task.

In practice nowadays, very few people train an entire CNN from scratch (with random initialization), because it is relatively rare to have a dataset of sufficient size. Instead, it is common to pretrain a CNN on a very large dataset (e.g. ImageNet, which contains 1.2 million images with 1000 categories) (CS231n Convolutional Neural Networks for Visual Recognition).

For our 3-way classification problem AD vs MCI vs Cognitively Normal (CN), Classification model is built by performing transfer learning using a state-ofthe-art CNN architecture, VGG16 (Simonyan & Zisserman, 2014). VGG16 is a 16-layer network built by Oxfords Visual Geometry Group (VGG). It participated in the ImageNet competition in ILSVRC 2014. It is one of the first architectures to explore network depth by pushing to 16 layers and using small (3×3) convolution filters.

In this paper, VGG16 is taken as the base model for applying transfer learning. The base model is used as a feature extractor and additional fully-connected layers are added on top of it. The resultant mathematical model obtained $P_F S_E C_{TL}$ is then trained on most informative brain MRI slices keeping the layers of the base model non-trainable.

Our main objective in this paper is to show how instead of training a completely new model from scratch, we can utilize transfer learning approach to build a classification model. Training a CNN which is as big as VGG16 requires huge computational resources and takes weeks to get trained, on the other hand performing transfer learning using same pretrained CNN takes much lesser time, typically few hours. Thus, we use transfer learning and show how a CNN because of its state-of-the-art generalization of local features can be used for medical images, even though medical images are from different domain than the actual images on which CNN is trained. Also, large CNNs require large corpus of data, sometimes in millions for their training while availability of medical images is comparatively lesser which is one more reason we were prompted to use transfer learning. Moreover, we also use only those sMRI slices of subjects that has most amount of information in it by comparing their entropy, hence strengthens the overall robustness of the model. As a result, we have a mathematical model $P_E S_E C_{TL}$ which is capable of correctly distinguishing between 3 different classes of different subjects with promising performance. The highlights of our paper are:

- MRI scans are 3D in nature therefore can also be considered as stack of 2D MRI slices. We can select a set of most informative slices from this stack to do the classification.
- Using CNN eliminates the task of manual feature extraction.
- Through transfer learning we can use a model trained on natural images to classify medical images.
- Transfer learning effectively reduces computational cost and advantageous when data available is in less amount.
- Overfitting of the model is main concern that is taken into account while building the model and dropout regularization is used to avoid it.
- MCI is the most difficult class to classify since it is intermediate stage between AD and CN.

The rest of the paper is organized as follows: Section 2 discusses the related work in the field of Alzheimer's dis-

ease prediction. Section 3 gives the complete explanation of the proposed methodology. Section 3 has been divided into 7 subsections, in which first Section 3.1 discusses the components that are used in building a CNN, Sections 3.3–3.6 discusses the steps followed to prepare the training data for the classification task. In Section 3.3 – the mathematical model is explained with subsequent Section 3.6 discusses how transfer learning has been used to perform classification. After this, Section 4 lists Experimental results, accuracy and loss function plots of our classification model and comparisons with other existing techniques. Finally, Section 5 depicts the final conclusion for the method also emphasizing its application with the future scope of it.

2. Related works

Medical imaging data comes in various modalities such as structural or functional MRI, Diffusion Tensor Imaging (DTI), Positron Emission Tomography (PET), Computed Tomography (CT scan). All these kind of scans have been used in different forms with different machine learning algorithms in the past for early diagnosis or detection of diseases (Arunkumar et al., 2018; Arunkumar et al., 2018; Gupta et al., 2018; Rebouças et al., 2018).

There can be some drawbacks of using neural networks (ANNs) for image classification tasks because for ANN it is usually seen that to yield high accuracy, it requires higher convergence time period as described in Hemanth, Vijila, Selvakumar, and Anitha (2014). In Hemanth et al. (2014), they tackled the problem by proposing two novel neural networks: Modified Counter Propagation Neural Network (MCPN) and Modified Kohenen Neural Network (MKNN) where main concept was to make ANN iteration-free. They analyzed the performances of these networks by classifying abnormal brain MRIs and were able to show promising results.

Majority of early AD diagnosis methods were based on classification of features extracted from brain images, where features are supposed to accurately capture AD related variations of anatomical brain structures. These features are usually then fed to classical machine learning algorithms such as SVM, Random-forest classifier or feed forward networks for classification. Ahmed, Benois-Pineau, Allard, Ben-Amar, and Catheline (2014) extracted two kind of visual features from the hippocampal region: visual local descriptors using the Circulars Harmonic Functions and the amount of CSF pixels in the hippocampal area. Then they first classified CHF-based visual signatures between the categories two by two with a state-of-the art SVM approach with a Radial Basis Function (RBF) kernel and performed classification of subjects on the basis of the CSF volume by a Bayesian classifier. Output of both the classifiers where then fed to a binary SVM classifier to get the final decisive output. Klöppel et al. (2008) used linear SVMs to classify the grey matter segment of T1-weighted MR scans from AD patients and CN elderly individuals obtained from two centers with different scanning equipment. Jongkreangkrai, Vichianin, Tocharoenchai, and Arimura (2016) measured the cerebral image features from T1 weighted images using FreeSurfer and then classified AD patients with NC subjects using SVM classifier. They also compared their performances by using different combinations of extracted features.

The method of Moradi, Pepe, Gaser, Huttunen, and Tohka (2015) consisted of two fundamental stages: a feature selection stage where they selected the most informative voxels (features) using a regularized logistic regression; and a classification stage, where they applied a semi-supervised low density separation (LDS) to produce the final prediction. Lerch et al. (2008) proposed an automated cortical thickness measurement tool to improve the way of clinical diagnosis of probable AD. Cheng, Liu, Shen, Li, and Zhang (2017) used cortical thickness data where they represented it in terms of their spatial frequency components by employing manifold harmonic transform.

Due to more availability of data as well as computational resources, researchers have started inclining towards deep learning for diagnosis in the medical field. Though the amount of medical data available is still less as compared to the requirement of deep neural networks but researchers have been able to resolve this drawback by using various techniques such as transfer learning or pretraining the network. Researchers have used different variants of CNNs such as in Gupta, Ayhan, and Maida (2013), Gupta et al. (2013) first pretrained CNN using sparse autoencoder (SAE) trained on random natural image patches and then used this 2D CNN for slice-wise extraction of features. Payan and Montana (2015) also took a two-stage approach whereby they first initially use a SAE to learn filters for convolution operations by training it on randomly selected 3D patches of brain MRI scans, and then build a 3D CNN whose first layer uses the filters learned with the autoencoder. Hosseini-Asl (2016) did the similar work, where they proposed an AD diagnostic framework that extracts features of a brain MRI with a source-domain-trained 3D-CAE and performs task specific classification with a target-domain-adaptable 3D-CNN. Their model was able to achieve good level of performance as shown in Table 4. Korolev, Safiullin, Belyaev, and Dodonova (2017) used network architectures similar to VGG16 and Residual Neural Network (He, Zhang, Ren, & Sun, 2015) for building their classification models and then trained the model on 3D brain MRI scans collected from ADNI database.

Cho, Seong, Jeong, and Shin (2012) proposed Multi-Domain Transfer learning (MDTL) framework which contains two key components: a multi-domain transfer feature selection (MDTFS) for selecting most informative feature subsets and multi-domain transfer classification (MDTC) for classification.

Glozman and Liba (2016) used transfer learning approach where they fine-tuned AlexNet (Krizhevsky, Sutskever, & Hinton, 2012) architecture trained on ImageNet dataset and also did data augmentation such as mir-

ror transformation on the data collected from ADNI database to increase the amount of data for training. Billones, Jan, Demetria, Hostallero, and Naval (2016) proposed DemNet architecture which is a modified version of the 16-layer CNN, VGG16. Key attribute of their work was that they trained their network on 20 randomly selected MRI slices of each subject and were able to outperform several classifiers. Hon and Khan (2017) did similar work but used a more intelligent way of selecting slices where they used entropy based mechanism to select the data rather than selecting data by intuition. Then they used transfer learning on two architectures: VGG16 and Inception V4 (Szegedy, Ioffe, Vanhoucke, & Alemi, 2017) and compared performances of both the architectures. They were able to achieve 96.25% accuracy at binary classification by Inception V4 architecture on OASIS dataset.

3. Methodology

Deep learning is a subfield of machine learning and a collection of algorithms that are inspired by the structure of human brain and try to imitate the functions of human brain, which is the reason these algorithms are most of the times also termed as "neural networks". These algorithms are called "deep" as the input passes through series of non-linear transformations before it becomes output. Convolution neural network (CNN) is one such deep learning algorithm in which the transformations are done using an operation called "convolution". Before, going through the overall methodology, we discuss the building blocks of CNN in the following section.

3.1. Building components of CNN

Convolution neural networks (CNNs) were first introduced by Lecun, Bottou, Bengio, and Haffner (1998). CNNs are specialized class of neural network for processing data that has a known grid-like topology. They use convolution operation in place of general matrix multiplication in at least one of their layers (Goodfellow, Bengio, & Courville, 2016).

Core operations and layers that are used in building a CNN are as follows:

3.1.1. Convolution operation

The Convolution layer is the core building block of a CNN that does most of the computational heavy lifting. Its parameters consist sets of learnable filters or kernels.

Convolution operations done on an image of size $h \times w$, with a kernel size of k, stride size s, and padding p, produces an output of size $\frac{(h-k+2p)}{s+1} \times \frac{(w-k+2p)}{s+1}$. The kernels act as feature detectors, convolved with the image, thereby producing a set of convolved features. In the neural network, the kernel size indicates the receptive field of a neuron, thus enforcing local connectivity of the neurons to the previous volume.

Output Z obtained from mathematical convolution operation between matrix X of size (P, Q) and matrix Y of size (R, S) can be expressed as:

$$Z(i,j) = \sum_{p=0}^{P-1} \sum_{q=0}^{Q-1} X(p,q) * Y(i-p, j-q) \tag{1}$$

where $0 \le i \le P+R-1$ and $0 \le j \le Q+S-1$.

For computing Z (0, 0) according to Eq. (1) Y is first rotated by 180° about its centre element and its centre is slide so that it lies on the top of $X(0, 0)$. After this, each element of the rotated Y is multiplied by element of X underneath it. For $Z(0, 0)$, all the individual products obtained are summed together.

While in case of CNN's convolution operation, first step of rotating Y by 180° about its centre is generally omitted and all other steps are same as described above.

This operation which does not consider flipping of matrix Y is more commonly known as cross-correlation operation and can be expressed mathematically for matrices X, Y, Z defined above as:

$$Z(i,j) = \sum_{p=0}^{P-1} \sum_{q=0}^{Q-1} X(p,q)\, \bar{Y}(p-i, q-j) \tag{2}$$

where $-(R-1) \le i \le P-1, -(S-1) \le j \le Q-1$ and bar over Y denotes complex conjugation.

3.1.2. Max-pooling operation

Max-pooling is an aggregation operation that extracts the maximum value in a region of size $h \times w$ on image of size $h \times w$, specified by kernel of size k and stride size s. The operation produces an output of size $\frac{(h-k)}{s+1} \times \frac{(w-k)}{s+1}$.

Main reason for inserting a layer with max-pooling operation in between the successive convolutional layers is to progressively reduce the size of spatial representation i.e. values of h and w so that number of parameters to be trained are lesser and overall computations in the network are reduced. Doing this also helps in controlling overfitting. Most common value for k as well as s is 2 which downsamples h and w by factor of 2.

3.1.3. Dropout regularization

The term "dropout" refers to dropping out neurons (both hidden and visible) in a neural network randomly. This technique was introduced by Srivastava, Hinton, Krizhevsky, and Salakhutdinov (2014) to mainly tackle the problem of overfitting in neural networks. Dropout in neural network sets the output of certain portion of neurons in a hidden layer, depicted by dropout ratio to 0. Dropout ratio is nothing but probability for a neuron in a particular layer to get dropped out. Hence, if it is set to 1 for a hidden layer, then all the neurons in that particular hidden layer will output 0. Neurons that are dropped out does not contribute to the forward pass and backward propagation steps. In this way, neural network samples a

different architecture at each forward-backward propagation steps, but all of these still share parameters.

3.1.4. Non-linearity layers

Generally convolutional layers are followed by non-linearity operations or functions, which are also called activation functions. In the earlier time, sigmoid and tanh were the most commonly used activation functions. But, due to certain drawbacks, researchers have proposed another activation functions such as rectified linear unit (ReLU) (Nair & Hinton, 2010) and its variants (leaky ReLU, Noisy ReLU, ELU) which are now preferred in most of the deep learning tasks. ReLU function and its variants are expressed mathematically as:

$$f(x) = \max(0, x) \tag{3}$$

$$f(x) = \begin{cases} x, x > 0 \\ 0.01x, otherwise \end{cases} \tag{4}$$

$$f(x) = \begin{cases} x, x \geq 0 \\ a(e^x - 1), otherwise \end{cases} \tag{5}$$

Here Eq. (3) is for Simple ReLU, Eq. (4) is for Leaky ReLU and Eq. (5) for Exponential LU (ELU). In Eq. (5) a is a hyper-parameter than can be tuned and $a \geq 0$.

3.1.5. Fully-connected layers

Neurons in a fully connected layer have full connections to all activations in the previous layer, as seen in regular neural networks (ANNs). Fully-connected layers are usually appended after a combination of convolution, max-pooling and dropout layers in CNN for multi-class or binary classification. Fig. 1 shows the VGG16 architecture that was built for ILSVRC 2014.

3.2. Data collection

Data used was obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI). The ADNI was launched in 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. The primary goal of ADNI has been to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's disease (AD).

T1-weighted sMRI data of 150 subjects: 50 AD; 50 CN; 50 MCI was selected for the classification task. Demographic characteristics of selected subjects including Age, gender and MMSE score are summarized in Table 1.

Description for all the MR images is magnetization-prepared 180° radio-frequency pulses and rapid gradient-echo (MP RAGE). All the images were downloaded in Neuroimaging Informatics Technology Initiative (NIfTI) file format.

3.3. Alzeihmer classification mathematical model – $P_F S_E C_{TL}$

In this paper, we develop a mathematical model for AD. Based on Fig. 4, this model is presented in three phases:

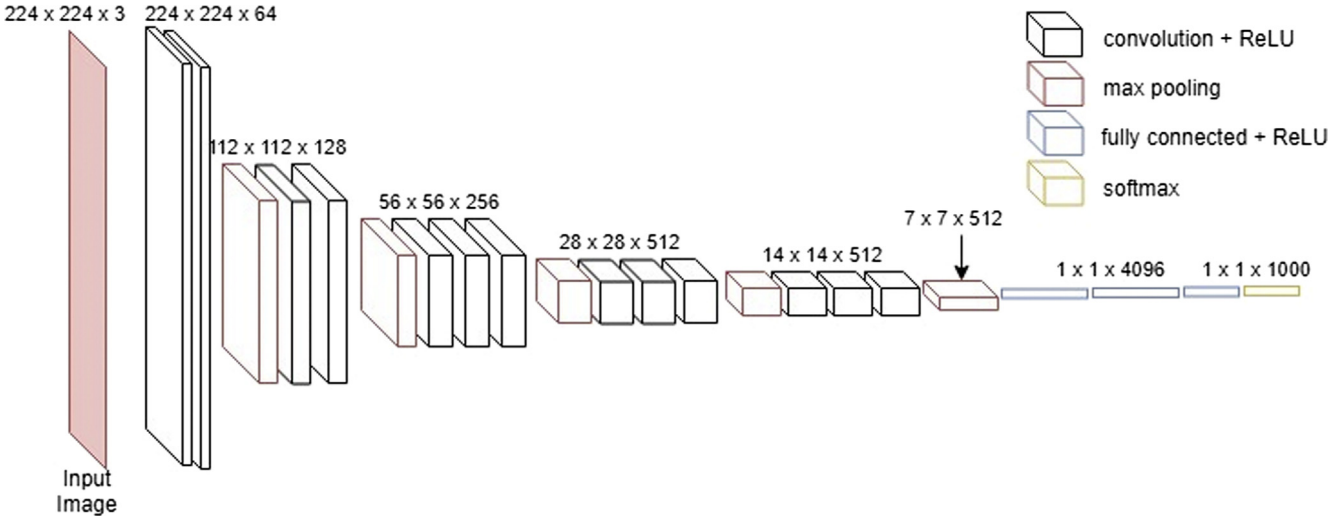

Fig. 1. VGG16 neural network architecture for ILSVRC 2014.

Table 1
Demographic characteristics of selected subjects.

Class	No. of males	No. of females	Age range	Average age	MMSE score range	Average MMSE score
AD	26	24	58.25–88.92	76.98	19–26	22.30
CN	33	17	57.30–70.95	67.18	25–29	27.94
MCI	29	21	58.31–83.55	63.8	25–29	27.16

Data Pre-processing using Freesurfer, Selection based on Entropy followed by further classification using transfer learning.

- Model parameters:
  - $outVol(I)$ which gives the ouput Volume for correcting motions between image pixels
  - $I(x), U(x)$ as the initial and uncorrupted Image represented as function of x pixel
  - $\widehat{f}_e^n$ as bias f with n noise and e error
  - X, Y, Z representing the 3 point coordinates of image pixel
  - E:entropy, S: RootMean Square Upgradation parameter with W weights

3.3.1. Data pre-processing using FreeSurfer (P_F)

To remove unnecessary details of brain MR images that might cause poor training of our classification model, cortical reconstruction and volumetric segmentation was performed with the FreeSurfer image analysis suite, which is documented and freely available for download online (FreeSurfer). Specifically, recon -all autorecon1 was used, that performs only 5 out of 31 transformation processes of recon-all. Outputs corresponding to the 5 processes are shown in Fig. 2. These 5 processes are:

(1) Motion Correction and conform: When there are multiple sources of volume, this process will correct for minor motions between them and will average them together.

Let $inVol_1\ inVol_2 \ldots inVol_n$ be the different input volumes of Image I, then after motion correction the new output Volume received will be

$$outVol(I) = inVol(I)_1 + inVol(I)_2 \cdots inVol(I)_n/n \tag{6}$$

(2) Non-Uniform intensity normalization (NU): Also called N3, corrects MR data by removing non-uniformity in intensity of the image.

The following equation describes the image formation model

$$I(x) = U(x)f(x) + n(x) \tag{7}$$

where I is the given image, U is the uncorrupted image, f is the bias field, and n is the noise.

Using formula $\widehat{v} = \log v$ in a noise free scenario, Eq. (7) becomes.

$$\widehat{I(x)} = \widehat{U(x)} + \widehat{f(x)} \tag{8}$$

$$\widehat{U}^n = \widehat{I} - \widehat{f}_e^n \tag{9}$$

$$= \widehat{I} - S\left\{\widehat{I} - E\left[\widehat{u}|\widehat{u}^{n-1}\right]\right\}$$

where $\widehat{u}^0 = \widehat{v}$, $\widehat{f}_e^0$ (is typically set to 0) and the smoothing operator, $S\{\cdot\}$, is a B-spline approximator.

(3) Talairach transform computation: This processing step computes the affine transform through a FreeSurfer script called talairach.

Initially the pixel coordinate are converted to Talairach coordinates using the following equations

$$X' = 0.88X - 0.8 \tag{10}$$

$$Y' = 0.97Y - 3.32 \tag{11}$$

$$Z' = 0.05Y + 0.88Z - 0.44 \tag{12}$$

Finally an affine transformation is applied to the above obtained coordinates

(4) Intensity normalization: This steps corrects for fluctuations in intensity. It scales intensities for all voxels such that mean intensity of white matter is 110.

(5) Skull stripping: This step removes the skull from the normalized image.

3.4. Selecting most informative MRI slices based on entropy (S_E)

Brain MR images are in NIfTI format. NIfTI images are volumetric (3D) images, therefore images that we have after pre-processing are all of size $256 \times 256 \times 256$. These images comprise of 2D images called slices. Hence, we have 256 slices corresponding to each NIfTI image.

Even though we can use all 256 slices corresponding to each of the 150 subjects for training model, but choosing the best possible data can certainly improve the chances of success of model. In recent methods (Billones et al., 2016), a set of slices are extracted at random by assuming that these slices contain most relevant information. Instead of extracting slices at random, image entropy based sorting mechanism is used to take most informative slices in which image entropy for each slice was calculated and top 32

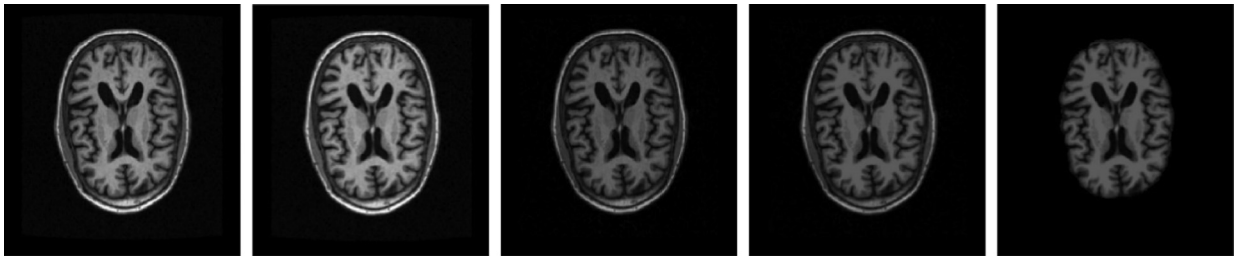

Fig. 2. Transformation of a slice of one of the sample from our dataset after going through different pre-processing steps of FreeSurfer in order from left-to-right.

slices based on entropy value were selected of each subject and rest of the slices were discarded, similar to the approach described in (Hon & Khan, 2017).

In general, for a set of M symbols with probabilities p_1, p_2,. . ., p_i the entropy E is calculated as follows:

$$E = -\sum_{i=1}^{M} p_i \log p_i \tag{13}$$

Above steps of data processing results in a balanced dataset of 4800 (150 subjects × 32 slices corresponding to each subject) slices which contains 1600 CE, 1600 MCI, and 1600 CN slices.

VGG16 was trained on RGB i.e. 3 channel images of size 224 × 224 × 3 and hence accepts input only if it has exactly 3 channels. Also width and height of image should be no smaller than 48. Therefore, images were also cropped and converted to 200 × 200 × 3 size which would be a valid input size for the model. Fig. 3 shows some of the image samples from our resulting dataset.

3.5. Creating training and test sets

Our balanced dataset of 4800 images is shuffled and split into training and test set with split ratio 80:20. Resulting training and test set sizes for 3-way classification (AD vs CN vs MCI) and 2-way classification (AD vs CN, AD vs MCI and CN vs MCI) are summarized in Table 2(a) and (b).

3.6. Classification using transfer learning (C_{TL})

For building the classification model, we have used CNN in the form of a feature extractor i.e. the CNN architecture, VGG16 pretrained on ImageNet dataset is taken as base model through which a brain MRI slice can be passed to extract the feature values belonging to that particular MRI slice.

To make use of the base model for our classification task fully-connected layers from the base model are removed

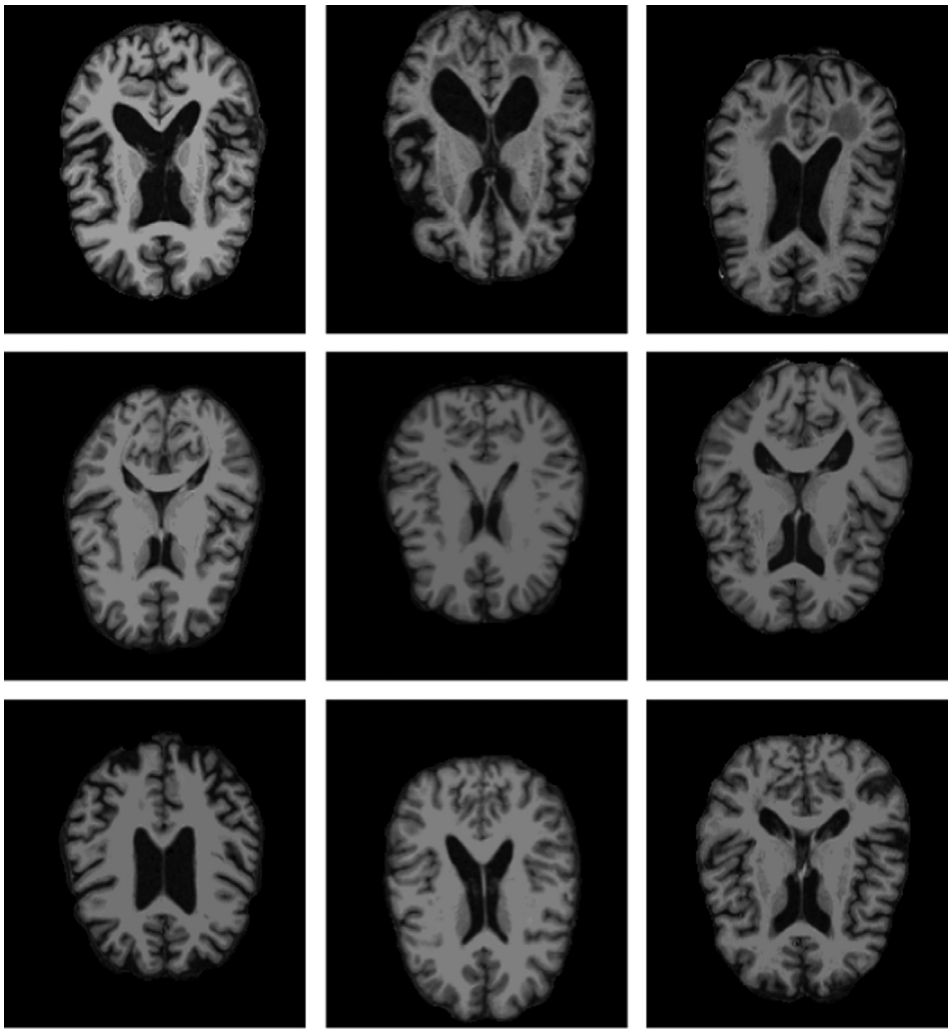

Fig. 3. Image samples from our dataset of class AD, CN and MCI.

Table 2
Training set and test set sizes.

Class label	Training set size	Test set size
(a) For 3-way classification		
0 (AD)	1280	320
1 (CN)	1280	320
2 (MCI)	1280	320
Total	3840	960
(b) For binary classification		
0 (AD or CN)	1280	320
1 (CN or MCI)	1280	320
Total	2560	640

since the outputs of those layers are 1000 class scores for classification task on ImageNet. After removing the later fully-connected layers of the model, output of the last convolutional layer is flattened into one column vector of size 18,432, and new fully-connected layers are added at the end of base model in which first layer consists of 256 neurons, second layer is the dropout layer with dropout ratio 0.5 which means half of the neurons will output 0 at any instance and at the end of the model is the softmax layer (output layer) whose output is 3 class scores for 3-way classification and 2 class scores for binary classification tasks. Fig. 4 shows the resultant classification model for 3-way classification.

Softmax is an activation function which outputs a value between 0 and 1 similar to sigmoid function, generally used when number of classes is more than 2 and is defined as:

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=0}^{k} e^{x_j}} \tag{14}$$

For training the model, categorical cross-entropy loss function is used. Cross-entropy, or log loss function measures the performance of a classification model whose output (class score) is a probability value between 0 and 1. Cross-entropy loss increases as the predicted output diverges from actual label. If number of classes is 2, binary cross-entropy loss is calculated as:

$$L(y,p) = -(y\log p + (1-y)\log(1-p)) \tag{15}$$

Otherwise, if number of classes >2, categorical cross-entropy loss is calculated as:

$$L(y,p) = -\sum_{c=1}^{M} y_{o,c}\log(p_{o,c}) \tag{16}$$

where M is the number of classes, y is the actual value and p is the predicted value.

Optimization algorithm used for updating parameter is RMSprop (Root Mean Square Propagation) with learning rate set to 0.0001. RMSprop is a gradient descent based adaptive learning method proposed by Geoffrey Hinton in one of his lectures (Rmsprop). For instance, parameter W (weights) corresponding to a particular neuron of a neural network can be updated using RMSprop as:

$$S = \beta S + (1-\beta)dW^2 \tag{17}$$

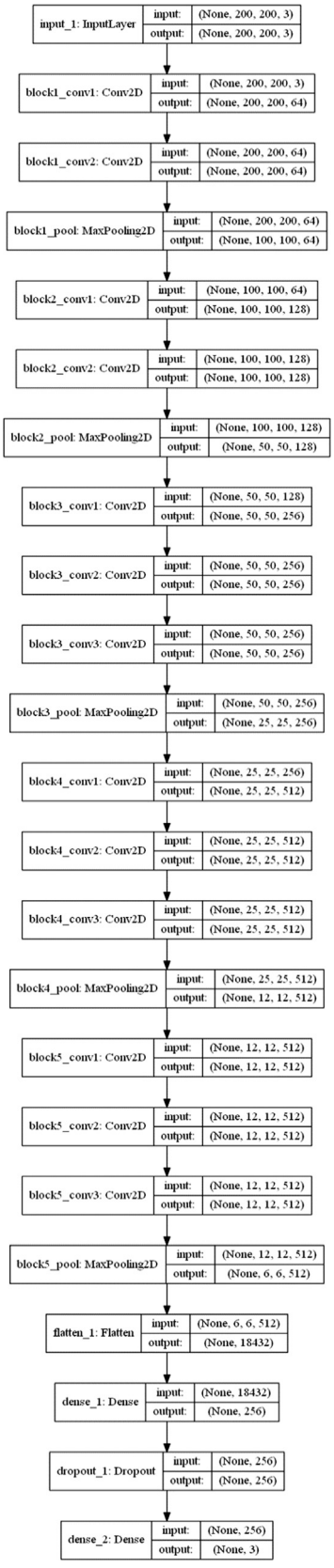

Fig. 4. Classification model for 3-way classification.

$$W = W - \alpha\frac{dW}{\sqrt{S}} \tag{18}$$

where $0 \leq \beta \leq 1$, $dW = \frac{dL}{dW}$ and α is the learning rate.

4. Experimental results

The classification model was built using Keras (Chollet, 2015), high-level neural networks API, written in Python with Tensorflow (Abadi, Agarwal, Barham, Brevdo, Chen, Citro, & Devin, 2016), an open source software library as backend. We chose Keras since it allows easy and fast prototyping and runs seamlessly on GPU.

Model for 3-way classification was fitted on training data of size 3840 in batch size of 40 in 50 epochs. 1 Epoch is defined as completed when entire dataset is passed forward and backward through neural network. In our case it took 96 steps for completion of 1 epoch. Parameter values are updated at every step of epoch. It took around 10 h for training our classification model on training set along with validation on test set at each epoch on NVIDIA Quadro K1200 GPU.

Since accuracy have been used as the primary evaluation metric, we set the parameter "metrics" for evaluation as "accuracy" in the compilation step. In Keras, accuracy is calculated as:

$$Accuracy = \frac{\sum_{i=1}^{n} B^i}{n} \tag{19}$$

where n is the number of samples and B^i is a Boolean function for i_{th} sample whose true class label is y^i_{true} and predicted class label is $y^i_{predicted}$. It is calculated as:

$$B^i = \begin{cases} 0, y^i_{true} \neq y^i_{predicted} \\ 1, y^i_{true} = y^i_{predicted} \end{cases} \tag{20}$$

In our approach, resulting accuracy for test (validation) set came out to be 95.73%.

Furthermore, 3 more separate models were trained on training data of 2560 slices (training and test set sizes for different classes are described in Table 2) to do binary classifications amongst 3 classes: AD vs CN, AD vs MCI and CN vs MCI.

These models were also trained for 50 epochs keeping the same optimizer and loss function. For binary classifications, models were able to achieve accuracy of 99.14%, 99.30% and 99.22% for AD vs CN, AD vs MCI and MCI vs CN classifications respectively.

Confusion matrices were also computed to describe the performance of all the four type of classifications on validation/test data. Table 3 describes the confusion matrices. For further evaluation of classification models, we also computed performance metrics such as Precision, Recall and F1-Score with the help of confusion matrices described in Table 3. These metrics are be calculated as:

$$Precision = \frac{TP}{TP + FP} \tag{21}$$

$$Recall = \frac{TP}{TP + FN} \tag{22}$$

Table 3
Confusion matrices for different type of classifications.

(a) For 3-way classification

		Predicted label		
		AD	CN	MCI
Actual Label	AD	290	3	27
	CN	0	310	10
	MCI	0	1	319

(b) For AD vs CN classification

		Predicted label	
		AD	CN
Actual Label	AD	316	4
	CN	1	319

(c) For AD vs MCI classification

		Predicted label	
		AD	MCI
Actual Label	AD	316	4
	MCI	0	320

(d) For CN vs MCI classification

		Predicted label	
		CN	MCI
Actual Label	CN	317	3
	MCI	2	318

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{23}$$

where TP, FP and FN are True positive, False positive and False negative predictions corresponding to particular class label. Table 4 gives overall classification report for all the four classification types.

We also plotted the accuracy on training and validation/test sets for 50 epochs using matplotlib library in python. Fig. 5 shows the accuracy plots and Fig. 6 shows the loss function plots for 3-way as well as binary classifications. Loss values were calculated using the cross-entropy loss functions described in Eqs. (8) and (9). Table 5 shows the comparison of our method with past methods that were described in Section 2 (see Fig. 7).

Table 4
Precision, recall and F1-score values for different type of classifications.

Classification type	Class label	Precision	Recall	F1-score
3-way	AD	1	0.91	0.95
	CN	0.99	0.97	0.98
	MCI	0.90	1	0.94
AD vs CN	AD	1	0.99	0.99
	CN	0.99	1	0.99
AD vs MCI	AD	1	0.99	0.99
	MCI	0.99	1	0.99
CN vs MCI	CN	1	0.99	0.99
	MCI	0.99	1	0.99

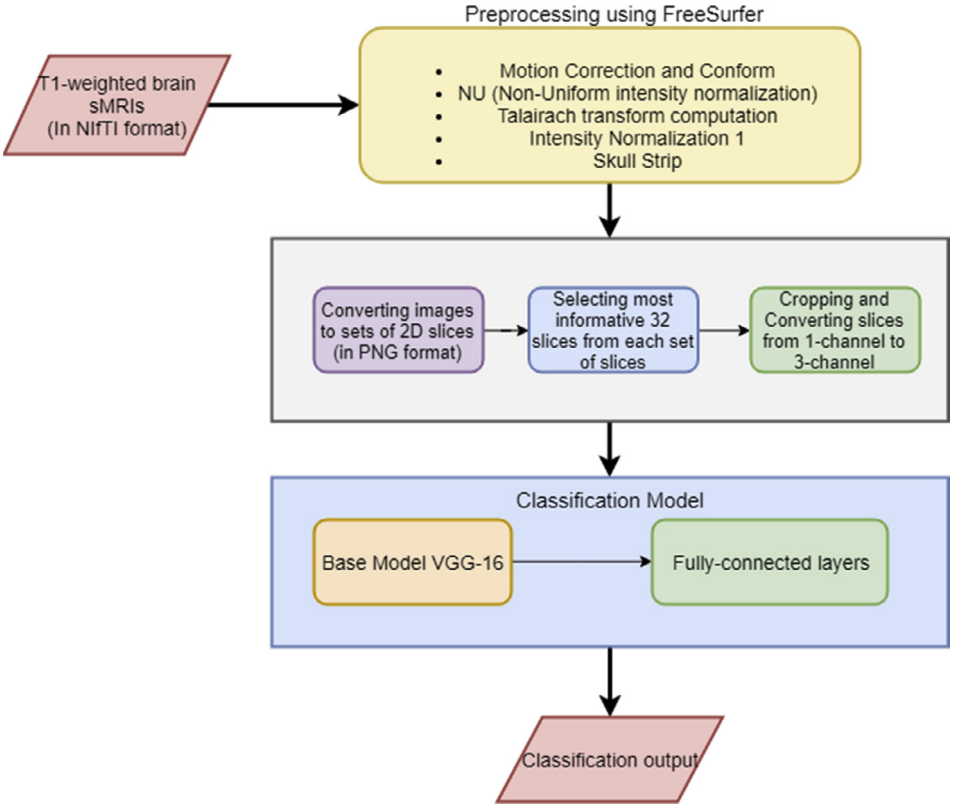

Fig. 5. Overall scheme of the method used.

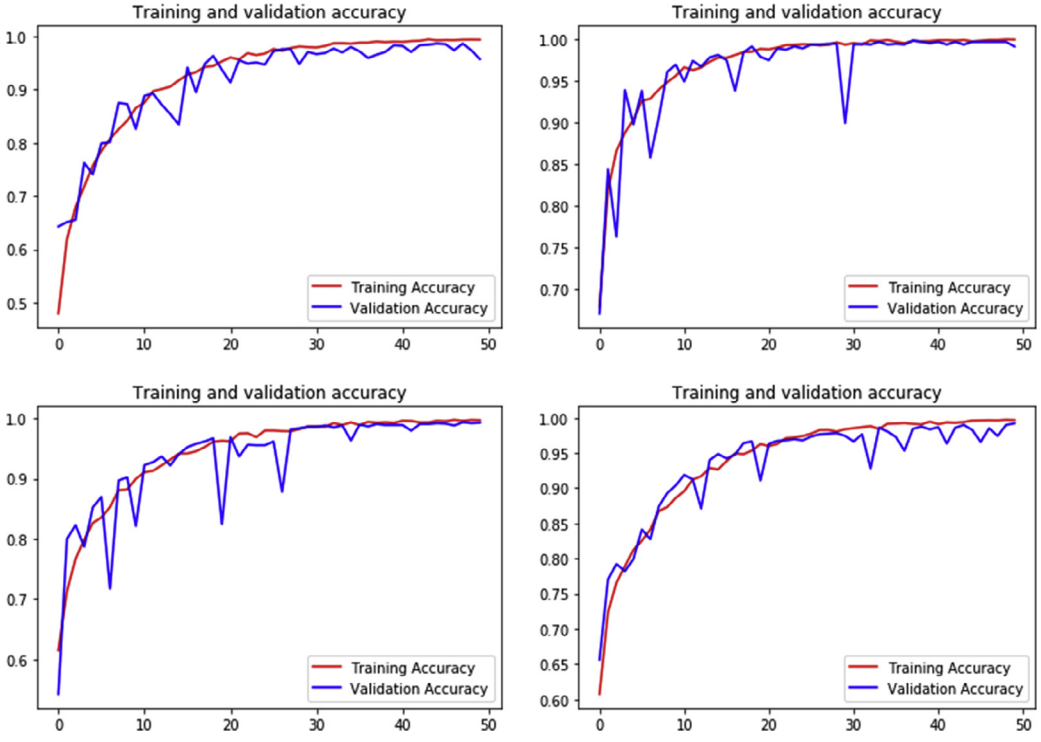

Fig. 6. Accuracy plots for 3-way (top-left), AD vs CN (top-right), AD vs MCI (bottom-left) and CN vs MCI (bottom-right).

Table 5
Comparison with past methods.

(a)

Approach	Classification type			
	3-way	AD vs CN	AD vs MCI	CN vs MCI
Gupta et al. (2013)	85	94.7	88.1	86.3
Payan and Montana (2015)	89.47	95.39	86.84	92.11
Hosseini-Asl (2016)	94.8	99.3	100	94.2
Korolev et al. (2017)	–	80	–	58
Glozman and Liba (2016)	60.66	83.57	–	–
Billones et al. (2016)	91.85	98.33	93.89	91.67
Hon and Khan (2017)	–	96.25	–	–
Our approach	95.73	99.14	99.30	99.22

(b)

Approach	Type of CNN (2D or 3D)	Modalities	Data (Number of subjects)
Gupta et al. (2013)	2D CNN	sMRI	ADNI (843)
Payan and Montana (2015)	Both	sMRI	ADNI (2265)
Hosseini-Asl (2016)	3D CNN	sMRI	ADNI (210) + CADDementia (30)
Korolev et al. (2017)	3D CNN	sMRI	ADNI (231)
Glozman and Liba (2016)	2D CNN	sMRI + PET	ADNI (1370)
Billones et al. (2016)	2D CNN	sMRI	ADNI (900)
Hon and Khan (2017)	2D CNN	sMRI	OASIS (200)
Our approach	2D CNN	sMRI	ADNI (150)

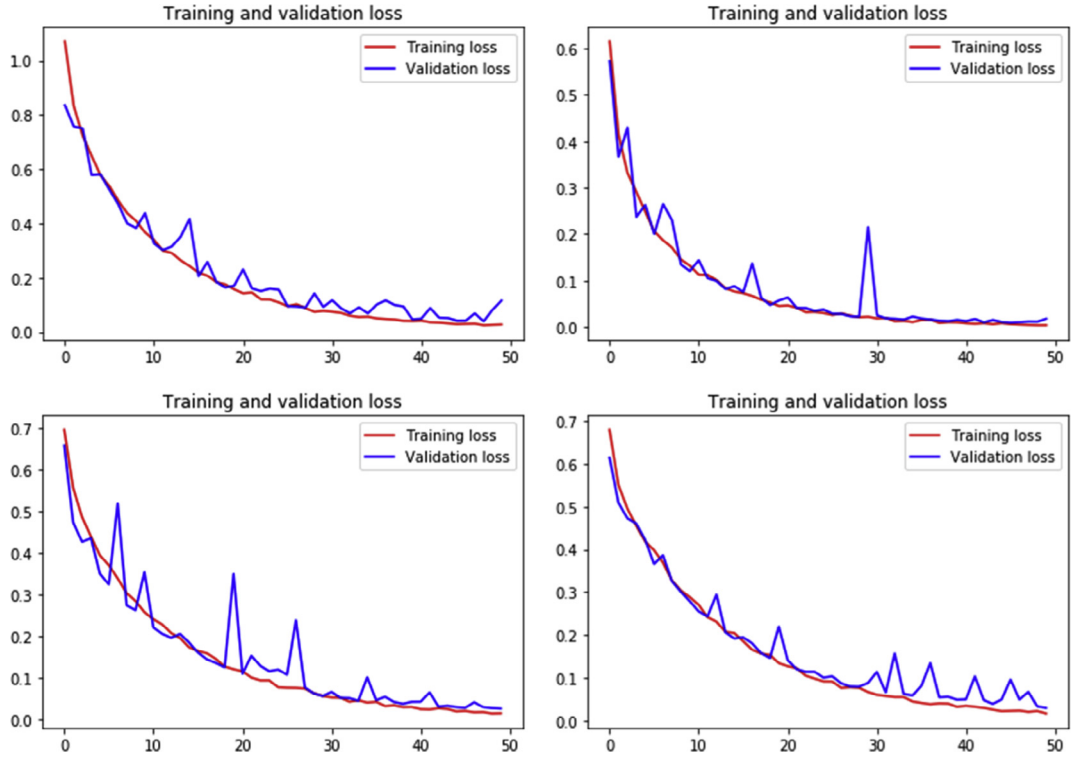

Fig. 7. Loss function plots for 3-way (top-left), AD vs CN (top-right), AD vs MCI (bottom-left) and CN vs MCI (bottom-right).

5. Conclusion and future prospectives

In this paper, we proposed a transfer learning approach for accurately classifying brain sMRI slices amongst 3 different classes: AD, CN and MCI. Here, we used a pre-trained VGG16 network for transfer learning and used it as a feature extractor. We demonstrated that even though VGG16 was trained on very general images of ImageNet dataset (natural images), it was still able to extract useful features for our classification task. We also computed various metrics to support the performance of our classification model and compared accuracy of our method with past methods in Section 4.

For future studies, researchers should try other neural networks such as Inception Network, Residual Network and more recent state-of-the-art networks as base model

for building the classifier. One should also try to achieve similar or better results by skipping preprocessing steps such as skull striping, intensity normalization. Moreover, overall performance can also be improved by fine-tuning i.e. training the pretrained convolutional layers of base model, provided the data used is in sufficient amount and available resources can handle the increased computational complexity.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Devin, M. TensorFlow: large-scale machine learning on heterogeneous distributed systems. 2016.

ADNI|Alzheimer's Disease Neuroimaging Initiative. [Online]. Available: <http://adni.loni.usc.edu/>.

Ahmed, O. B, Benois-Pineau, J., Allard, M., Ben-Amar, C., Catheline, G. (2014). Classification of Alzheimer's disease subjects from MRI using hippocampal visual features. Multimedia Tools and Applications, Springer Verlag, pp. 35. <hal-00993379>.

Arunkumar, N., Mohammed, M. A., Ghani, M. K. A., Ibrahim, D. A., Abdulhay, E., Ramirez-Gonzalez, G., & de Albuquerque, V. H. C. (2018). K-Means clustering and neural network for object detecting and identifying abnormality of brain tumor. *Soft Computing*.

Arunkumar, N., Mohammed, M. A., Mostafa, S. A., Ibrahim, D. A., Rodrigues, J. J. P. C., & de Albuquerque, V. H. C. (2018). Fully automatic model-based segmentation and classification approach for MRI brain tumor using artificial neural networks. *Concurrency and Computation: Practice and Experience* e4962.

Billones, C. D., Jan, O., Demetria, L. D., Hostallero, D. E. D., Naval, P. C. (2016). DemNet: a convolutional neural network for the detection of Alzheimer's disease and mild cognitive impairment. p. 3728–3731.

Bron, E. E., Smits, M., Van Der Flier, W. M., Vrenken, H., Barkhof, F., Scheltens, P., ... Pinto, M. (2015). Standardized evaluation of algorithms for computer-aided diagnosis of dementia based on structural MRI: The CADDementia challenge. *NeuroImage*.

Chen, H., Dou, Q., Yu, L., Qin, J., & Heng, P.-A. (in press). VoxResNet: deep voxelwise residual networks for brain segmentation from 3D MR images. *NeuroImage*, In press.

Cheng, B., Liu, M., Shen, D., Li, Z., & Zhang, D.the A. D. N. Initiative. (2017). Multi-domain transfer learning for early diagnosis of Alzheimer's disease. *Neuroinformatics, 15*(2), 115.

Cho, Y., Seong, J.-K., Jeong, Y., & Shin, S. Y. (2012). Individual subject classification for Alzheimer's disease based on incremental learning using a spatial frequency representation of cortical thickness data. *NeuroImage, 59*(3), 2217–2230. https://doi.org/10.1016/j.neuroimage.2011.09.085.

Chollet, F. (2015). Keras. [Online]. Available: <https://keras.io/>.

CS231n Convolutional Neural Networks for Visual Recognition. [Online]. Available: <http://cs231n.github.io/transfer-learning/>.

Dou, Q., Chen, H., Yu, L., Zhao, L., Qin, J., Wang, D., et al. (2016). Automatic detection of cerebral microbleeds from MR images via 3D convolutional neural networks. *IEEE Transactions on Medical Imaging, 35*(5), 1182–1195.

Fargo, K. (2014). Alzheimer's Association Report: 2014 Alzheimers disease facts and figures. *Alzheimer's Dement, 10*(2), e47–e92.

FreeSurfer. [Online]. Available: <http://surfer.nmr.mgh.harvard.edu/>.

Glozman, T. & Liba, O. (2016). Hidden cues: Deep learning for Alzheimer's disease classification CS331B project final report. No. 1.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Gupta, D., Julka, A., Jain, S., Aggarwal, T., Khanna, A., Arunkumar, N., & de Albuquerque, V. H. C. (2018). Optimized cuttlefish algorithm for diagnosis of Parkinson's disease. *Cognitive Systems Research, 52*, 36–48.

Gupta, A., Ayhan, M., & Maida, A. (2013). Natural image bases to represent neuroimaging data. In Proceedings of the 30th International Conference on Machine Learning (ICML-13) (pp. 987–994, vol. 28, no. 3). JMLR Workshop and Conference Proceedings, May 2013.

He, K., Zhang, X., Ren, S., Sun, J. (2015). Deep residual learning for image recognition, arXiv preprint arXiv: 1512.03385.

Hemanth, D. J., Anitha, J., & Balas, V. E. (2016). Fast and accurate fuzzy C-means algorithm for MR brain image segmentation. *International Journal of Imaging Systems and Technology, 26*(3), 188–195.

Hemanth, D. J., Selvathi, D., & Anitha, J., 2012. Application of adaptive resonance theory neural network for MR brain tumor image classification.

Hemanth, D. J., Vijila, C. K. S., & Anitha, J. (2011). Application of Neuro-fuzzy model for MR brain tumor image classification. *International Journal of Biomedical Soft Computing and Human Sciences: the official journal of the Biomedical Fuzzy Systems Association, 16*(1), 95–102.

Hemanth, D. Jude, Vijila, C. K. S., Selvakumar, A. I., & Anitha, J. (2014). Performance improved iteration-free artificial neural networks for abnormal magnetic resonance brain image classification. *Neurocomputing, 130*, 98–107.

Hon, M. & Khan, N. (2017). Towards Alzheimer's disease classification through transfer learning.

Hosseini-Asl, E., Gimel'farb, G., & El-Baz, A. (2016) Alzheimer's disease diagnostics by a deeply supervised adaptable 3D convolutional network. No. 502.

Jongkreangkrai, C., Vichianin, Y., Tocharoenchai, C., & Arimura, H. (2016). Computer-aided classification of Alzheimer's disease based on support vector machine with combination of cerebral image features in MRI. *Journal of Physics: Conference Series, 694*(1).

Klöppel, S., Stonnington, C., Chu, C., Draganski, B., Scahill, R., Rohrer, J., Fox, N., Jack, C., Ashburner, J., & Frackowiak, R. (2008). Automatic classification of MR scans in Alzheimer's disease. *Brain, 131*(3), 681–689.

Korolev, S., Safiullin, A., Belyaev, M., & Dodonova, Y. (2017). Residual and plain convolutional neural networks for 3D brain MRI classification. In Proceedings of international symposium on biomedical imaging (pp. 835–838).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 1097–1105.

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of IEEE, 86*(11), 2278–2324.

Lerch, J. P., Pruessner, J., Zijdenbos, A. P., Collins, D. L., Teipel, S. J., Hampel, H., & Evans, A. C. (2008). Automated cortical thickness measurements from MRI can accurately separate Alzheimer's patients from normal elderly controls. *Neurobiology of Aging, 29*(1), 23–30.

Moradi, E., Pepe, A., Gaser, C., Huttunen, H., & Tohka, J. (2015). Machine learning framework for early MRI-based Alzheimer's conversion prediction in MCI subjects. *Neuroimage, 104*, 398–412.

Nair, V. & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In J. Fürnkranz and T. Joachims (Eds.) Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML'10) (pp. 807–814). USA: Omnipress.

Payan, A. & Montana, G. (2015). Predicting Alzheimer's disease: a neuroimaging study with 3D convolutional neural networks. pp. 1–9.

Rebouças, E. de S., Marques, R. C. P., Braga, A. M., Oliveira, S. A. F., de Albuquerque, V. H. C., & Rebouças Filho, P. P. (2018). New level set approach based on Parzen estimation for stroke segmentation in skull CT images. *Soft Computing*.

Rmsprop: Divide the gradient by a running average of its recent magnitude – Optimization: How to make the learning go faster | Coursera. [Online]. Available: <https://www.coursera.org/lecture/neural-networks/rmsprop-divide-the-gradient-by-a-running-average-of-its-recent-magnitude-YQHki>.

Russakovsky, O. et al. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision, 115*(3), 211–252.

Sahiner, B., Chan, H.-P., Petrick, N., Wei, D., Helvie, M. A., Adler, D. D., et al. (1996). Classification of mass and normal breast tissue: A convolution neural network classifier with spatial domain and texture images. *IEEE Transactions on Medical Imaging, 15*(5), 598–610.

Simonyan, K. & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition, pp. 1–14.

Srivastava, N., Hinton, G., Krizhevsky, A. & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting.

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. *AAAI*, 4278–4284.

Tajbakhsh, N., Shin, J. Y., Gurudu, S. R., Hurst, R. T., Kendall, C. B., Gotway, M. B., & Liang, J. (2016). Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE Transactions on Medical Imaging, 35*(5), 1299–1312.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, 27.

Zidan, A., Ghali, N. I., Ella Hassanien, A., Hefny, H., & Hemanth, J. (2012). Level set-based CT liver computer aided diagnosis system.