IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

3D CNN design for the classification of Alzheimer's disease using brain MRI and PET

Bijen Khagi[1], Goo Rak Kwon[2]

[1, 2] Department of Information and Communication Engineering, Gwangju, South Korea.

Corresponding author: Goo R. Kwon (email: grkwon@chosun.ac.kr)

ABSTRACT Attempt to diagnose Alzheimer's disease (AD) using imaging modalities is one of the scopes of deep learning. While considering the theoretical background from past studies, we are trying to identify convolutional neural network (CNN) behaviors moving from 2D to 3D architecture. This study aims to explore the output from a variety of CNN models implemented in the MRI or/and PET classification tasks for AD prediction while trying to summarize its characteristics with a variety of parameters that are tuned and changed. There are many architectures available; however, we are testing a basic architecture with a change in the reception area based on the convolutional layer's kernel size and its strides. The architecture has been categorized as converging, diverging, or equivalent if the filter kernel size is unchanged. This investigation studies a simple non-recurrent encoder based CNN with a sequential flow of features from low-level to high-level feature extraction. The idea is to present a diverging reception area by increasing the filter size and stride from a lower to a higher level. As a result, the feature redundancy is reduced and the trivial features keep on diminishing. The proposed architecture is referred to as 'divNet', and several experiments were performed to determine how effective the architecture is in terms of the consumed memory, the number of parameters, running time, classification error, and the generalization error. This study surveys several related experiments by changing the hyper-parameters setting, the architecture selection based on the depth and area of the reception feature, and the data size.

INDEX TERMS 3D CNN, CNN architecture, Alzheimer's disease, reception area, feature redundancy, data size, MRI classification.

I. INTRODUCTION

In reference to the Alzheimer's Association Report (AAR) [1], the molecular and neurological causes for Alzheimer's disease (AD) takes place in the neurons, i.e. the brain nerve cell connection area also called the synapsis; this is where the neurotransmitters are released. The synapsis helps with the information flow caused by tiny bursts of chemicals that are released by one neuron and are detected by a receiving neuron. During AD, there is an accumulation of ß-amyloid proteins and tau proteins, also known as tau tangles that are around the synaptic region. This ß-amyloid is suspected to cause neuron death by interfering with neuron-to-neuron communication at the synapsis. In addition, the tau tangles block the supply of nutrients and other essential molecules inside the neurons. Brains with advanced AD have a dramatic shrinkage due to cell loss, inflammation, and widespread debris from dead and dying neurons. This causes memory loss problems (e.g. dementia) with the inclination of age. This is the molecular and physiological level analysis for AD. However, there is a corporal change in the common AD-related variation of anatomical brain structures such as the enlargement of ventricles, shrinkage of the hippocampus shape, change in the cortical thickness, and other cerebral areas containing white matter and gray matter brain tissue as well as cerebrospinal fluid. These changes and atrophies are rationally visualized through the brain imaging by the clinician while using a variety of medical imaging modalities like magnetic resonance imaging (MRI), positron emission tomography (PET), and computed tomography (CT) scanning. Here comes the true usage of image processing and machine learning. Image processing improves the quality of the image for better visualization of the brain whereas machine learning assists clinicians to perform other logical operations like segmentation, classification, and quantification, which can be time-consuming and sometimes baffling. The logical operation once modeled with proper

supervision can later follow the designed algorithm to reach a prediction, the more the prediction is true, the better the model is, and the higher will be the chance of reliability. Mild cognitive impairment (MCI) is a transitional stage between normal aging and the preclinical phase of dementia. MCI is considered to be a possible early stage of AD, and it can either progress into AD (pMCI) or remain in the same stage throughout life, which is called stable MCI (sMCI). Here, we are combining both types in a single MCI group to ease the classification process. A healthy MRI is called normal aging/cognitively normal (CN). Since AD contains a genome that affects the disease, no known stimulant causing it is identified. However, the influencing factors for AD include genetics, low education or professional involvement, lack of-mental exercise, family chronicles, and external or internal brain injuries [2].

Image processing aims to find a discriminative pattern of image features by collecting the same groups of the MRI into one. It means that the pattern that we eventually discover for AD patients will behave the same for other AD patients' recognition but are differentiated with the CN and MCI effected MRI. Once the MRI is translated into an image from the magnetic resonance frequency, it represents the pixel value for each structure and these pixels will be assigned to a class. Ultimately, AD classification will be based on the features that are extracted from these brain image pixels. The main features required to accurately capture the major AD-related variations of the anatomical brain structure includes the size of the ventricles, hippocampus shape, cortical thickness, and brain volume [3]. Although such alterations may resemble other brain-related diseases like Parkinson's disease (PD) and encephalitis [4]. In that case, more clinical and physiological tests should be performed on a genetic level. Consequently, the idea of identifying pathogenic scans from a healthy one seems easier than identifying a particular disease from a pool of pathogenic scans. Thus imaging technique alone may not be the only valid proof to diagnose a person with AD. However, based on the brain phenotype reflected in the imaging, the discriminative features from the trained network can help identify AD prone images.

This study presents results that answer a few questions related to the use of deep learning for medical imaging. It starts with the background story of CNN and recent literature reviews of its implication in medical image classification. Then the related inquisition of the CNN role for its architecture, hyper-parameters, depth, and data-size is discussed in section II. The mathematical orientation and used pseudo-codes are discussed in section III. In the succeeding sections, we have discussed the performance of different architectures, the role of the hyper-parameters, the selection of data, and the effect of dataset size for the design of the optimal network so it can be implemented practically. Subsequently, we have surveyed with shallow to deep layers using different feature sampling region and finally came up with a diverging architecture being supportive in the case of both MRI and PET. The proposed architecture, which is referred to as 'divNet', and its sibling architectures have been thoroughly investigated and the results are presented in sections VII and VIII. All the results of these experiments are meticulously presented, discussed, and analyzed here. So, we are stating it as a survey-based research paper.

II. THE BACKGROUND STORY

A. 3D CNN

Inspired by the neural network architecture of the mammalian cerebrum, an artificial neural network (ANN) tries to mimic the information flow and the decision-making pattern of the brain. As demonstrated by Hubel and Wesel [5], they recorded the activity of a single brain cell in cats. It was stated that some cortical cells respond to contours of a specific orientation. Aside, patterns of light stimuli are most effective in influencing units at one level and they may no longer be the most effective for the next. Although millions of neurons and synapses receive the stimuli, only certain neurons are trained to respond to those specific features or aspects of an image [5]. Similar to the brain when we receive any stimuli, the neuron spike is generated for only a specific area, ANN will only have a few activated nodes for each shape, which may be a horizontal, vertical, or diagonal line. The node activated for each line is different and unique. This means that the node activated for a horizontal line in one image is activated for the horizontal line in another image and so on; this is the basic principle of an ANN. The layer-wise connection between the nodes may indicate the heavy connection between the neurons.

CNN is similar to ANN, except it has convolutional filter elements (weights) unlike single-node multiplication in ANN. Besides, CNN has extra feature investigators in the form of pooling and activation functions. Thanks to the newly developing algorithms that train the CNNs more effectively, which has ultimately surpassed human-level accuracy for natural image classification [6] [7]. With a wide variety of CNN based topology, the prominent ones include residual (Resnet50, ResNet101 [26]), recurrent (RCNN [24]), inception (GoogLeNet [21]), encoder-decoder (U-net [39]), and so on. One can notice that the common element in all of the topologies is the encoder unit i.e. convolution-normalization-activation-pooling, which acts as the fundamental unit for feature generation. Therefore, we are building blocks of a combination of these encoding layers.

The existing ideas in the 3D CNN are mainly 'the best patch' or 'multiple patches trained for the CNN ensemble' based architectures [8]. In 'the best patch' approach, a single region of the brain is selected based on the recommended region of interest (ROI) or it is manually assisted from the anatomic region of atrophy, like the hippocampus and amygdala whereas in 'multiple patches trained for the CNN ensemble'

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3040486, IEEE Access

multiple CNNs from multiple ROIs are trained separately for each region, later performing feature concatenation at the last fully connected layer (FCL) before classification. One of the reasons behind using only limited/selected/informative pixels to feed in 3D CNN may be due to GPU memory constraints and also to increase the information with quality. Non-discriminative parts although play a role in feature construction at a low level may not necessarily support the cohort classification, hence information becomes redundant using a whole-brain model. Also, selecting an ROI patch, or simply the best region makes the system semi-automatic; hence, the truest sense of automatic feature extraction is not applied in these cases. This research aims to make the classification simpler and candid rather than a multifaceted process. That's why we want to build an automatic and discriminative CNN that can work for MRI, PET, and any other 'pixelary '(pixel-based) object/entity irrespective of its input size.

Yechon et al. [3] works were mainly focused on the hippocampi region; they proposed multimodal 3D CNN that uses hippocampi region ROI from MRI and hippocampi and/or cortices ROI from PET, without segmentation as a prerequisite task. They separately trained the CNN referenced with VGG architecture, for MRI and PET modalities based ROI and later concatenated from final FCL before final classification. In other similar attempt done for multimodality based 3D CNN, Liu et al. [9] also proposed a simpler CNN model like Yechon et al. but instead of concatenating the final FCL, the concatenation was done in the convolution layer, from each CNN (trained using PET and MRI patch) for sequential convolution until flattening features at FCL. They experimented with T1-MRI and FDG-PET based cascaded CNN, which utilizes a 3D CNN to extract features, and adopted another 2D CNN to combine multi-modality features for task-specific classification. In 2016, Hosseini-Asl et al. [14] proposed a deeply supervised and adaptable 3D CNN (DSA-3D-CNN), trained on structural MRI (sMRI) images, which gives the prediction for the AD vs. MCI vs. CN task. Similarly, Payan et al. [35] also used sparse auto-encoder (SAE) patch-based 3D CNN to classify MRI scans using dataset partitioning unlike Oh et al. [36], where they performed 5 fold cross-validations (CV) using convolutional auto-encoder (CAE) based volumetric or 3D CNN for AD vs. NC and supervised transfer learning for sMCI vs. pMCI classification.

B. WHY MOVE FROM 2D TO 3D?

This study aims to explore one more dimension for CNN i.e. the depth. And the key question that needs to explore is: can we only depend on the 2D CNN results?

As mentioned, the 2D CNN can easily be misled [38] in the sense that a target domain trained CNN can only give a probability score for each trained class. Besides, a few pixel changes can make the prediction a disaster [38]. Some researchers have suggested possible improvement in performance over 2D images if 3D whole-brain structure is used to train CNN [10], due to its deeper architecture. But deeper architecture means more parameters (weights in layers) to train, and at the same time requires bigger and better training material. CNN either 2D or 3D follows a generic feature extraction pattern [11][12], here generic features might suggest CNN features, also called 'off the shelf CNN features' [13] which is basically the image features extracted from the multiple convolutional layers as the weights (as a decimal number) of the trained network, applying various activation functions. Typically, the final feature weights from the FCL are graphed out to decide the performance of CNN. This means a well-separated class-based segmented graph generally depicts a well-trained classifier [14].

The problems with 2D CNN are to select appropriate slice or slices and its orientation as training inputs. A number of the literature suggests the 'best scan' [15] [16] or 'best multiple slices' [17] [14] for efficient performance, which rather mystifies the slice selection process. This is problematic and quite impracticable every time. Some important information might be missed if we focus only on limited scans or orientation. So the best and safest way to ensure is to use whole brain volume, which comes in a three-dimensional pixel value meaning, pixel value for x, y, and z dimension in planar geometry. In our previous work [18] we demonstrated that 2D CNN when trained from fewer MRI images results in

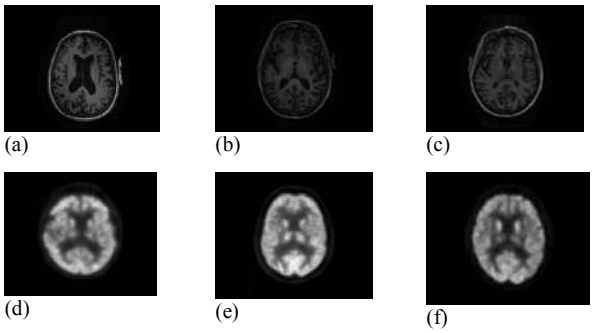


FIGURE 1. MRI and PET scans of: (a) AD prone MRI; (b) Healthy MRI; (c) MCI affected MRI; (d) AD prone PET; (e) Healthy PET; (f) MCI affected PET.

poor classification performance, moreover the selection of slice or slices is still ambiguous. Furthermore, the dimension constraints of 2D-CNN need to make the architecture deeper and bulkier to accommodate thousands of images per class. Hence to make the MRI classification universal and less tedious, 3D MRI fits readily into 3D CNN. Besides, the transfer learning idea seems an inappropriate choice as the popularly available models like AlexNet [19], ResNet [20], GoogLeNet [21], ZNet [22] are all 2D based architecture. The use of 3D input requires fewer pre-processing steps like slice-correction, slice-selection, and slice-extraction. As a result, the manual processing step is reduced and makes the system more robust and automatic which is the goal of this

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3040486, IEEE Access

study. Regarding image preprocessing we have only performed image resize and normalization before being fed into the CNN because we want to make the system less customized and work indifference to the imaging protocols and scanners, selections to be discussed in section VII B. Aside, mature preprocessing steps require more effort and operation time, keeping in mind that it is already processed from Alzheimer's disease Neuroimaging Initiative (ADNI), (we are not provided with the raw image from scanner, but a semi-corrected processed MRI). This can eventually be useful for the generalization of the trained model.

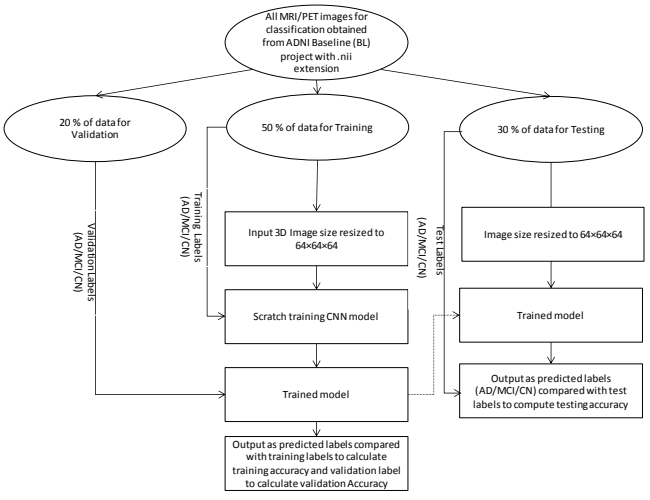


FIGURE 2. **(a) Work flow of the experiment (b) Pictorial representation of proposed 3D CNN architecture for the MRI/PET classification on the basis of the diverging area of the reception, which is referred to as 'divNet'. The original Visio image for better visual attached in Appendix.**

C. FINDING THE CORRECT ARCHITECTURE AND HYPER-PARAMETERS

Although CNN can be easily misled, CNN is quite smart. Irrespective of the depth (deep or shallow layer), the training material (good or bad), or the training size (small or big), CNN finally learns something when it is trained. This 'something' may not typically relate to the human interpretable logical features (say like the number of legs in a dog in comparison to a human) however they will categorically learn some details so it can be classified. Most of the time this involves basic shapes, edges, corners, and patterns on the objects. So, we don't need to worry about selecting architecture every time, nevertheless when it comes to finding the best architecture, with ease of training, and good performance. The trio gives an ultimate contest to any deep learning researchers. Performance results, training time, validation period, the confidence of prediction, generalizability, and other factors are the key to determine the state-of-the-art winner. The results of our experiment are highlighted in Tables I, II, and III.

D. HOW DEEP SHOULD WE GO?

Recent studies have suggested that a CNN can extract convenient features directly from a raw image, unlike a manually supervised learning algorithm and it has a strong capability to locate key points and features in object detection tasks for natural images [23] [24]. This property of the CNN has been explored in a region-based convolutional neural network (R-CNN) for region-based detection in 2D images. Other work in segmentation using a CNN suggests that segmentation results itself do not contain information needed for the classification, hence not being a pre-requisite for the classification task subsequently the CNN can learn useful features without labeling the voxels itself [3]. These entire experiments advocate supporting the generic feature extraction property of CNN. But how deep should we go is the question. Our obvious choice of going deeper is to extract more meaningful features to perform a relevant operation of classification or segmentation from the trainee dataset. In general, we will have more feature vectors with more layers, and subsequently a large pool of features to extract from. This will help in terms of 'judging' the best out of the good features. Nevertheless 'we should go deeper' [25] doesn't necessarily mean for the deep learning model and not every time. Besides, the result is not that supportive. The work of He et al. [26] in ResNet shows that a deeper network with 1,202 layers in comparison to 50, 101, and 152 convolutional layers has no significant improvement with an aggressive depth. With the additional cost of extra training, more depth for a network may make it more prone to overfitting by learning "too well" and this may not generalize the model at the cost of running expensive GPUs which makes it more challenging to build models, being able to understand all details [27].

E. DATA AS FUEL FOR CNN, BUT HOW LARGE SHOULD OUR DATA BE?

The breakthrough of the ImageNet dataset with its implementation in Alexnet suggests that the better the data, the better would be the result. To support this theory artificial dataset are also created with different augmentation techniques. And well, the result seems to be supported by the use of extensive synthetic MRI for improved performance in segmentation and classification tasks [28] [29]. The case with ImageNet is the classification

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3040486, IEEE Access

of 1000 classes with around 8000 images in each class, which means more classes with more distinctive images, similar is the case with other datasets like CIFAR101, Caltech, etc. where data acts like oil to AI [30]. Having said that, what may be the case with the medical image? Considering labels as the most precious assets for the data scientist, how voluminous should the training materials be? In the case of medical images, the task is more challenging, with an image-based feature; we can rarely detect the atrophy pattern. Particularly if we look at AD vs. MCI or MCI vs. NC MRI or PET [FIGURE 1]. Hence to solve this we are experimenting with various sizes of the datasets, one big and the other small for MRI and PET tests. The results are highlighted in Table IV. Detailed demographics for each dataset type tabulated in the Appendix.

F. VISUALIZING FEATURES: WHAT HAS THE CNN EXTRACTED AND LEARNED?

A generalized CNN follows the reduction of features from the input to the final classification layer. The same is in this case, the input for the CNN is a 3D MR image obtained in NIfTI (Neuroimaging Informatics Technology Initiative) format with .nii extension. Once input is read using *niftiread* function inbuilt in MATLAB, it can be resized from its original size 256×256×256 or 256×256×170 to 64×64×64. After multiple down-sampling using the max-pool operation for each convolutional layer, it is reduced to 1,728 for the first FCL. To reduce overfitting, we have used the dropout and the other conceding FCL to make the final output 100 features per class, which is the input for the softmax layer. This idea of using multiple FCLs to map the target domain is often called target domain fine-tuning, which is the basis for transfer learning while using pre-trained networks. The activated features in the initial convolutional layer can detect pixel changes based on attributes like line, edge, and color [31] in a small window filter. These edge-based features pass through the intermediate layers of the CNN, and they are combined in a large number of filters, whose weights (initially kept at random weights or initialized using Xavier, He, Gaussian) is updated using backpropagation training following a specific optimization path like stochastic gradient descent (SDG) or Adam. These intermediate layers detect the activated parts of the image whereas the final layer learns discriminative features in the shape and pattern amongst the target domains. Once training reaches convergence, which means no more weight changes occur and the training accuracy reaches its maximum, the training stops. The network is now trained and it's a generic feature extractor, which is like a traditional algorithm that generates features. The generated features are the discriminative features that are used to distinguish the classes. This study uses multiple 3D filters that give 4D output in each layer i.e. one 3D feature map per filter, see [FIGURE 2]. Convolving the image with these filters produce a feature map that detects the presence of those features in the image. This nature of a CNN is the essence of its auto feature extraction and it helps in the automatic computer-aided design (CAD) system.

It is difficult to predict the features that a CNN can learn without training it; thus, making it a tedious task to analyze the features. Since a single network may contain millions of parameters and we cannot mathematically predict the final converged value in each filter without training them. Hence, every time we train the CNN, the learned features need to be investigated. Once trained, the CNN is loaded with the filter weights, which are used to make the predictions with the test images. It is convolved in each layer to obtain different results for the different MRIs. The trained network is used to obtain the features as described in Pseudo-code 1, 2, and 3.

III. PARAMETER INITIALIZATION

Let's assume that the MRI/PET has a 64×64×64 matrix represented by I (i.e. $I = \left[I_{x_i y_i z_i} \right]_{i=1 \text{ to } 64}$). In total, this will result in 262,144 gray-scale values, which is the numerical representation for the 3D image. Since we are working in 3D, we will call each of these values a voxel, not a pixel. Each voxel has a 3D value with x, y, and z coordinates. Here, we are simply representing the MRI as a cube.

Hence, each voxel value mathematically assigns three coordinates, but for easy representation, we will use the single vector notation v where, $v = \left[I_{x_i y_i z_i} \right]$ to make the computation simple. Let us consider the first convolution in the first layer as in Equation (1). Here, b_1^1 and $w_{N,1}^1$ represent the initial bias and the weight of the first convolution kernel in the N^{th} filter, which uses an initialization algorithm. Note ⊗ represents element-wise multiplication.

$$[x_1^1, x_2^1, x_3^1, .., x_{64}^1] = [b_1^1, b_2^1, b_3^1, .., b_{64}^1] + [v_1, v_2, v_3, .., v_9] \otimes [(w_{1,1}^1, w_{1,2}^1, w_{1,3}^1 ..... w_{1,9}^1)] \quad (1)$$

The window of the convolution operation then keeps on moving according to the stride size. To reduce this mathematical expression, this can be rewritten with shorter terms. For each node of the 3D convolution filter:

$$x_k^l = b_k^l + \sum_{i=1}^{N_{l-1}} conv_{\cdot 3}\left(w_{ik}^{l-1}, s_i^{l-1} \right) \quad (2)$$

where $conv_{\cdot 3}$ is a regular 3-D convolution without zero paddings on the boundaries. Following Equation (2), x_k^l is the input, b_k^l is the bias of the k^{th} neuron at layer l, and s_i^{l-1} is the output of the ith neuron at layer $l–1$. w_{ik}^{l-1} is the kernel (weight) from the ith neuron at layer $l–1$ to the k^{th} neuron at layer l. $conv_{\cdot 3}$ represents an element-wise multiplication of the [3×3×3] kernel size. For the very first convolutional layer, the input s_i^{l-1} is the 3×3×3 matrix of the image pixel value (maybe normalized) that is scanned by a window of the same size.

When represented in a matrix or a discrete form, the N-dimensional convolution for the discrete, N-dimensional variables A and B can be defined with (3):

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3040486, IEEE Access

$$C(j_1, j_2, \ldots, j_N) = \sum_{k_1} \ldots \sum_{k_N} A(k_1, k_2, \ldots, k_N) B(j_1 - k_1, j_2 - k_2, \ldots, j_N - k_N) = conv_N(A, B) \quad (3)$$

Each k_i runs over all of the values that can lead to legal subscripts for A and B. Thus, the 3D convolution runs as follows. The layer convolves the input by moving the filters along the input vertically and horizontally. Afterward, it computes the dot product of the weights and the input, and then it adds a bias term. As the filter moves along the input, it uses the same set of weights and the same bias for the convolution; thus, forming a feature map.

Pseudo-code 1	**Pseudo-code 2**	**Pseudo-code 3**
% Activated feature extraction of convolution layer from a single MRI '*I*' % For the trained network 'N' at layer '*l*' we have, Filter feature (F_l) = weights of the filters at layer '*l*' with a size $k \times k \times k \times f$; where $k \times k \times k$ is the size of the filter and f the number of filters at layer '*l*' Initial activated feature (A_1) =*conv* (F_1 ,*I*); Here, *conv* performs the basic 3D convolution as described in section III. This A_1 goes through batch normalization and max-pooling to downsample its size and also pass through the activation layer. Similarly, *Activated feature* (A_l) =*conv* (F_l , A_{l-1}) of size $k \times k \times k \times f$ [A_{lmax}] = maximum value of A_l Activated feature for Visualization= A_{lmax} of size $k \times k \times k$ is resized to 64×64×64 and each slice viewed separately in the 2D domain.	% Activated feature visualization of the FCL from the whole test set using the T-SNE projection % For the trained network 'N' at layer '*l*' similar operation to Pseudocode 1 is performed for the FCL so that, *FCL features* (FCL_l) = weights of FCL with size T× S, where T is the number of test subjects, S=O×I, and O, and I represent the output and input, respectively, of the FCL at the l^{th} layer. Now, $FCL_{l\text{-}tsne}$=T-SNE (FCL_l), where T-SNE perform T-distributed Stochastic Neighbor Embedding to find the 2-dimensional feature matrix of size T×2, while performing feature reduction from the N dimension. $FCL_{l\text{-}tsne}$ is plotted in the x-y plane against its target class-color to visualize the discriminative pattern. The better the separation of the same colored set, the better the classification. The inclusion of a few odd color data in a cluster leads to errors in the test set. This is used when comparing the output for the four different architectures as presented in Fig 6.	% Final layer FCL weights of the trained networks for direct visualization % The previous two pseudocodes are used to examine the property of a single trained network; however, in order to compare the networks directly, the easiest way is to plot the final FCL weights. For a network 'N' with '*l*' as the last FCL layer and the number of classification categories 'n' FCL_l = Weights of the FCL_l of size O×n, where O is the output size of the penultimate FCL, which is 100 in our case, 'n' is the output size of the final FCL at layer l, which is equivalent to the number of classes (here n=3). FCL_l is an O×n matrix, which is simply plotted in the X-Y plane as a linear graph. By having more distance between the three lines, it becomes a better classifier. This is used when comparing the output for the four different architectures as presented in Fig 8.
When visualized, the clearer the visual, the better the features that are learned. The difference between the images in each MRI type represents the difference in the pattern of the MRI, and the complexity of the discrimination increases with the increasing layer number. This is used when comparing the output for the four different architectures as presented in Fig 4.		

In the SGD algorithm, the filter weights during the optimization are iteratively updated as shown in Equation (4) and Equation (5), where W_l^t denotes the weights in the l^{th} convolutional layer for the t^{th} iteration and E denotes the cost function (updated using backpropagation for minimizing the cost function) over a mini-batch of size N.

$$W_l^{(t+1)} = W_l^t + V_l^{(t+1)} \quad (4)$$

where $V_l^{(t+1)}$ is calculated as

$$V_l^{(t+1)} = m.V_l^t - \gamma^t.\alpha_l \frac{dE}{dW_l} \quad (5)$$

Here, α_l in Equation (5), is the learning rate for the l^{th} layer, m is the momentum due to the previous weight update in the current iteration, and γ is the scheduling rate that decreases the learning rate for the completion of each epoch. If α_l =0 then this depends on the value of l. All of the layers from 1: l are not updated in terms of their weight; hence, the weights are transferred in the final version of the trained model.

A. PARAMETER TRAINING

$$Total\_error(E) = E(y_1^L, \ldots y_N^L) = \sum_{i=1}^{N_L} \left(y_i^L - t_i\right)^2 \quad (6)$$

This error in Equation (6) is a mean squared error, which is obtained by adding the MSE value of the deviation from each of the samples (i.e. training data (t_i) from the predicted value (y_i^L)). Here, the upper subscript L denotes the output for the final layer. Based on the obtained error (E), backpropagation (BP) is performed to update the weights for each parameter as in Equation (7) [32]:

$$\frac{\partial E}{\partial w_{ik}^l} = \frac{\partial E}{\partial x_k^{l+1}} \frac{\partial x_k^{l+1}}{\partial w_{ik}^l} = \frac{\partial E}{\partial x_k^{l+1}} y_i^l \quad (7)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3040486, IEEE Access

Here, the output of the x_k^{l+1} filter 'k' is the number of filters in the l^{th} layer, and the weights of the previous layer '$l+1$' give the output y_i^l of the l^{th} layer during the BP. Similarly, the bias is also updated as Equation (8):

$$\frac{\partial E}{\partial b_k^l} = \frac{\partial E}{\partial x_k^l}\frac{\partial x_k^l}{\partial b_k^l} = \frac{\partial E}{\partial x_k^l} \tag{8}$$

As a result, it is written for the whole length of 1 to $l+1$ layers; hence, it can be summed up as follow for N number of filters in the $l+1$ layer to obtain y in the l^{th} layer as in Equation (9):

$$\frac{\partial E}{\partial y_k^l} = \sum_{i=1}^{N_{l+1}} \frac{\partial E}{\partial x_i^{l+1}}\frac{\partial x_i^{l+1}}{\partial y_k^l} \tag{9}$$

During training, we need to backpropagate the gradient of the error ∂E through this transformation, and to compute the gradients with respect to the parameters as the batch normalization (BN) transforms.

All experiments were conducted using MATLAB R2019a academic software on Windows 10 OS. Network models were trained on NVIDIA GeForce RTX 2070 GPU with 24 GB of memory and tested in Intel® Core™ i5-9600K CPU @ 3.70 GHz with 32 GB of memory. The trained mat file will be provided to researchers upon request to the authors.

IV. TEST ON DIFFERENT CNNs

In order to define an optimal number of layers for our input of 64×64×64 3D scan, we tested from an initial layer of single encoder i.e. Convolution-Batch normalization-ReLu-max-pooling, and stated it as an L1 layer. Similarly, the encoder blocks were further implemented on the L2, L3, L4, L5, and L6 layer consecutively. In L6, the final feature size from the sixth convolution was [2 2 2] for each of the 64 filters. This means that the filter kernels have only two pixels in length for each filter; hence, expanding this to the L7 layer would be an inoperable idea and will ultimately reduce the features. Hence, we didn't use seven convolutions based architecture. Table I shows the result of classification on these layer-wise CNN, whereas Table II presents the result of classification using four different architectures based on the reception area i.e. window size of the convolution kernel. Similarly, the training and validation graph was also studied to observe, how the architectures affect the training and also help to better understand the convergence process of each CNN, Figure 3. Correspondingly, to understand the extracted features, from each convolution layer, a single MRI from each target domain was passed and the feature was observed as in Figure 4. On minute observation we could find the difference in the lines, edges, intensities, and other patterns based on the class domain. Moreover, FCL layers were visualized using t-SNE projection as in Figure 5 for each architecture, so we could support our finding. Here, the features were visualized for the whole test set, so this will help us to judge which architecture has segregated the feature in a better way. Finally, the results from different hyper-parameter settings and datasets are tabulated in Table III and Table IV respectively.

V. WHY DIVERGING ARCHITECTURE?

The filter size determines the scanning window during the convolution and the size of this window can be analogized as the reception area. We have increased our filter size by two strides in each consecutive layer so that the feature extracted will be sequentially extracted at a low level, an intermediate level, and a high level with a higher area of reception for the successive layers. The low-level features are extracted from the 3×3×3 filter window and it is max-pooled by the 2×2×2 windows with a stride of one from the first convolution layer (i.e. conv_1 to max-1) [FIGURE 2 (b)]. We call this a diverging network in the sense that the size of the filter kernel keeps on increasing with an increase in the step size or the stride; however, the number of filters in each layer is same (i.e. 64) to maintain the channel size for the input of 64×64×64. Beginning from the first convolutional layer, with filter size 3×3×3; hence, a minute detail can be easily captured. Once the layer deepens, we can accumulate the features by increasing the window size for each layer. Consequently, the max-pool stride is also increased to reduce the redundancy in the feature. Conversely, the area of the reception keeps on decreasing with an initial filter size of 9×9×9 in the converging network, whereas in the equivalent architecture, a uniform kernel size of 3×3×3 is used in each convolutional layer. All of the details in the architecture and the results of the experiment after training and testing are highlighted in Table II, which includes the parameters in the second column.

VI. PET OR MRI OR BOTH?

To find the effect of the size of the training material, we trained the L4 diverging network with a variety of datasets and the results are shown in Table IV. The used MR images and PET images were all obtained from patients of ADNI BL visits obtained under the ADNI 1 project [42]. We used 3D scans of T1 weighted structural MR images of whole-brain; normalized, and processed using ADNI pipeline also few scaled (listed in Appendix), whereas PET scans were also obtained from ADNI BL; processed for smoothing, co-registration, and few standardized (listed in Appendix). Our experiment showed that MRI is a better imaging modality than PET for 3D CNN classification. When the network is trained with the smallest dataset including MRI1 (see Table IV, 5th column for the type), the network gets under-fitted; hence, the testing accuracy was low at 74.5%, which is slightly lower than the validation accuracy. However, the training achieved convergence as the accuracy reaches 100%. The same network when trained with the BASELINE_MRI data (type MRI2, see Table IV) under the same environment achieved the highest testing accuracy of 94.5%. The reason

behind the increased accuracy may be due to the higher scans per patient ratio (SPR), which decreases the variability for each scan and loses its generality in the network. The PET scan performed the worst in the L4 divNet with increased training time. The BASELINE_PET_SMALL dataset, PET1, has a testing accuracy of only 66.34%, whereas the bulkiest PET dataset (i.e. BASELINE_PET_ALL, PET2) testing accuracy reached only 50.21%, along with difficulties in achieving convergence with 100 epochs and GPU training time almost three times of PET1 though it is ten times bigger in size than PET1. Finally, the MRI2+PET1 datasets were merged and trained in a single network however, it could only reach a 90% training accuracy after convergence and reached the testing accuracy of up to 82%. As a result, it seems like MRI is a better choice for CNN, and PET only has a complementary role for the AD prediction. It is worth mentioning that the PET image is visually not so discriminative by the target class in comparison to the MRI image (see FIGURE 1), which may have resulted in the MRI's better performance.

A. TEST ON DIFFERENT LAYERED CNN

Table I highlights the results from the diverging architecture-based configuration with the use of different layers, starting with two convolution encoding layers to six. The parameter column details the filter size, number of filters, max-pool filter size, stride, and FCL input and output number as indexed in each row. Training accuracy reached almost 100% for each configuration, whereas the validation and testing accuracy start dropping after the L4 layer. This could be the optimal case as plotted in training and validation loss against the epoch numbers as shown in Figure 3(a) to 3(f), with the remarks for overfitting or under-fitting cases.

VII. EXPERIMENTAL RESULT

We present all of the results of our experiments in the tables and figures below.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3040486, IEEE Access

TABLE I

TRAINING AND TESTING RESULTS FOR THE DIVERGING ARCHITECTURES WITH CHANGING NUMBER OF LAYERS AS SPECIFIED IN THE PARAMETERS COLUMN. HERE, C [W*W*W N, S] REPRESENTS A CONVOLUTIONAL LAYER WITH N FILTERS SIZED W EACH DIMENSION, MOVING BY STRIDE S AND N BIASES. TC [W*W*W N, S] REPRESENTS A TRANSPOSED CONVOLUTIONAL LAYER WITH N NUMBER OF FILTER SIZED W EACH DIMENSION, MOVING BY STRIDE S AND N BIASES. BN [N] REPRESENTS THE BATCH NORMALIZATION WITH AN OFFSET OF N AND N SCALE VALUES AS LEARNABLE PARAMETERS. R REPRESENTS THE RELU ACTIVATION. M[W*W*W S] REPRESENTS THE MAX POOLING WITH W KERNELS WITH A STRIDE S, FC[O*I] REPRESENTS THE FULLY CONNECTED LAYER WITH INPUT I AND THE OUTPUT O. CT, D, S, AND C REPRESENT THE CONCATENATION, DROPOUT, SOFTMAX, AND THE CLASSIFICATION LAYER, RESPECTIVELY. THE TRAINING PATTERN IS SHOWN IN FIG 3.

Diverging architecture	Parameters	Training accuracy	GPU training time (min)	Validation accuracy (%)	Testing Accuracy (%)
2 LAYER CONV (L2)	C[5*5*5 64,1] BN[64] R M[2*2*2 2] C[9*9*9*64 64,1] BN[64] R M[2*2*2 4] FC[1728*32768] D FC[864*1728] D FC[100*864] D FC[3*100] S C	99	778	93.4	94.26
3 LAYER CONV (L3)	C[5*5*5 64,1] BN[64] R M[2*2*2 2] C[7*7*7*64 64,1] BN[64] R M[2*2*2 3] C[9*9*9*64 64,1] BN[64] R M[2*2*2 4] FC[1728*1728] D FC[864*1728] D FC[100*864] D FC[3*100] S C	100	664	91.88	94.66
4 LAYER CONV (L4)	C[3*3*3 64,1] BN[64] R M[2*2*2 1] C[5*5*5*64 64,1] BN[64] R M[2*2*2 2] C[7*7*7 *64 64,1] BN[64] R M[2*2*2 3] C[9*9*9*64 64,1] BN[64] R M[2*2*2 4] FC[1728*1728] D FC[864*1728] D FC[100*864] D FC[3*100]S C	100	842	95.43	95.59
5 LAYER CONV (L5)	C[3*3*3 64,1] BN[64] R M[2*2*2 1] C[5*5*5*64 64,1] BN[64] R M[2*2*2 2] C[5*5*5*64 64,1] BN[64] R M[2*2*2 2] C[7*7*7*64 64,1] BN[64] R M[2*2*2 3] C[9*9*9*64 64,1] BN[64] R M[2*2*2 4] FC[1728*64] D FC[864*1728] D FC[100*864] D FC[3*100] S C	100	786	93.4	92.91
6 LAYER CONV (L6)	C[3*3*3 64,1] BN[64] R M[2*2*2 1] C[5*5*5 64,1] BN[64] R M[2*2*2 2] C[5*5*5 64,1] BN[64] R M[2*2*2 2] C[7*7*7 64,1] BN[64] R M[2*2*2 3] C[7*7*7 64,1] BN[64] R M[2*2*2 3] C[9*9*9 64,1] BN[64] R M[2*2*2 4] FC[1728*64] D FC[864*1728] D FC[100*864] D FC[3*100] S C	100	780	95.43	92.57

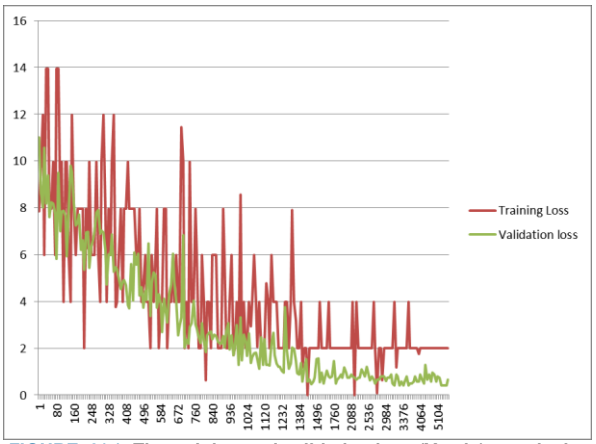


FIGURE 3(a). The training and validation loss (Y-axis) graph showed under each iteration (X-axis) of 100 epochs for the L1 convolution as presented in Table I.
Remarks: The VL is much less than TL, which indicates a possible overfitting case

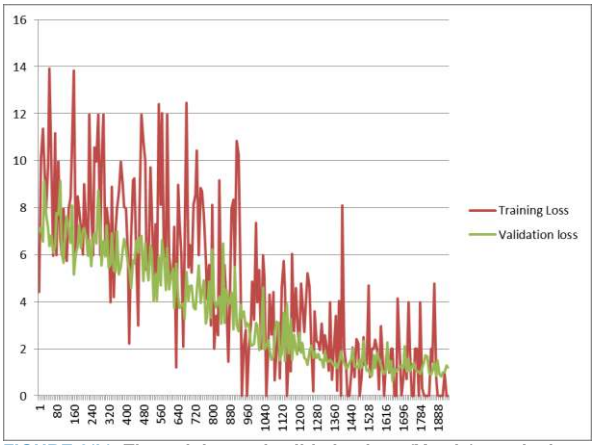


FIGURE 3(b). The training and validation loss (Y-axis) graph showed under each iteration (X-axis) of 100 epochs for the L2 convolution as presented in Table I.
Remarks: The VL is less than TL, which indicates a possible overfitting case.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3040486, IEEE Access

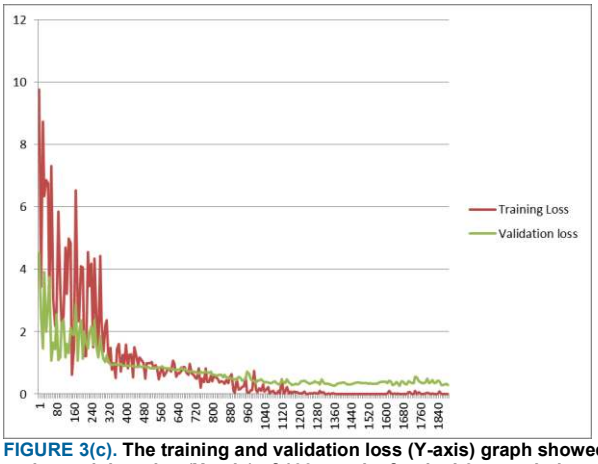


FIGURE 3(c). The training and validation loss (Y-axis) graph showed under each iteration (X-axis) of 100 epochs for the L3 convolution as presented in Table I.
Remarks: The VL is higher than TL, which indicates a possible under-fitting case.

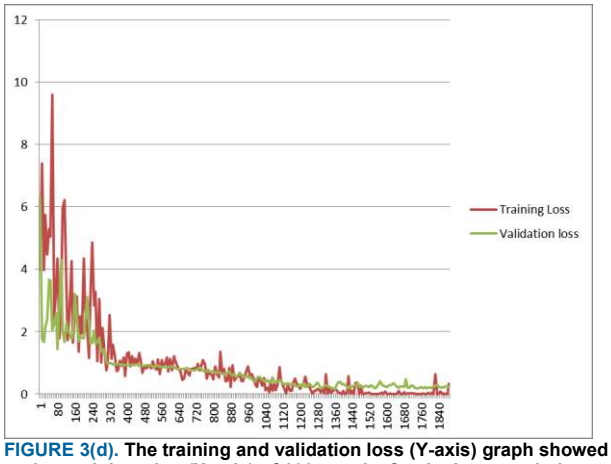


FIGURE 3(d). The training and validation loss (Y-axis) graph showed under each iteration (X-axis) of 100 epochs for the L4 convolution as presented in Table I.
Remarks: The VL is slightly higher than TL, which indicates a possible optimal case.

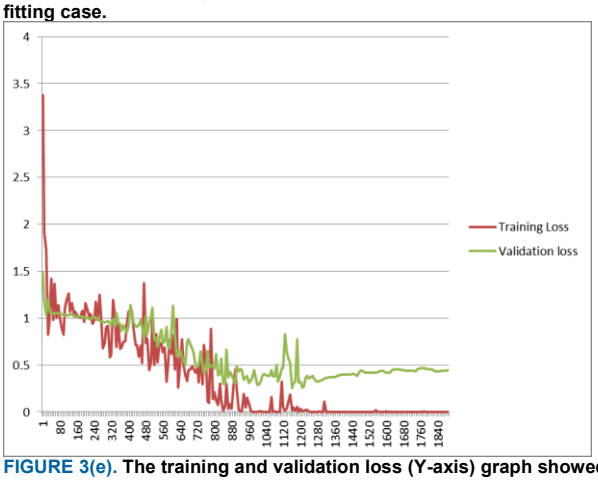


FIGURE 3(e). The training and validation loss (Y-axis) graph showed under each iteration (X-axis) of 100 epochs for the L5 convolution as presented in Table I.
Remarks: The VL is much higher than TL, which indicates a possible under-fitting case.

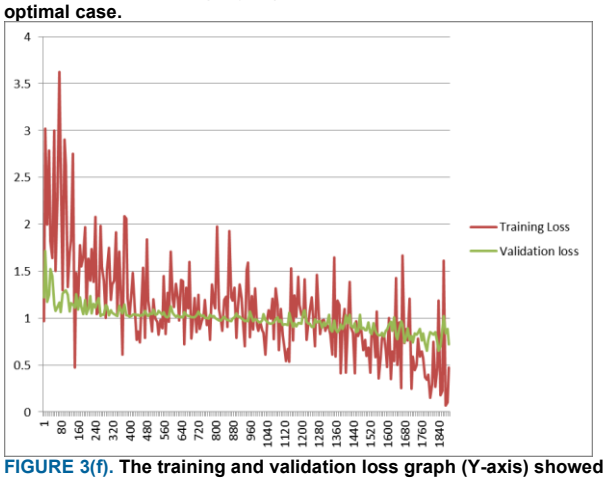


FIGURE 3(f). The training and validation loss graph (Y-axis) showed under each iteration (X-axis) of 100 epochs for the L6 convolution as presented in Table I.
Remarks: The VL and TL both have higher values, which indicate a possible under-fitting case.

B. TEST ON DIFFERENT ARCHITECTURES

As discussed in section IV, the results using different architectures based on the reception area of convolving filter size i.e. the results from 4 architectures viz; diverging, equivalent, converging, and U-net are presented as in Table II. The parameter column is indexed as same as in Table I.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3040486, IEEE Access

TABLE II
TEST RESULTS USING VARIOUS TYPES OF ARCHITECTURES. THE PARAMETERS ARE INDEXED AS IN TABLE I.

DIFFERENT ARCHITECTURE	PARAMETERS	TRAINING ACCURACY	GPU TRAINING TIME (MIN)	VALIDATION ACCURACY	TESTING ACCURACY IN THE BASELINE_MRI	PREDICTED CONFUSION MATRIX (CM) FOR TESTING	GROUND TRUTH (GT) MATRIX FOR TESTING
ENCODER-DECODER BASED (U-NET) [39]	C[3*3*3 32,1] BN[32] R C[3*3*3*32 64,1] BN[64] M[2*2*2 2] C[3*3*3*64 64,1] BN[64] R C[3*3*3*64 128,1] M[2*2*2 2] C[3*3*3*128 128,1] BN[128] R C[3*3*3*128 256,1] R C[3*3*3*256 256,1] R C[3*3*3*256 256,1] R TC[2*2*2*512 512,2] CT C[3*3*3*768 256,1] R C[3*3*3*256 256,1] R TC[2*2*2*256 256,,2] CT C[3*3*3*384 128,1] R C[3*3*3*128 128,1] R TC[2*2*2*128 128,2] CT C[3*3*3*192 64,1] R C[3*3*3*64 64,1] R TC[2*2*2*64 64 ,2] FC[100*786432] R D FC[512*1000] R D R FC [3*512]S C	100	3988 (20 EPOCHS)	48.73	41.81	9 32 22 15 47 29 36 56 50	63 0 0 0 91 0 0 0 142
CONVERGING	C[9*9*9 64,1] BN[64] R M[2*2*2 1] C[7*7*7*64 64,1] BN[64] R M[2*2*2 2] C[5*5*5 *64 64,1] BN[64] R M[2*2*2 3] C[3*3*3*64 64,1] BN[64] R M[2*2*2 4] FC[1728*1728] D FC[864*1728] D FC[100*864] D FC[3*100]S C	100	1429	94.92	94.59	60 1 2 0 88 3 5 5 132	
L4 DIVERGING (DIVNET)	C[3*3*3 64,1] BN[64] R M[2*2*2 1] C[5*5*5*64 64,1] BN[64] R M[2*2*2 2] C[7*7*7 *64 64,1] BN[64] R M[2*2*2 3] C[9*9*9*64 64,1] BN[64] R M[2*2*2 4] FC[1728*1728] D FC[864*1728] D FC[100*864] D FC[3*100]S C	100	842	95.43	94.59	58 3 2 0 86 5 1 5 136	
EQUIVALENT	C[5*5*5 64,1] BN[64] R M[2*2*2 1] C[5*5*5*64 64,1] BN[64] R M[2*2*2 2] C[5*5*5 *64 64,1] BN[64] R M[2*2*2 3] C[5*5*5*64 64,1] BN[64] R M[2*2*2 4] FC[1728*1728] D FC[864*1728] D FC[100*864] D FC[3*100]S C	100	790	95.94	93.92	55 1 7 0 85 6 4 0 138	

C. TEST FOR DIFFERENT HYPER-PARAMETER SETTINGS

As discussed in section II C, hyper-parameters play an important role to reach an optimal case for the best performance of the network so we experimented with several activation functions, initialization techniques, and optimization algorithms to find the best case as shown in TableIII.

TABLE III
CLASSIFICATION PERFORMANCE RESULTS FOR THE BASELINE_MRI DATA; UNDER A DIFFERENT HYPER PARAMETER SETTING THAT IS INVESTIGATED IN THE L4 DIVERGING ARCHITECTURE AS LISTED IN TABLE VI.

Selected Architecture	Hyper parameter Description	Selected technique	GPU Training Time (min)	Training accuracy % (50%)	Validation accuracy % (20%)	Testing accuracy % (30%)
L4 Diverging	Initialization technique (Adam optimized, ReLu activated)	Xavier Glorot	842	100	95.43	94.59
		He	850	100	92.39	92.91
	Optimization (Glorot initialized, ReLu activated)	Adam	842	100	95.43	94.59
		SDG	844	100	93.908	92.91
	Activation (Adam, Glorot)	ReLu	842	100	95.43	94.59
		Tanh	850	100	94.42	92.23
		Leaky-ReLu	905	100	93.51	95.61

D. FIGURE FOR EACH ARCHITECTURE'S CONVOLUTIONAL TRANSFORMATION

Convolutional transformation is visualized using Pseudo-code 1; here we present Figure 4 for each class domain analysis, visualized using a single patient MRI scan. The number of features keeps on reducing from the former convolutional layer to the latter one. The result from the L4 diverging architecture network is presented in slice-view, scaled to 64×64 for better visualization.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3040486, IEEE Access

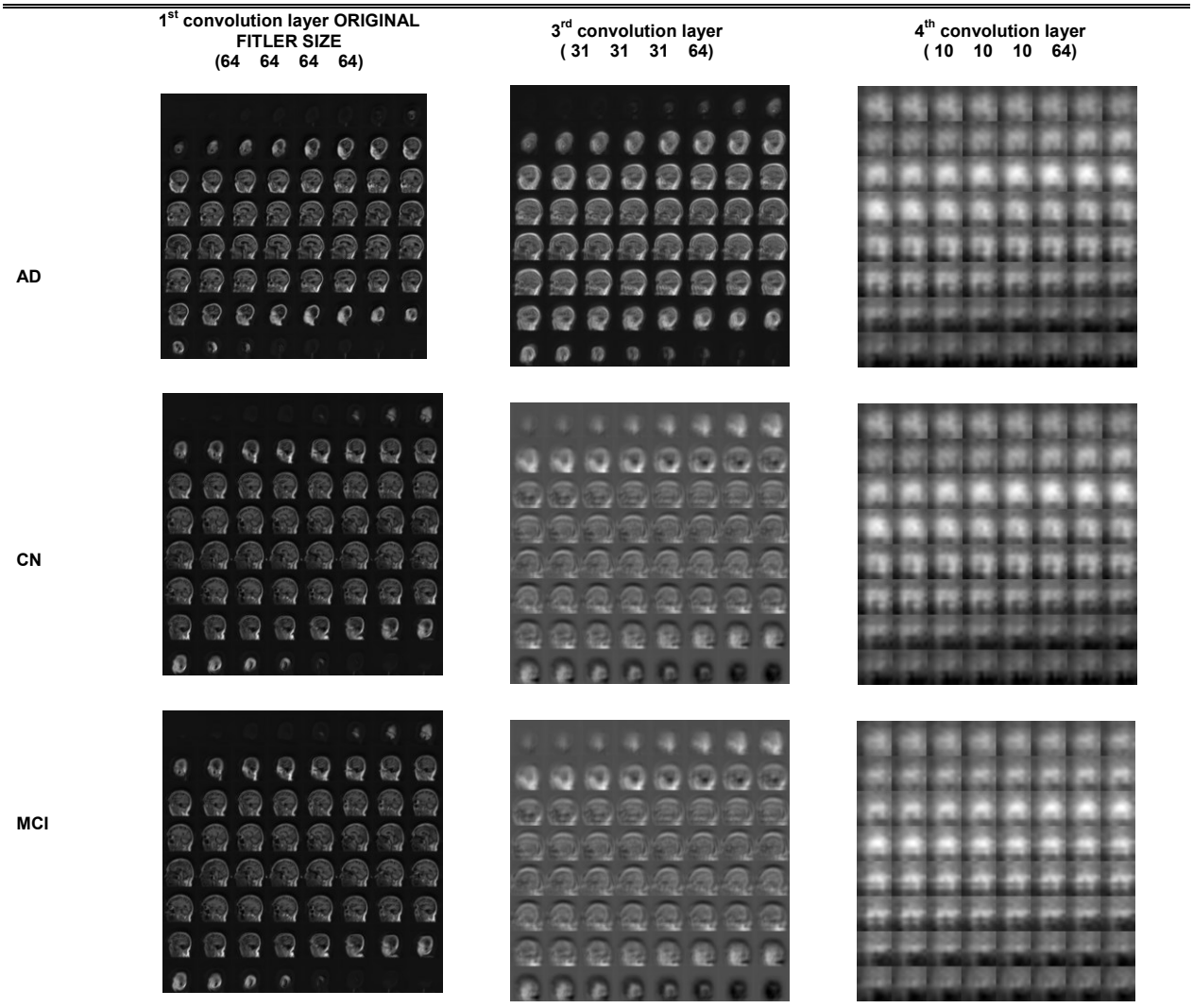


FIGURE 4. Convolution layer visualization of maximally activated feature using single MRI scan, original size resized to [64 64 64], using pseudocode 1, employed network is L4 diverging. Each convolution layer for a typical MRI of AD, CN and MCI category.

E. TEST ON DIFFERENT DATASETS

Although the network is finalized, still the dataset size should be determined as it can heavily impact the network performance. So, we were interested to see how the number of training material affects the testing accuracy and hence we performed experiments for the different datasets as shown in Table IV. Demographic details and file type are listed in the Appendix.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3040486, IEEE Access

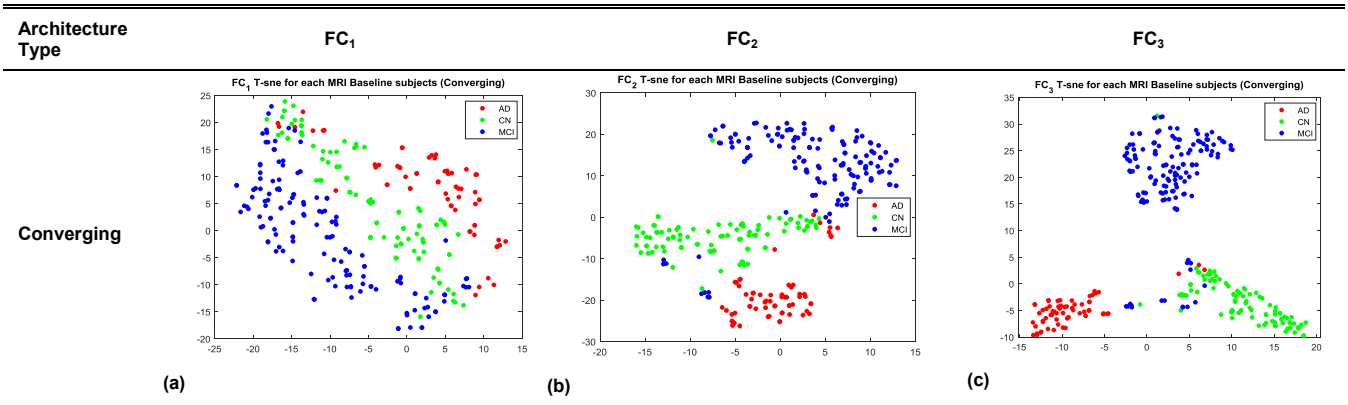
TABLE IV

RESULTS OF THE CLASSIFICATION FOR THE DIFFERENT DATASET SIZES USING L4 DIVERGING. THIS WAS TESTED ON A VARIETY OF DATASET SIZES IN MRI AND/OR PET IMAGING THAT RANGES FROM SMALL TO LARGE SIZE DATASETS. THE MRI1, MRI2 AND PET1, PET2 TYPE ARE DETAILED IN THE APPENDIX.

DATASET TYPE	AD MRI/PET COUNT	CN MRI/PET COUNT	MCI MRI/PET COUNT	INCLUDED MRI/ PET TYPE	TRAINING ACCURACY (50%)	TRAINING TIME (MIN)	VALIDATION ACCURACY (20%)	TESTING ACCURACY (30%)	CONFUSION MATRIX (CM)	GROUND TRUTH (GT)
BASELINE_ MRI_SMALL	54	75	58	MRI1	100%	59	76.32%	74.55%	12 0 4 4 15 3 0 3 14	16 0 0 0 22 0 0 0 17
BASELINE_ MRI	209	305	474	MRI2	100%	842	95.43%	94.59%	58 3 2 0 86 5 1 5 136	63 0 0 0 91 0 0 0 142
BASELINE_ PET_SMALL	102	109	125	PET1	100%	99	66.67%	66.34%	22 6 3 6 20 7 5 7 25	31 0 0 0 33 0 0 0 37
BASELINE_ PET_ALL	1165	1057	974	PET2	70-75%	3026	49.50%	50.21%	134 205 10 3 312 2 17 240 35	349 0 0 0 317 0 0 0 292
BASELINE_ MRI+BASELINE_PET_ SMALL	311	414	599	MRI2+PET1	85-90%	1164	78.79%	82.12%	55 1 37 0 94 30 0 3 177	93 0 0 0 124 0 0 0 180

F. FIGURES FOR EACH ARCHITECTURE'S FCL T-SNE TRANSFORMATION

FC layers weights are visualized using T-SNE transformation as stated in Pseudo-code 2, the result of experiments from each architecture type is shown in Figure 5, where we have presented the class-wise representation of figures for the last three FCL used.



This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3040486, IEEE Access

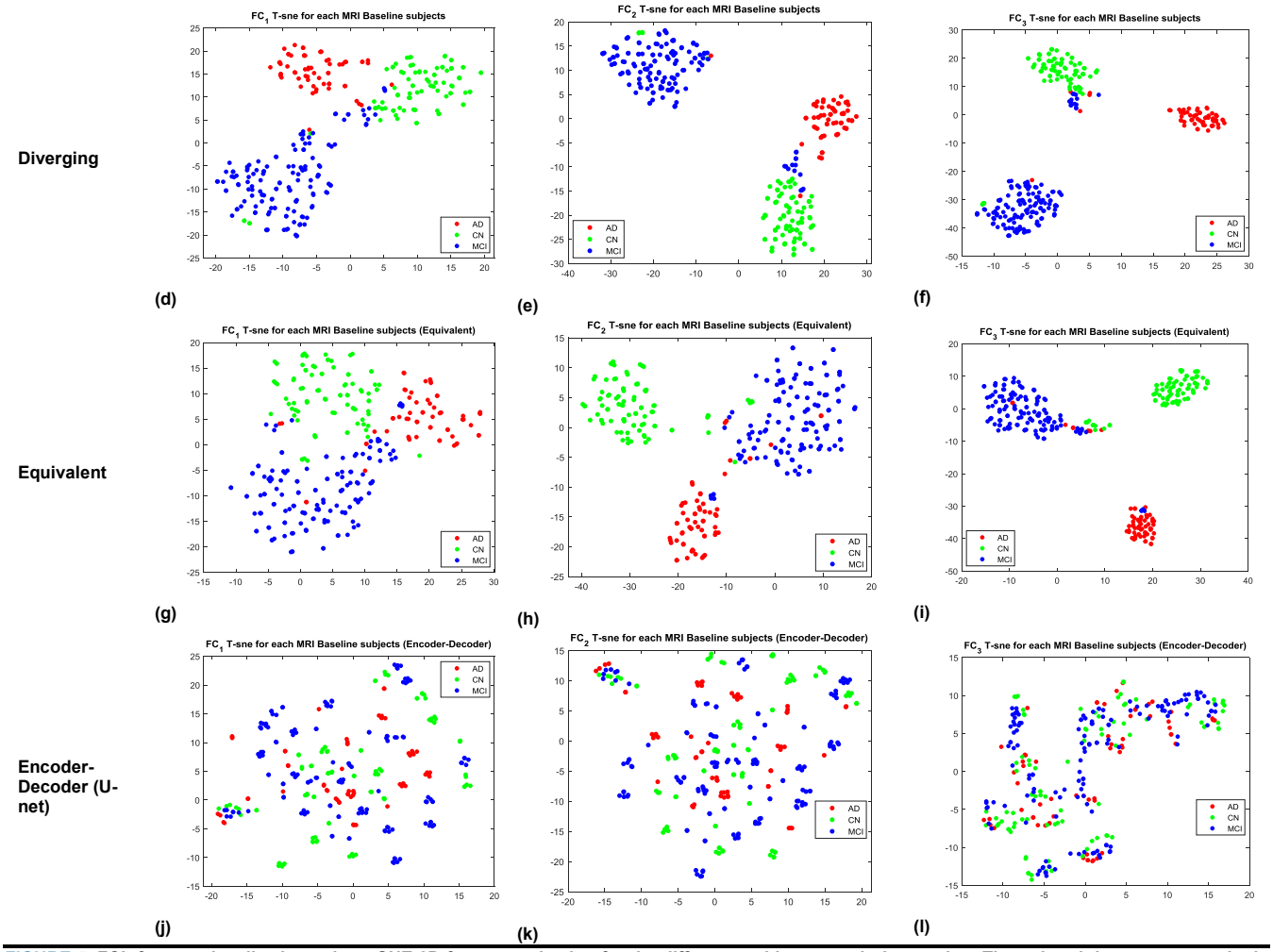


FIGURE 5. FCL feature visualization using t-SNE 2D feature projection for the different architectures during testing. The colored dots represent single MRI scan features from the test set in the first three FCL, namely FC_1, FC_2, and FC_3. The feature starts to show a class-domain property from an FCL, and it is visualized by the start of the formation of the same colored cluster. Based on the visual inspection, we determined that the diverging architecture-based features are better clustered and separated than the others, Fig 5(d)-(f). Meanwhile, there is poor separation in the case of the U-net-based architecture as shown in Fig 5(j)-(l). Here, the training environment and the training material used for training were all the same; the generated models are detailed in Table II. The X-axis and Y-axis represent the values of the 1st dimension and 2nd dimension obtained from t-SNE 2D projection respectively.

VIII. 3D CNN STATE OF THE ART COMPARISON

Hosseini et al. [14] used a deeply supervised adaptable 3D CNN (DSA-3D-CNN) based on the autoencoder network for AD classification that demonstrates feature maps for the various layers. The reported accuracy is 97.06% for the binary classification of the AD/NC using only the MRI dataset. The reported accuracy is from a 10-fold CV, which means that only one MRI in a batch of ten is used in testing, whereas the other nine are used for training and validation. Hence, only 10% of the total image (i.e. 21 subjects) is used for testing [37]. Besides, each image participates in training and testing; thus, the idea of an untouched test set seems to be avoided during cross-fold validation. Oh et al. [36] also performed 5-fold CV with a moderately sized dataset with an accuracy of around 84.5%. Evgin Foceri [33] and Gupta et al. [34] reported accuracies of 98.06% and 94.74% respectively, where they used data splitting and tested them in 20% and 10% of the dataset respectively. Although the accuracy is higher, the SPR ratio is still high, which may cause a generalization error. Payan et al. [35] had an optimal performance for larger data size, with an accuracy of around 89.47% for three classes of AD/MCI and HC. However, here the testing ratio is only 10%, which may suggest the case of possible overfitting. They have trained 3D CNN using 5x5x5 patch-based so not a whole MRI itself. Conversely, we tested using the whole MRI and PETs in different data sizes, splitting the data in 5:2:3 ratios for training, validation, and testing. Hence, the 30% untouched data when tested can give us a reliable result.

In Table V, the term SPR is introduced, which indicates the use of multiple scans from a single patient, but not necessarily at the same time. As a result, multiple MRIs and PETs were acquired from a single patient for SPR greater than '1'; however, the image acquisition and preprocessing steps were different for each of the scans. A lower SPR value can bring variability in the dataset; therefore, the value of '1' indicates a single scan from a patient. This may eventually

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3040486, IEEE Access

bring generality in the trained model; however, this can result in a low performance due to the constraint of the limited training material as in our case with the MRI scans, where the accuracy dropped to 74.55% [φ] from our best outcome of 94.5% (see Table V)[ξ]. Later to check with the PET, we first trained it with a smaller database with scans from each unique patient (i.e. SPR=1); however, the results were poor. It was then tested with a larger PET database and a higher SPR. This also resulted in a low performance[ζ] that led us to conclude that PET is not a good choice for image-based 3D-CNN classification. On further tests with PET+MRI as presented in the last row of table V, we found a moderate result that is merely due to higher true positives from the MRI scans than from the PET. Thorough experiments were performed with a different number of subjects to find the effect of data-size in both MRIs and PETs; hence we did not use the same number of patients.

TABLE V
COMPARISON WITH OTHER ALGORITHMS WITH 3D CNN BASED ARCHITECTURE.

AUTHORS	METHOD	3D SCANS	# OF PATIENTS	SPR	TESTING ACCURACY %
EVGIN GOCERI [33]	SOBOLEV GRADIENT BASED OPTIMIZED CNN	TYPE: MRI CN: 568 AD: 570	CN: 130, AD: 185	4.36:3.08	98.06 (AD/CN)
GUPTA ET AL. [34]	SPARSE AUTO ENCODER (SAE) BASED CNN	TYPE: MRI CN:1278 AD: 755 MCI: 2282	CN: 232, AD: 200, MCI: 411	5.50:3.75:5.55	94.74 (AD/MCI/NC)
PAYAN AND MONTANA [35]	SPARSE AUTO ENCODER (SAE) PATCH BASED 3D CNN	TYPE: MRI CN:755 AD:755 MCI:755	CN:755, AD:755 MCI:755	1:1:1	89.47 (AD/MCI/HC)
HOSSEINI ASL ET AL. [14]	DSA-3D-CNN	TYPE: MRI -	CN:70, AD: 70	-	97.60 (AD VS. CN)
OH ET AL. [36]	INCEPTION AUTO ENCODER BASED 3D CNN	TYPE: MRI -	NC:230 AD:198 SMCI: 101	-	84.5% (AD VS. NC) 74.07% (AD VS. SMCI)
PROPOSED DIVNET	DIVERGING CNN	TYPE: MRI CN:305 AD:209 MCI:474	CN:60 AD:65 MCI:87	5.08:3.21:5.44	94.59% (AD/CN/MCI) [ξ]
		TYPE: MRI CN:75 AD:54 MCI:58	CN:31 AD:28 MCI:48	2.41:1.92:1.20	74.55% (AD/CN/MCI) [Φ]
		TYPE: PET CN: 109 AD: 102 MCI: 125	CN: 109 AD: 102 MCI: 125	1:1:1	66.6% (AD/CN/MCI)
		TYPE: PET CN: 1057 AD: 1165 MCI: 974	CN: 109 AD: 136 MCI: 337	9.69:8.56:2.89	50.21% (AD/CN/MCI) [ζ]
		TYPE: PET+MRI	CN: 414 AD: 311 MCI: 599	-	82.12% (AD/CN/MCI)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3040486, IEEE Access

A. PERFORMANCE ANALYSIS AND DISCUSSION

To study the proposed model performances listed in Table V, we visualized the convolutional layer as well as the FCL with the help of Pseudo-code 1, 2, and 3. The convolution results have been discussed earlier; here we will discuss the FCL output. FIGURE 6 depicts the distribution of the features for the test image set, which consists of 296 scans that are separated layer-wise during classification from the first convolution to the last FCL. The classification performance of the converging and diverging architecture is the best out of the four selected architectures (Table II). Even so on the basis of FCL patient-level visualization, as demonstrated in Figure 6, we see that the features for each class start to separate well in the diverging architecture than the converging one. From the first FCL FC_1 to the third FCL FC_3, the data visualization using t-SNE shows a better separation in the second case (i.e. diverging, see FIGURE 6). Similarly, based on the final FCL graph plotted as separate color curves for each cohort domain against the real weights of the final 100 parameters from the trained network without any projection (see FIGURE 8), shows a better demarcation between each colored graph than the 512 parameters from the U-net architecture. Afterward, we moved back to the training curve of these three networks (see FIGURE 7) to finalize the best performance. It was observed that the validation loss is significantly higher than the training loss in the converging and equivalent architecture. This indicates that the network can still be optimized, which was achieved with a diverging architecture and proper hyper-parameter selection.

Regarding the hyper-parameter selection, it is important to maintain proper and timely training and good performance of the trained model. Concerning our experiment, the L4 diverging architecture was the best among the selected architecture as specified in Tables II and III, whereas important hyper-parameters like the initialization, activation, and optimization algorithms were selected using Table III.

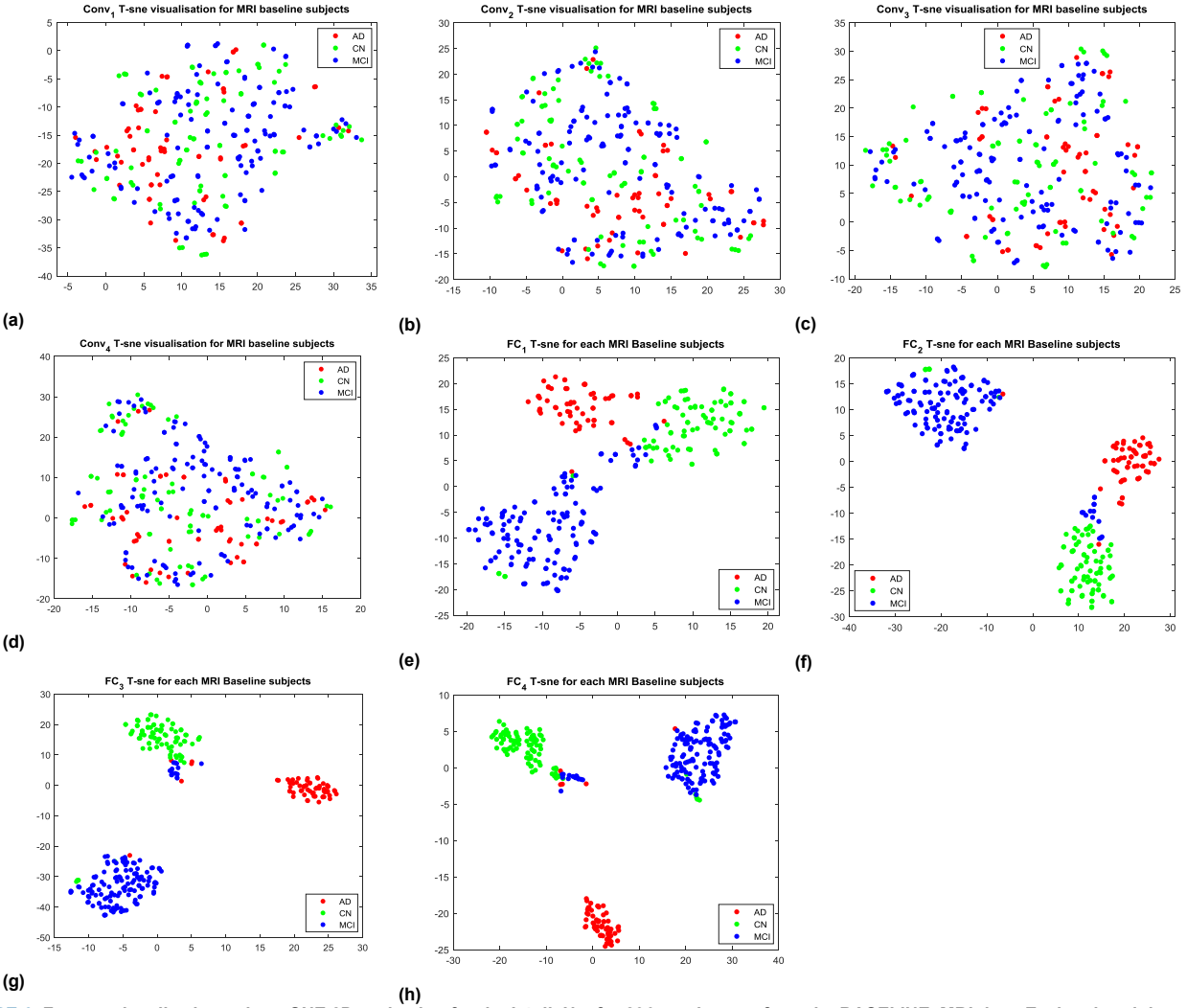


FIGURE 6. Feature visualization using t-SNE 2D projection for the L4 divNet for 296 test images from the BASELINE_MRI data. Each colored dot represents the feature of a single MRI of the indexed class. This starts from the 1st convolution to the 4th convolution (i.e. from Fig (a) to Fig (d)). The features from similar groups start to segregate, and it can be distinctly visualized from the 1st FCL (i.e. FC_1, Fig (e)). It continues until the last FCL (i.e. FC_4), where only a few colored dots are found in the wrong cluster (Fig (h) near the green CN group and a few in the blue MCI group). This overlapped region may be due to the possible false positives or false negative predictions that are subjected to errors in the test set prediction. The X-axis and Y-axis represent the values of the 1st dimension and 2nd dimension obtained from the t-SNE 2D projection respectively.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3040486, IEEE Access

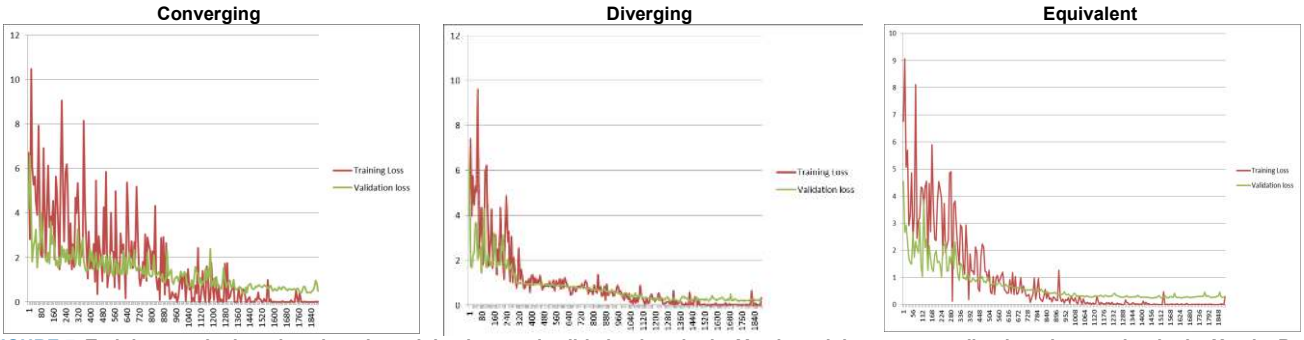


FIGURE 7. Training graph plotted against the training loss and validation loss in the Y-axis and the corresponding iteration number in the X-axis. By having more iteration numbers, the longer the epochs are. The training plot of the converging architecture has a validation loss that is much higher than the training loss. This may cause a poor performance, which is similar in the case with an equivalent architecture. However, the validation loss is quite reduced in the diverging architecture; thus, making it the optimal choice. Here, the training material and the training environment are identical in all three cases.

B. Generalization and overfitting problem

If we look at the recent architectures [14] [33] [34] [35] and the performance results, we find that the reported precision and accuracy rate are very high, more than 90%. In MR-guided image acquisition, various technical specifications like acquisition instrument, spatial positioning, contrast intensity, plane orientation, registration template, correction method, and a wrapping protocol can bring variability in the MRI of the suspected class [41]. Hence, a neural network trained on one 'variety' of an MRI, may find it ambiguous to detect an MRI of the same target class, if this is acquired differently, this causes a generalization error in the network. The generalization error is one of the leading challenges in medical imaging diagnosis. In this case, we have tested our network/model with other data from the ADNI, which we denoted as MRI_adapted. This is because it was partly adapted from [40] which differs in participants under the ADNI project. The MRI_adapted dataset was used only for testing of the generalization, which consists of 135 AD, 162 CN, and 134 MCI 3D scans; the testing results are presented in FIGURE 9. The other way to scrutinize could be with the visualization of the features. By extracting the better features, CNN will learn better. Similarly, overfitting is a contemporary part that comes with the generalization error. A non-generalized model learns 'too well' so that it only memorizes the training pattern that causes overfitting. Once we solve the overfitting problem, generality is also achieved.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3040486, IEEE Access

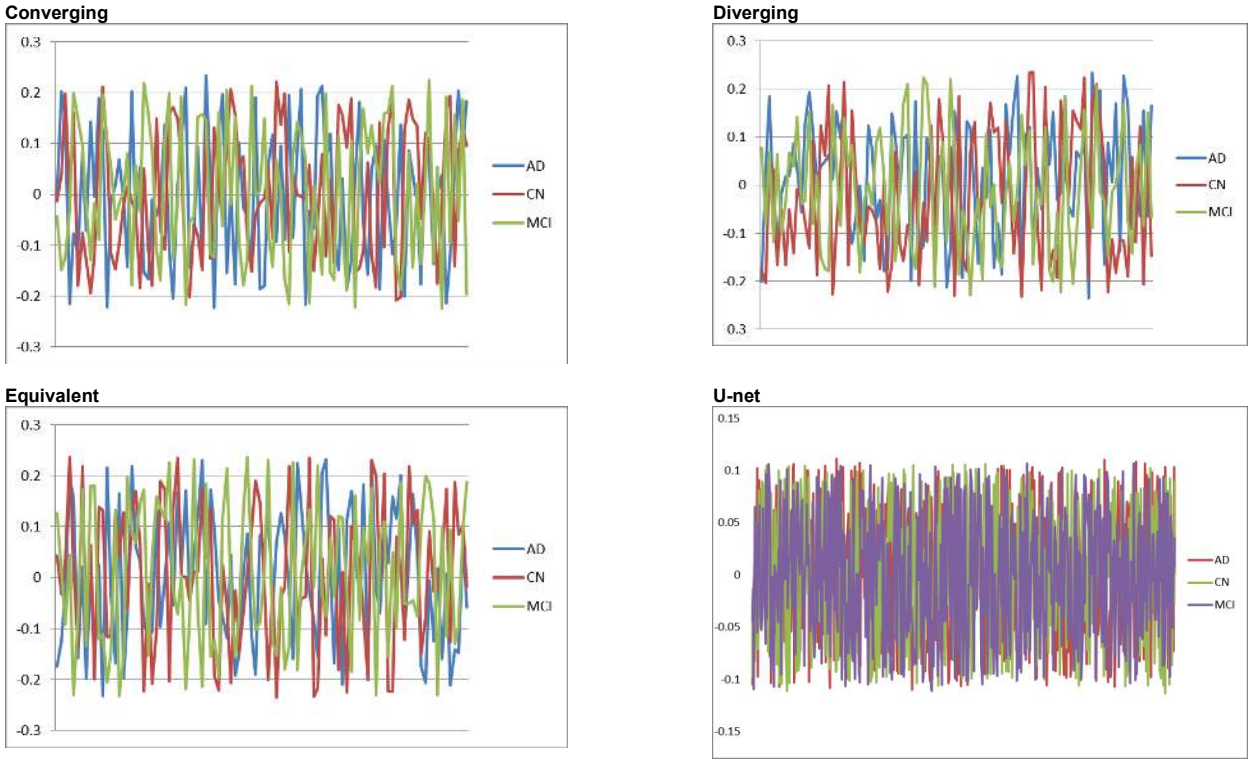


FIGURE 8. Final FCL weights values plotted in Y-axis directly for three target domains separately for each tested architecture using Pseudocode 3. X-axis extends from 0-100 for first 3 graphs whereas it extends from 0-512 in fig (d). The first three graphs have 100 parameters before producing the final three outputs for the softmax classifier whereas U-net has 512 parameters.

IX. CONCLUSION

CNN like ANN, itself is a semi-supervised learning algorithm that doesn't require prior heavy feature engineering, and its self-auto generic feature extraction property is already discussed in section II. Few researchers have been successful to develop optimization algorithm as [33], however, the most important contribution is the design of the better architectural unit itself [34] [35] [36], whether simple or complex, the result should be satisfactory and properly analyzed, which we think we have done to some extent. Besides, the prevailing techniques are mainly 2D image-based methodology, so the 3D architecture based concept is itself an initiative approach. This concluding section summarizes the key points that may be helpful for other researchers working with medical imaging in the same field with 3D CNN.

- The deep learning process heavily depends on the choice of training materials. Closely related images (training) can enhance the training performance; however, it can simultaneously 'spoil' the model due to overfitting. 'Good data' rather than 'big data' is required to generate a good network.
- Although our trained CNN is not deep enough to prototype a human brain structure, unlike reconstruction and segmentation models, it is however good enough to classify the MRIs, based on the segregated features learned in the convolutional layers, which is the actual aim of our study.
- MRI can be a better choice than PET for image-based CNN models. This may be due to its diverse pixel value of the MRI.
- Selection of hyper-parameters like initial-learn rate, learn-rate drop factor, the activation function and the initialization algorithm can affect the training process, but it has little effect on its performance once the convergence is achieved.
- The architecture and depth heavily affect the performance of the model thus, it is very important to have a generalized cum optimized model. In regards to the selection of features, we are convinced that the diverging window or reception area in each layer will be more beneficial than the contemporarily used converging or equivalent reception area.
- 'Overfitting' and 'generalization' problems are the biggest challenges for deep learning models.
- Since we have proposed an optimized DL based CNN for classification of AD, MCI, and NC using MRI/PET, it will assist the medical clinicians as an initial rapid test to identify the patient's condition using brain image scans only. Besides, MCI being an early stage of dementia means MCI identification will also help in the early prognosis of AD.

Based on our findings we hope this can be helpful in many ways to researchers working in the same field of MRI/PET

classification. Our study here is limited in the ADNI dataset, and may not act as universal CAD for AD detection yet more avenues are to be explored. The constantly developing deep learning methods can prove to make this process more optimal, robust, rapid, and automatic, with a minimum level of human supervision.

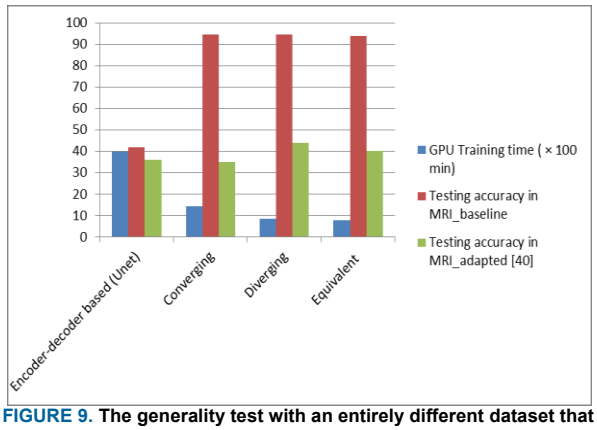


FIGURE 9. The generality test with an entirely different dataset that was not involved in training and was acquired from another ADNI project [40].

APPENDIX

Appendix 1 with a list of files for MRI and PET types in Table I along with demographic details in Table II. Also, a high-quality visio image for FIGURE 2 is presented.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2019R1A4A1029769, NRF-2019R1F1A1060166).

Data collection and sharing for this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf. ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: AbbVie, Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Araclon Biotech; BioClinica, Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir, Inc.; Cogstate; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Lumosity; Lundbeck; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Therapeutic Research Institute at the University of Southern California. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of Southern California. Correspondence should be addressed to Goo-Rak Kwon: grkwon@chosun.ac.kr.

REFERENCES

[1] 2017 Alzheimer's Disease Facts and Figures. Alzheimer's Disease Association, 2017.

[2] Apostolova, Liana G. "Alzheimer disease", Continuum: Lifelong Learning in Neurology 22, no. 2 Dementia (2016): 419.

[3] Huang, Yechong, Jiahang Xu, Yuncheng Zhou, Tong Tong, and Xiahai Zhuang. "Diagnosis of Alzheimer's Disease via Multi-modality 3D Convolutional Neural Network." Frontiers in Neuroscience 13 (2019): 509.

[4] Burton, Emma J., Ian G. McKeith, David J. Burn, E. David Williams, and John T. O'Brien. "Cerebral atrophy in Parkinson's disease with and without dementia: a comparison with Alzheimer's disease, dementia with Lewy bodies and controls", *Brain* 127, no. 4 (2004): 791-800.

[5] Hubel, David H., and Torsten N. Wiesel. "Receptive fields of single neurones in the cat's striate cortex." *The Journal of physiology* 148, no. 3 (1959): 574-591.

[6] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification", In *Proceedings of the IEEE international conference on computer vision*, pp. 1026-1034. 2015.

[7] https://medium.com/syncedreview/iclr-2019-fast-as-adam-good-as-sgd-new-optimizer-has-both-78e37e8f9a34 accessed 11th February 2020.

[8] Cheng, Danni, Manhua Liu, Jianliang Fu, and Yaping Wang. "Classification of MR brain images by combination of multi-CNNs for AD diagnosis." In *Ninth International Conference on Digital Image Processing (ICDIP 2017)*, vol. 10420, p. 1042042. International Society for Optics and Photonics, 2017.

[9] Liu, Manhua, Danni Cheng, Kundong Wang, Yaping Wang, and Alzheimer's Disease Neuroimaging Initiative. "Multi-modality cascaded convolutional neural networks for Alzheimer's disease diagnosis." *Neuroinformatics* 16, no. 3-4 (2018): 295-308.

[10] Liu, Manhua, Danni Cheng, Weiwu Yan, and Alzheimer's Disease Neuroimaging Initiative. "Classification of Alzheimer's disease by combination of convolutional and recurrent neural networks using FDG-PET images." *Frontiers in neuroinformatics* 12 (2018): 35.

[11] Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." In *European conference on computer vision*, pp. 818-833. Springer, Cham, 2014.

[12] Donahue, Jeff, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. "Decaf: A deep

convolutional activation feature for generic visual recognition." *In International conference on machine learning*, pp. 647-655. 2014.

[13] Sharif Razavian, Ali, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. "CNN features off-the-shelf: an astounding baseline for recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 806-813. 2014.

[14] Hosseini-Asl E, Ghazal M, Mahmoud A, Aslantas A, Shalaby AM, Casanova MF, et al. "Alzheimer's disease diagnostics by a 3D deeply supervised adaptable convolutional network", *Front Biosci*. 2018;23:584–596

[15] Wang SH, Zhang Y, Li YJ, Jia WJ, Liu1 FY, Yang MM. "Single slice based detection for Alzheimer's disease via wavelet entropy and multilayer perceptron trained by biogeography-based optimization", *Multimed Tools Appl*. 2016;77:1-25.

[16] Yang G, Zhang Y, Yang J, et al. "Automated classification of brain images using wavelet-energy and biogeography-based optimization." *Multimed Tools Appl*. 2015;75(23):15601-15617.

[17] Jha, Debesh, Ji-In Kim, and Goo-Rak Kwon. "Diagnosis of Alzheimer's disease using dual-tree complex wavelet transform, PCA, and feed-forward neural network." *Journal of healthcare engineering* 2017 (2017).

[18] Khagi, Bijen, Goo Rak Kwon, and Ramesh Lama. "Comparative analysis of Alzheimer's disease classification by CDR level using CNN, feature selection, and machine learning techniques." *International Journal of Imaging Systems and Technology* 29, no. 3 (2019): 297-310.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet classification with deep convolutional neural networks", Proc. NIPS, 2012; 1097–1105.

[20] He, K., Zhang, X., Ren, S. & Sun, J. "Deep residual learning for image recognition", In *Proc. IEEE Conference on Computer Vision and Pattern Recognition,* 770–778 (IEEE, 2016)

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions", Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2015; 1–9.

[22] Litjens, Geert, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I. Sánchez. "A survey on deep learning in medical image analysis", *Medical image analysis* 42 (2017): 60-88.

[23] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." In *Advances in neural information processing systems*, pp. 91-99. 2015.

[24] He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. "Mask r-cnn." In *Proceedings of the IEEE international conference on computer vision*, pp. 2961-2969. 2017.

[25] https://medium.com/finc-engineering/cnn-do-we-need-to-go-deeper-afe1041e263e accessed 11th February 2020.

[26] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.

[27] Karpathy, Andrej. "Connecting images and natural language." PhD diss., Ph. D. thesis, Stanford University, 2016.

[28] Pereira, Sérgio, Adriano Pinto, Victor Alves, and Carlos A. Silva. "Brain tumor segmentation using convolutional neural networks in MRI images." IEEE transactions on medical imaging 35, no. 5 (2016): 1240-1251.

[29] Sajjad, Muhammad, Salman Khan, Khan Muhammad, Wanqing Wu, Amin Ullah, and Sung Wook Baik. "Multi-grade brain tumor classification using deep CNN with extensive data augmentation", Journal of computational science 30 (2019): 174-182.

[30] https://qz.com/1034972/the-data-that-changed-the-direction-of-ai-researchand-possibly-the-world/. Accessed January 8, 2019

[31] Krizhevsky, Alex, and Geoffrey Hinton. "Learning multiple layers of features from tiny images." (2009): 7.

[32] Kiranyaz, Serkan, Turker Ince, and Moncef Gabbouj. "Real-time patient-specific ECG classification by 1-D convolutional neural networks." *IEEE Transactions on Biomedical Engineering* 63, no. 3 (2015): 664-675.

[33] Goceri, Evgin. "Diagnosis of Alzheimer's disease with Sobolev gradient based optimization and 3D convolutional neural network", International journal for numerical methods in biomedical engineering 35, no. 7 (2019): e3225.

[34] Gupta, Ashish, Murat Ayhan, and Anthony Maida. "Natural image bases to represent neuroimaging data", In International conference on machine learning, pp. 987-994. 2013.

[35]Payan, Adrien, and Giovanni Montana."Predicting Alzheimer's disease: a neuroimaging study with 3D convolutional neural networks". *In: ICPRAM*. 355–362 (2015).

[36] Oh, Kanghan, Young-Chul Chung, Ko Woon Kim, Woo-Sung Kim, and Il-Seok Oh. "Classification and Visualization of Alzheimer's Disease using Volumetric Convolutional Neural Network and Transfer Learning", *Scientific Reports* 9, no. 1 (2019): 1-16.

[37] https://en.wikipedia.org/wiki/Cross-validation_(statistics)#k-fold_cross-validation accessed on 11th February, 2020

[38] Nguyen, Anh, Jason Yosinski, and Jeff Clune. "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images", In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 427-436. 2015.

[39] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234-241. Springer, Cham, 2015.

[40] Cuingnet, Rémi, Emilie Gerardin, Jérôme Tessieras, Guillaume Auzias, Stéphane Lehéricy, Marie-Odile Habert, Marie Chupin, Habib Benali, Olivier Colliot, and Alzheimer's Disease Neuroimaging Initiative. "Automatic classification of patients with Alzheimer's disease from structural MRI: a comparison of ten methods using the ADNI database." *neuroimage* 56, no. 2 (2011): 766-781

[41] Baert,A. L., R. W. Günther, and G. K. von Schulthess. *Interventional magnetic resonance imaging*. Springer Science & Business Media, 2012.

[42] https://ida.loni.usc.edu/home/projectPage.jsp?project=ADNI accessed 11th Februray 2020.

Bijen Khagi received his B.Eng. from Purbhanchal University, Nepal, and is a graduate research student in the Department of Information and Communication Engineering, Chosun University, in the Ph.D. program. His main research interests include machine learning, computer vision application, image processing, and deep learning.

Goo Rak Kwon received his M.S. degree from the School of Electrical and Computer Engineering, SungKyunKwan University, in 1999, and the Ph.D. degree from the Department of Mechatronic Engineering, Korea University, in 2007. He has been a professor at Chosun University, since 2017. He has contributed 54 and 81 articles to journals and conference proceedings, respectively. He also holds 27 patents on the medical image analysis and the security of multimedia contents for digital rights management. He is a member of IEEE, IEICE, and IS&T in the international institute. And also he was an editorial board member of J. imaging in MDPI, an associate editor of KIPS, and an editorial board member of KMMS. His research interests include medical image analysis, A/V signal processing, video communication, and application