A Deep Learning Pipeline to Classify Different Stages of Alzheimer's Disease From fMRI Data

Yosra Kazemi
Dept. of Computer Science
Brock University
St. Catharines, ON, Canada
Email: yk15ff@brocku.ca

Sheridan Houghten
Dept. of Computer Science
Brock University
St. Catharines, ON, Canada
Email: shoughten@brocku.ca

***Abstract*—Alzheimer's disease (AD) is an irreversible, progressive neurological disorder that causes memory and thinking skill loss. Many different methods and algorithms have been applied to extract patterns from neuroimaging data in order to distinguish different stages of Alzheimer's disease (AD). However, the similarity of the brain patterns in older adults and in different stages makes the classification of different stages a challenge for researchers.**

In this paper, convolutional neuronal network architecture AlexNet was applied to fMRI datasets to classify different stages of the disease. We classified five different stages of Alzheimer's using a deep learning algorithm. The method successfully classified normal healthy control (NC), significant memory concern (SMC), early mild cognitive impair (EMCI), late cognitive mild impair (LMCI), and Alzheimer's disease (AD). The model was implemented using GPU high performance computing. Before applying any classification, the fMRI data were strictly preprocessed. Then, low to high level features were extracted and learned using the AlexNet model. Our experiments show significant improvement in classification. The average accuracy of the model was 97.63%. We then tested our model on test datasets to evaluate the accuracy of the model per class, obtaining an accuracy of 94.97% for AD, 95.64% for EMCI, 95.89% for LMCI, 98.34% for NC, and 94.55% for SMC.

I. Introduction

Alzheimer's disease (AD) is an irreversible, progressive neurological disorder that causes memory and thinking skill loss. The disease is a neurodegenerative type of dementia that begins with mild deterioration and gets progressively worse [16][25]. Reports show that more than 5 million Americans are living with Alzheimer's disease [2].

Alzheimer's disease progresses at various rates and each individual may experience different symptoms at different times [2]. Distinguishing different stages of the disease is usually a challenge for researchers because the between-class variance in different stages of Alzheimer's disease is low. Thus, scientists have become interested in studying the brain changes in Alzheimer's disease in order to have a better understanding of different stages and the changes that cause the disease.

Many neuroimaging tools and examinations are available to study the brain. Some of the most common brain imaging tools are magnetic resonance imaging (MRI), functional magnetic resonance imaging (fMRI), and positron emission tomography (PET) scan.

Machine learning methods have helped computer scientists to classify and distinguish some stages of Alzheimer's disease using neuroimaging data. However, distinguishing between different stages of Alzheimer's is not easy. Most previous works have classified the neuroimaging data into binary classes (AD vs. normal control) or into three classes (AD, mild cognitive impair, and normal healthy control). Different machine learning methods have been applied for classification, including conventional methods like the general linear model (GLM)[15], and multi-voxel methods such as support vector machines (SVMs) [5][3][4].

In the case of using fMRI scans as a diagnostic method for understanding brain changes and functionality in different areas, we need a very careful classifier. The typical machine learning approaches contain a feature selection step. However, in deep learning feature selection is an automatic process. This feature of the deep learning algorithm has caused a significant improvement in the accuracy of learning methods. Besides, it will remove the level of subjectivity (selecting the feature to use) from the previous methods [18].

In this study, previously successful deep learning methods [18][22] are used to classify more classes of Alzheimer's disease. Our classifier in this study is able to classify five different stages of the disease: normal control (NC), significant memory concern (SMC), early mild cognitive impair (EMCI), late mild cognitive impair (LMCI), and Alzheimer's disease (AD). We used the deep learning model AlexNet [9] to classify our large dataset into different stages, and to the best of our knowledge we are the first to do so.

II. Previous Work

As Alzheimer's disease is one of the main reasons for death in the United States [2] researchers have become interested in understanding its causes, and in the prediction of the disease by monitoring the brain changes during its progression. In computer science studies, researchers have become interested in studying the classification and prediction of the disease.

The researchers in the studies described in [26] [27] classified Alzheimer's disease (AD), mild cognitive impair (MCI), and MCI converter (those individuals whose stage of disease changed to AD after 15 months) using the ADNI dataset [1] for both MRI and PET imaging. Their data was acquired from 93

AD subjects, 76 MCI converters (MCI-C) and 128 MCI non-converters (MCI-NC), and 101 NC subjects. Before applying the classification method the MRI data was preprocessed by the typical procedures of skull-stripping, cerebellum removal, and spatial normalization to standard data. They used MIPAV software and FAST in FSL package for preprocessing the data. They designed an auto-encoder network for extracting features from images. SVM classification was applied as a learning method. Their method achieved accuracies of 95.35% (AD vs. NC), 85.67% (MCI vs. NC), and 75.92% (MCI-C vs. MCI-NC).

In 2015, Siqi Liu et. al. [12] implemented a new multi modal feature extraction for multi class Alzheimer's disease (AD) diagnosis. In their model, in order to save all the information and features in the data, a zero-masking strategy was applied. To extract higher-level features they used stacked auto-encoder (SAE) networks. SVM classification and multi-modal feature extraction algorithms were applied. The best result of their study was 86.86% accuracy.

Despite the fact that use of SVM classifiers in fMRI problems improved the accuracy of the classifier, it was not always a better solution. The possible reason is that fMRI data is considered "big data". Also, SVMs need a feature selection step which is time consuming [18].

Payan et al. [17] developed an algorithm for classification of Alzheimer's disease into three classes: Alzheimer's disease (AD), mild cognitive impair (MCI), and normal control (NC) subjects. They applied a 3D convolutional neural network with an auto-encoder and 2D CNNs on their datasets. They achieved an accuracy of 89.47% for the 3D classification model. Their results for the 2D CNN model were the same as the results that Siqi Liu et. al. [13] obtained from their 2D experiments, which was approximately 85.53% accuracy.

Another, and the most recent, study on classifying Alzheimer's disease was performed by Saman Sarraf et. al. [21] [22]. They designed a pipeline to classify both fMRI and MRI data of Alzheimer's disease subjects and normal control subjects. They applied two different methods for their binary classifiers. The CNN architectures that were applied included LeNet-5 [10] and GoogLeNet [28]. They achieved outstanding results using both models. An average accuracy of 99% and 100% were acquired respectively in the LeNet and GoogLeNet models for fMRI data.

III. Methods

A. Data Acquisition

In this study, a subset of the Alzheimer's Disease Neuroimaging Initiative (ADNI) database [1] was used to train and validate our convolutional neural network (CNN) classifier. This subset included resting-state functional magnetic resonance imaging (rs-fMRI) scans of 197 subjects with average age > 74. Based on their mini mental state examination (MMSE), these subjects were in different stages of disease. As stated in [1] "The purpose of the ADNI study is to track the progression of the disease using biomarkers to assess the brain's structure and function over the course of four disease states". ADNI categorizes different stages of Alzheimer's disease for participants: Healthy normal control (NC), Significant Memory Concern (SMC), Early Mild Cognitive Impairment (EMCI), Late Mild Cognitive Impairment (LMCI), and Alzheimer's disease (AD) classes.

TABLE I
Number of Subjects and their Mean Age

Group	Subj.	Female	Mean Age	Male	Mean Age
NC	55	30	74.73	25	78.52
SMC	25	17	75.06	8	75.45
EMCI	46	29	71.66	17	74.24
LMCI	39	14	74.36	25	75.04
AD	29	16	72.16	13	75.45

The scans in ADNI were performed on two different Tesla scanners, namely Philips Medical systems and SIEMENS. Philips Medical system scans were obtained with an EPI sequence of 144 volumes, Field Strength=3.0 tesla, Flip Angle=80.0 degree, TE=30.0 ms, TR=3000.0 ms, 64×65 matrix, and 6720.0 slices of 3.31 mm thickness for Resting State fMRI. The EPI sequence of Extended Resting State fMRI with the Philips Medical system scanner was 200 volumes, Field Strength=3.0 tesla, Flip Angle=90.0 degree, TE=30.0 ms, TR=3000.0, 64×65 matrix, and 9600.0 slices of 3.31 mm thickness. For SIEMENS scanner the EPI sequence was 197 volumes, Field Strength=3.0 tesla, Flip Angle=80.0 degree, TE=30.0 ms, TR=2999.99, 448×448 matrix, and 197 slices of 3.4 mm thickness. We considered these differences during the preprocessing steps to avoid any influence on our results.

In this study we chose 197 subjects from NC, SMC, EMCI, LMCI, and AD participants. 107 of the participants were female and 90 were male with an average age of 74.4. We eliminated the MCI class since there were not enough subjects available in this class (only two). The fMRI data were available for ADNI GO and ADNI 2 projects. However, the only project with MCI individuals was ADNI 1. Table I shows the number of subjects in each category that we used in this study.

We obtained our dataset with the above description in the Digital Imaging and Communications in Medicine (DICOM) fMRI format. To analyze the data we needed to convert it to Neuroimaging Informatics Technology Initiative (NIfTI) format in order to work with the available neuroimaging toolbox, and then apply some preprocessing on the raw data. This will be described in the following section.

B. Data Preprocessing

In order to take raw data from the scanner and prepare it for analysis, preprocessing is the most important step. Preprocessing is a series of data transformations to reduce the noise in the raw data. Preprocessing increases the sensitivity of analysis (SNR) and certifies the validity of the statistical model. As previously mentioned, the raw fMRI data is in DICOM format. However, the required format in most fMRI analysis tools is the NIfTI format. We first converted the

DICOM data to NII format by using the dcm2nii toolbox developed by Chris Roden et al. [19].

1) Brain extraction: Since we are studying brain tissue, we need to remove the non-brain areas such as skull and neck voxels from fMRI data. The FSL-BET toolbox [24] was used to extract the brain area from fMRI images. The next step in preprocessing is motion correction. It is common for subjects to move their heads during a fMRI session. Within the functional images over time, the motion changes the position of the brain. Therefore, over time a voxel's time series may not refer to the same point of the brain. Despite the scale of the motion (sub-voxel motion or obvious motion), this can have destructive effects on the statistical analysis. Thus, motion correction is a very important step in fMRI data preprocessing. In this part of the study, we used the FSL-MCFLIRT toolbox [6]. The FSL-MCFLIRT tool applies rigid-body transformation for motion correction.

2) Slice timing corrections: The fMRI volumes are scanned one slice at a time. The timing of this acquisition is equally spread over the repetition time (TR= 3000 ms). Therefore, each voxel is scanned at a different time. The changes in the timing should be correctly justified to match the stimulus and response timing so the statistical analysis will be able to fit the model more accurately. The purpose of the slice timing correction is to justify the voxel time series to have a reference timing for all voxels. Slice timing corrections use some form of interpolation to shift the time series of values either forward or backward in time to achieve the temporal adjustment. In this study we used Hanning- windowed Sinc interpolation [11].

3) Spatial smoothing: At this point in preprocessing, the spatial smoothing of each volume is performed. The aim of performing spatial smoothing is to reduce the noise level while protecting the underlying signal. In order to protect the underlying signal from being reduced along with the noise, the extent of the spatial smoothing should not be larger than the size of the activated region. The most common way of smoothing is deconvoluting 3D images with a 3D Gaussian filter. The degree of smoothing is proportional to the full width at half-maximun (FWHM) of the Gaussian distribution. In this work, we performed spatial smoothing by using a Gaussian kernel of 5mm FWHM.

4) High-Pass filtering: To remove low level unwanted signals in the voxels' time series we applied high pass filtering. These low level noises could be the results of some physiological artefacts such as breathing, heartbeats, or by physical noises like scanner drifts. It is necessary to remove these low level noises since they will appear in the statistical analysis later and affect the fitting of the data to the model. The signal drifts have a low frequency. Therefore, we can remove them by applying a high-pass filter. This filter allows the high frequencies (stimulus activities) to pass, but removes low frequencies (the signal drifts). In this work, we used temporal high-pass filter with 0.01 HZ cut-off frequency.

5) Spatial normalization: The next step in fMRI preprocessing is spatial normalization. During the fMRI session, we have a particular high-resolution structural scan (T1-weighting (T1w)) according to the subject's brain shape and layout. However, to analyze the data of many subjects we need to have the same conditions. Only with equivalent brains (common spatial domain) are we able to compare changes in specific voxels in particular areas for all subjects. Registration of our anatomical brain to a template as a process in the fsl FLIRT toolbox [24][7] can perform the alignment of fMRI scans to a common spatial domain (reference template).

For spatial normalization first we need to put our fMRI data in T1w space by using FSL's FLIRT [24][7]. We used a linear transformation with 7 degrees of freedom (7 DOF). Then we registered MNI152 standard space (derived from average of 152 structural images after high-dimensional nonlinear registration) as our template. The MNI152 is a common template provided in FSL toolbox which has been used in previous works [20] [22] as well. To register the template we performed a linear transformation with wider degree of freedom (12 DOF) rather than the reregistration that we applied to create our own T1w space. This helps to fit our data all over the shape and the size by applying translations, rotations, zooms, and sheers.

As a result of these preprocessing steps, we obtained a 4D NIfTI file for each subject. Each of these NIfTI files contains $64 \times 45 \times 64$, 3D volume per time course.

6) Image Conversion: In order to create a proper dataset for the experiment we decomposed 4D NIfTI files into 2D matrices along z axis and time course. Next, 2D matrices were converted to lossless Portable Network Graphics (PNG) format using MATLAB's NIfTI and analyze image toolbox [23]. The last 3 slices of each time course for each subject were removed as they did not carry any functional information. The result of this conversion for each subject was 43 slices of 64×64 PNG images per time course. For example, the resting state fMRI data acquired by Philips system with 140 volumes (time course), $43 \times 140 = 6020$ images of size 64×64 was obtained for each individual. By converting NIfTI data for all subjects, 1,322,085 PNG images of size 64×64 were collected.

TABLE II
PERCENTAGES AND SIZES OF DATASETS

	Training	Validation	Testing
Percentage	60%	20%	20%
Images	793251	264417	264417

C. Our Experiments

Once the data had been converted, the images were randomly shuffled. To validate the classification, five-fold cross validation against the whole dataset was applied and five subsets were randomly created. In each experiment, as shown in Table II, 60% of the data was used for training, 20% for validation and 20% for testing. Table III shows the number of images in each of the categories for our datasets.

As described in the previous section, the AlexNet Architecture was used as a CNN classifier in this study. First, we

TABLE III
NUMBER OF IMAGES IN EACH CATEGORY

	Normal	SMC	EMCI	LMCI	AD
Training	227,016	113,265	169,848	171,882	111,240
Validation	75,672	37,755	56,616	57,294	37,080
Testing	75,672	37,755	56,616	57,294	37,080

labeled our datasets into five classes (AD as 0, EMCI as 1, LMCI as 2, NC as 3, and SMC as 4).

In this study we used the Caffe Deep Learning framework[8] to train our model. To use Caffe, we first converted our datasets to Lightning Memory Mapped Database (LMDB) and resized images to 256×256 pixels. Then we generated the mean image of the training data and subtracted it from each input image to ensure that the mean of every feature pixel is zero.

D. Model and solver definition

In this subsection we explain how we implemented our model in the Caffe framework. We adjusted the AlexNet model for 30 epochs since the learning rate with our configuration (described below) after epoch 21 approaches zero. Then we used the Caffe Stochastic Gradient Descent (SGD) solver method. The solver generates the training and testing networks for, respectively, learning and evaluation purposes. It takes snapshots of the model and the solver states during optimization. In order to learn the model, during each iteration, the solver computes the output and loss by calling the network forwards, and computes the gradients by calling the network backwards. Based on the solver method it will update parameters differently. Once the parameters are updated, the solver will update its state based on the learning rate, method, and history.

The Stochastic Gradient Descent (SGD) solver method uses the following to update the weights W:

$$\begin{aligned} V_t + 1 &= \mu V_t - \alpha \bigtriangledown L(W_t) \\ W_t + 1 &= W_t + V_t + 1 \end{aligned}$$

where $\alpha \bigtriangledown L(W)$ is the negative gradient, V_t is the previous weight update, α is the learning rate, and μ is the momentum as hyper parameters. This is as Krizhevsky et al.[9] described in their model, for deep learning using SGD as the solver. The best value to initialize the learning rate α is close to 0.01, and then it drops by some constant value after each *stepsize* iterations. The common learning rate policy is a step which drops the learning rate by a factor of Gamma in each stepwise iteration. Usually in deep learning, the learning rate and momentum parameter are related. In this study, the momentum parameter is $\mu = 0.9$. The momentum smoothes the weight updates during the iterations, which makes the model more stable and fast. The momentum after many iterations multiplies the size of weight updates by a factor of $\frac{1}{1-\mu}$. Then, it is important to decrease the size of α in the case of increasing the momentum.

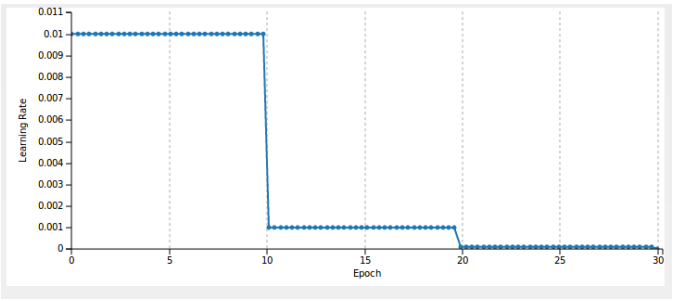

Fig. 1. Learning Rate drops by a factor of *gamma*.

```
test_iter: 8264
test_interval: 6198
base_lr: 0.01           # begin training at a learning rate of 0.01 = 1e-2
display: 40
max_iter: 185940        # train for 185940 iterations total
lr_policy: "step"       # learning rate policy: drop the learning rate in "steps"
                        # by a factor of gamma every stepsize iterations
gamma: 0.1              # drop the learning rate by a factor of 10
                        # (i.e., multiply it by a factor of gamma = 0.1)
momentum: 0.9
weight_decay: 0.0001
stepsize: 61361         # drop the learning rate every 61361 iterations
snapshot: 6198
snapshot_prefix: "snapshot"
solver_mode: GPU
net: "train_val.prototxt"
solver_type: SGD
```

Fig. 2. Snapshot of solver.prototxt file.

In this study, we initialized the hyper parameters for the AlexNet model in Caffe (.prototxt training and validation file) as *Gamma* $= 0.1$, *momentum* $= 0.9$, *Weight-decay* $= 0.001$, and *Learning rate* $= 0.01$. Figure 1 shows the policy applied for dropping the learning rate every *stepsize* iterations.

Figure 2 is a snapshot of the solver.prototxt file in Caffe, to show all parameter initialization we used. The architecture of our model after initializing parameters is shown in Figure 3.

We repeated our experiment five times on the Amazon Web Service (AWS). The configuration of AWS was Linux G2.8xlarge, with four NVIDIA GPUs each with 4G memory, 1,536 CUDA cores, and 32 Intel Xeon E5-2670 vCPUs. Each experiment was trained on our server for five to six days.

IV. RESULTS

After running our experiments for training an average accuracy of 97.64% was achieved. Table IV shows the accuracy of our model in each experiment. As shown, we obtained very good accuracy in all experiments.

Besides accuracy, we also monitored the loss in the training and testing datasets during the learning process. Figure 4 shows the accuracy of the first experiment as an example.

Classifying different stages of Alzheimer's disease needs accurate preprocessing and careful feature learning due to the similarity in brain anatomy in older adults and also similarity intensities in the images. As described earlier, we used a very careful preprocessing on our dataset in order to prevent any problem during image analysis. The preprocessing of fMRI data has a major influence on the high and consistent accuracy.

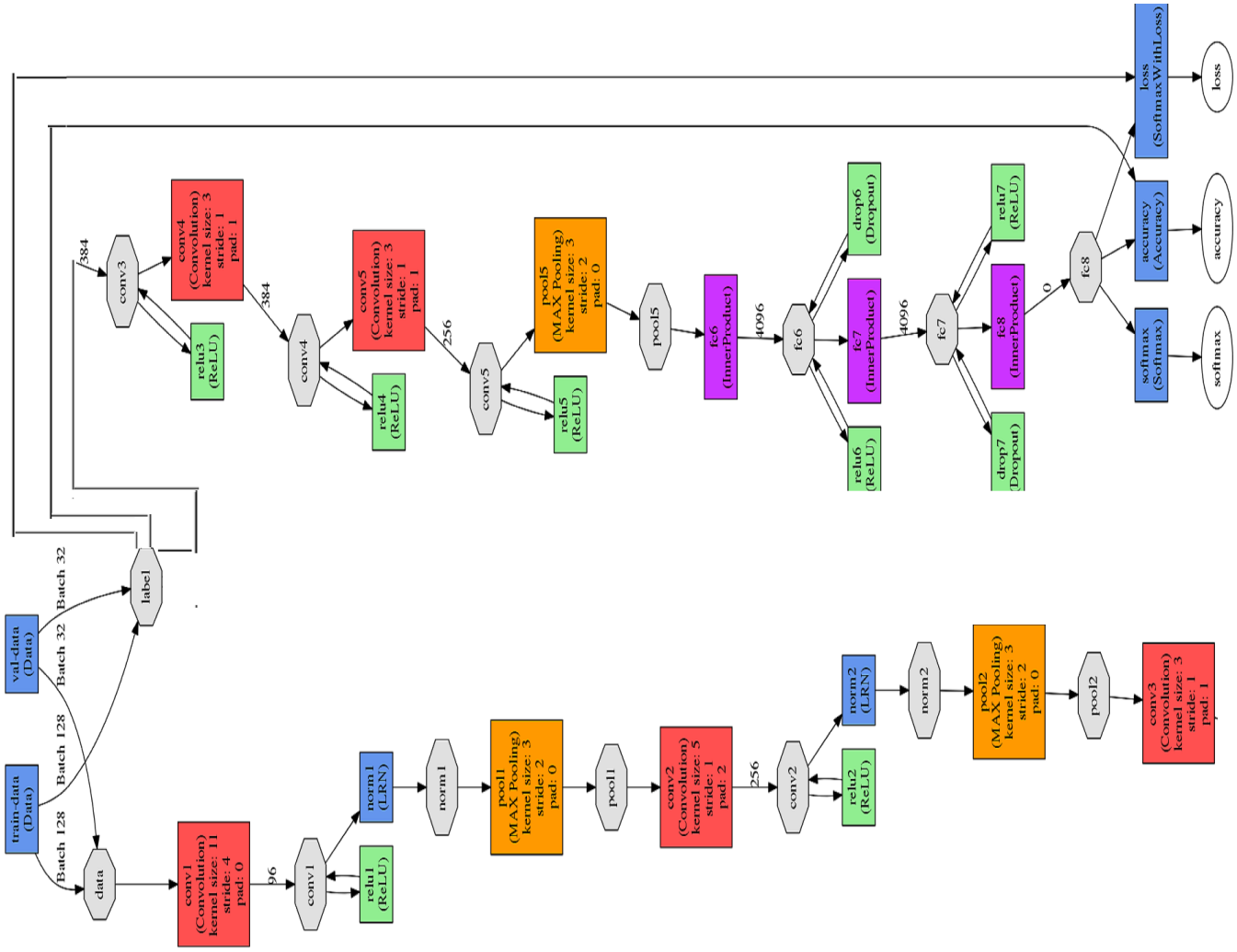

Fig. 3. AlexNet model for training fMRI datasets

TABLE IV
ACCURACY IN EACH EXPERIMENT AND AVERAGE ACCURACY

Experiments	1	2	3	4	5	Average
Accuracy	97.70	97.62	97.61	97.63	97.62	97.64

During classification, the AlexNet model was trained and tested with a large amount of data that we acquired from 4D fMRI images. A set of learnable filters was applied in the model to extract low to high level features from images.

V. VISUALIZATION OF THE RESULTS

The results show that CNN architectures such as AlexNet are powerful models to classify different stages of Alzheimer's disease. In order to have a better understanding of the model and the convolutional layers, we visualized the weights filters and the statistical information of them in each layer of the model for many samples of our test dataset. The first visualization technique we used was monitoring the activations of the network during the forward pass. In the AlexNet model, which uses the ReLU method, the activations are more shapeless and dense at the beginning and they become more localized and sparse after the training progresses. This visualization can help us to check if some activation maps become all zero for different inputs. In the cases in which the activation maps became all zero we are facing a problem called "dead filters" which may be caused by high learning rates. This problem will result in a poor learning model and failure in classification.

In our model, since the first slices did not have enough functional information we noticed that activations have more zero values, which was expected. However, with enough functional information we monitored better activation behavior.

To show samples of both successful and failed classification in the model, we chose two different images from our test dataset. The first sample (as failure in prediction) was the image of an SMC patient in the second slice and over a time course of 136, and is shown in Figure 5. After using the trained model on this sample the result of prediction was

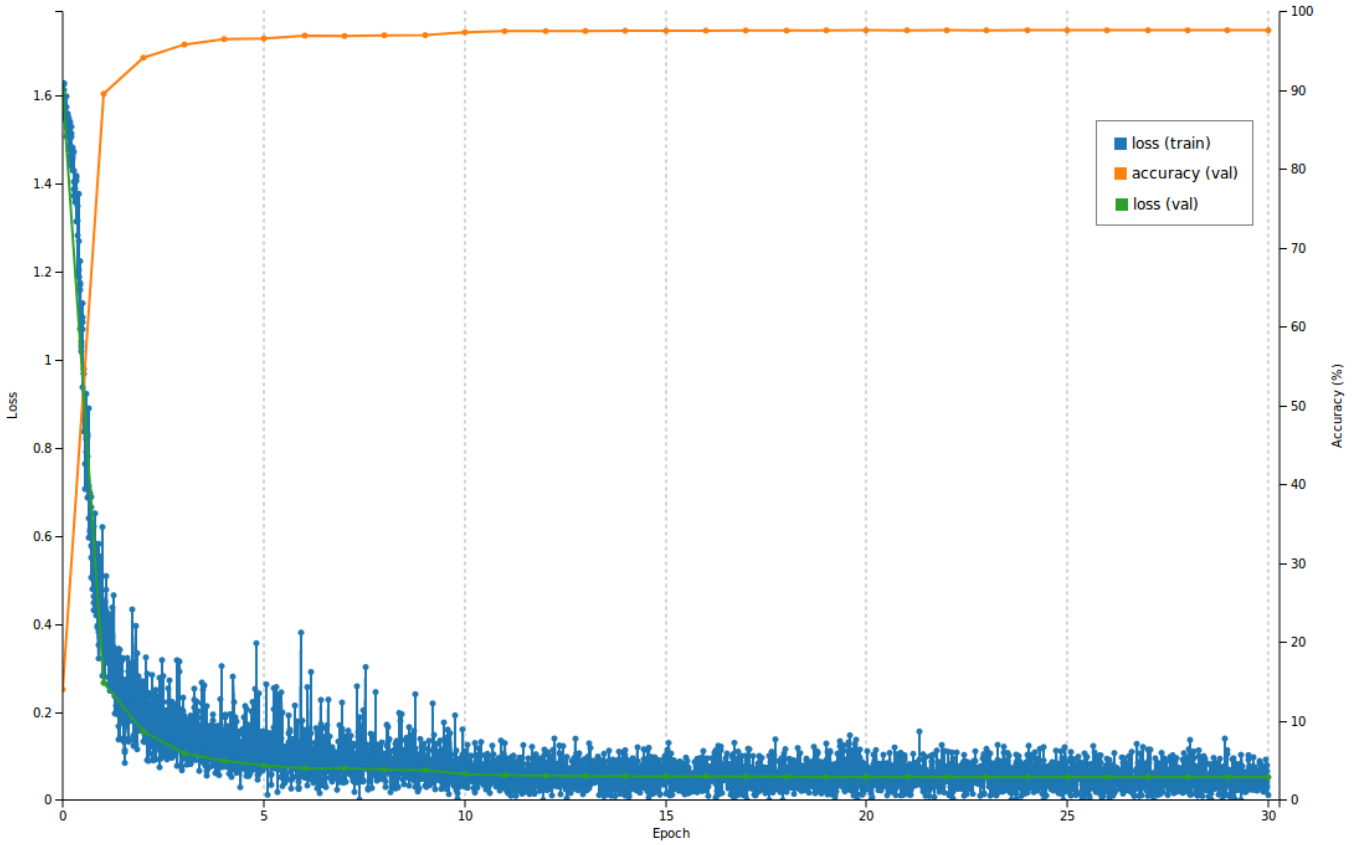

Fig. 4. The accuracy and loss of the first experiment over 30 epochs.

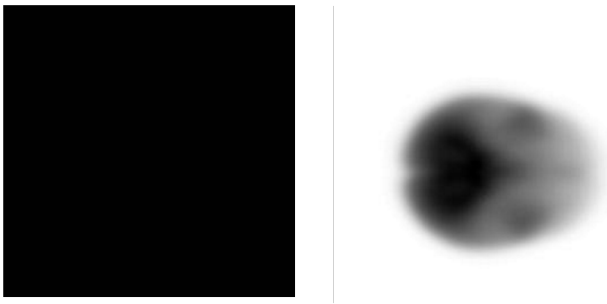

Fig. 5. Second slice of fMRI scan on time course of 136 for a SMC patient (left). The activation data as an input for the model. The size of the filter applied in this part was 227×227.

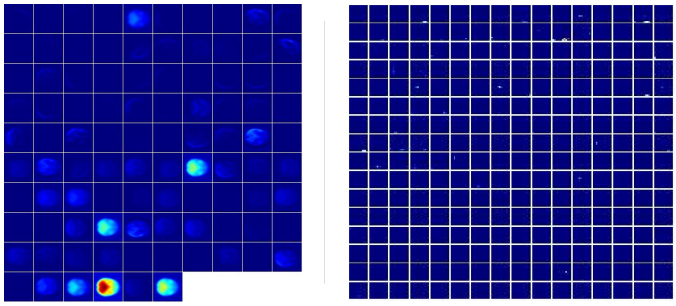

Fig. 6. In the first CONV layer of AlexNet in a given trained model 96 filters of 55×55 pixels were applied and visualized for a SMC patient's sample (left). In the fifth CONV layer of our model 256 filters of 13×13 were applied (right).

28.18% NC, 22.31% LMCI, 21.61% EMCI, 17.63% AD, and 10.27%. Figure 6 shows the activation maps of this image in the first layer (left) and fifth layer (right). As seen in Figure 6, because of a lack of functional information in our data, the model failed to predict the result correctly. The visualization shows that the number of dead filters in this case is significant.

However, the rest of the results for most of the other images in our dataset show how our model is able to predict different stages with high accuracy. Figure 7 is another example given to the trained model. In this case the image was in the LMCI category and the model predicted it as 100% LMCI. As seen in Figure 8, the number of dead filters is reduced significantly and as a result we obtain 100% accuracy in our prediction.

The second method to interpret the visualization of our CNN model is by visualizing the weights. As the first CONV layer of the network is working directly with raw data, we used this layer for our interpretation. Monitoring the weights of the network is useful since the less noisy and more smooth filters are, the more likely that our model trained at its best. Existence of noisy patterns is a sign that the model has not been trained

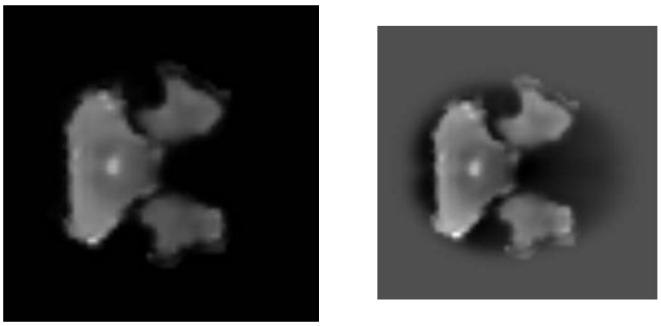

Fig. 7. Slice number thirty two of an fMRI scan on a time course of 172 for a LMCI patient (left). The activation data as an input for the model. The size of the filter applied in this part was 227×227

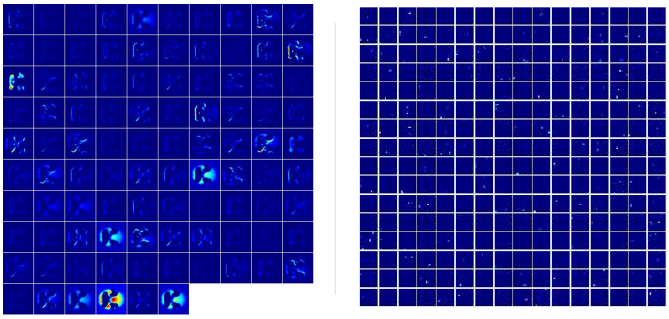

Fig. 8. In the first CONV layer of AlexNet in a given trained model. 96 filters of 55×55 pixels were applied and visualized for a LMCI patient's sample (left). In the fifth CONV layer of our model 256 filters of 13×13 were applied (right).

enough or that there might be some over fitting in the model. Figure 9 shows a sample of successful prediction of LMCI patients.

In our model all samples show smooth patterns without noise. This was an indication of acceptable performance of the trained model in our experiments.

VI. ROC CURVES AND ACCURACY OF THE MODEL FOR EACH CLASS

The next step we considered was studying the accuracy of our model for each class. For this purpose, we choose 264,000 images from our test datasets and applied the trained model. The result of this experiment was the accuracy of prediction of each class. For example, after giving an image from EMCI we obtained a 0.9837 probability that the image was classified as EMCI, and the probabilities of the image being a member of other classes were $NC = 0.0082$, $LMCI = 0.0066$, $SMC = 0.0011$, and $AD = 0.0004$. We used this information to create a confusion matrix for all of the 264,000 testing images. The value of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) were calculated for each class. Then, the accuracy of the performance of each class was calculated as $Accuracy = \frac{TP}{TP+FN}$.

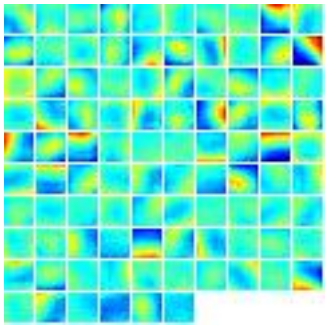

Fig. 9. In the trained AlexNet model, 96 filters of size of 11×11 pixels were applied and visualized for the first layer of our network.

TABLE V
CONFUSION MATRIX AND ACCURACY PER CLASS FOR TESTING 264,000 IMAGES IN OUR TRAINED MODEL

	AD	EMCI	LMCI	NC	SMC	Accuracy
AD	35148	159	386	1195	121	94.97%
EMCI	161	54130	327	1836	141	95.64%
LMCI	25	224	54809	2033	69	95.89%
NC	332	382	463	74276	76	98.34%
SMC	253	298	232	1271	35653	94.55%

Table V shows the confusion matrix and accuracy. The accuracy of the normal control class was higher. According to the number of images available in the normal control class, we expected better performance in this class. This is because deep learning methods perform better with larger datasets.

Last but not least, the receiver operating characteristic (ROC) curves were calculated for each class to validate the performance of our model. For each class TP, FP, TN, and FN values were calculated for 100 thresholds from 0 to 1 using the MATLAB Neural Network toolbox [14]. To draw curves, the sensitivity and specificity values were calculated, where *Sensitivity = True positive rate (TPR)* $= \frac{TP}{(TP+FN)}$ and *Specificity = True negative rate (TNR)* $= \frac{TN}{(FP+TN)}$. The axes of the ROC curve are Sensitivity (i.e. true positive rate) and $1-$ Specificity (i.e. false positive rate).

Figure 10 is an illustration of the ROC curves for all classes. The performance of the model for each class is tending towards the ideal classification. Also, for all of the classes the performance was significantly better than a random guess.

We also calculated the area under the curve (AUC) of the ROC curves. For SMC class the AUC value was 0.9334, for NC class it was 0.9486, 0.9500 for LMCI class, 0.9491 for EMCL class, and finally the AUC value calculated for AD class was 0.9422. This shows the consistency of our results.

VII. CONCLUSIONS

The goal of this paper was classification of all stages of Alzheimer's disease which was missing in previous works.

We used strict preprocessing steps on raw fMRI data obtained from the ADNI dataset, and applied the AlexNet CNN classifier on preprocessed data to classify different stages of Alzheimer's disease (NC, SMC, EMCI, LMCI, and AD). The low to high level features were learned during classification,

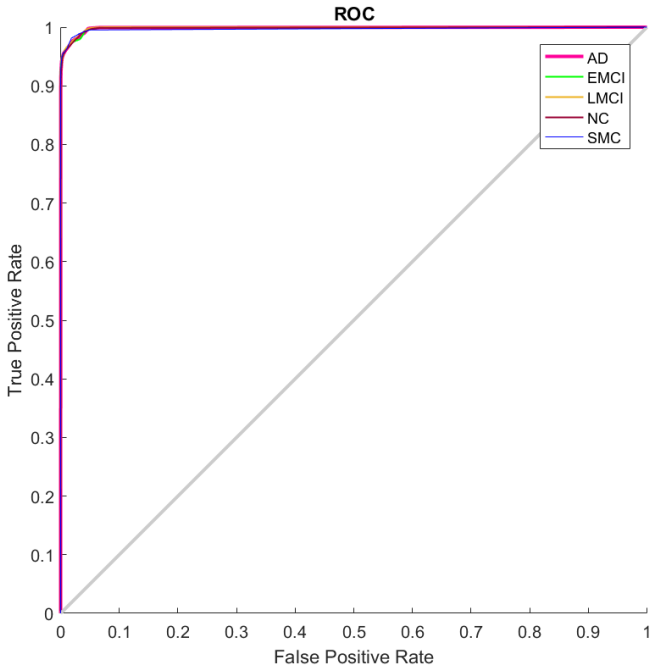

Fig. 10. ROC Curves for each class

resulting in an average accuracy of 97.64%. This is an outstanding result, compared to applications of other methods for classifying different stages of Alzheimer's disease. In addition, previous classification methods were applied to either AD vs. NC as binary classification or with three classes of NC vs. MCI vs. AD. This is, to the best of our knowledge, the first time that a deep learning algorithm has been applied to classifying Alzheimer's disease according to all other available stages.

As deep learning models are improving everyday, and applied in different fields, it is possible to use newer models such as VGG, googLeNet, ResNet, etc. to compare the results. Also, applying pre-trained models and transfer learning may improve results, as pre-trained models have shown outstanding results in medical imaging studies [29].

Acknowledgements

This work was funded in part by the Natural Sciences and Engineering Research Council of Canada.

References

[1] ADNI. Background and Rationale, http://adni.loni.usc.edu/study-design/background-rationale, 2017.

[2] Alzheimer's association. 2017 Alzheimer's disease facts and figures. Special Report, https://www.alz.org/documents_custom/2017-facts-and-figures.pdf.

[3] J. Carp, J. Park, T.A. Polk, and D.C. Park. Age differences in neural distinctiveness revealed by multi-voxel pattern analysis. *NeuroImage*, 56(2):736–743, 2011.

[4] M.N. Coutanche, S.L. Thompson-Schill, and R. T. Schultz. Multi-voxel pattern analysis of fMRI data predicts clinical symptom severity. *NeuroImage*, 57(1):113–123, 2011.

[5] F. De Martino, G. Valente, N. Staeren, J. Ashburner, R. Goebel, and E. Formisano. Combining multivariate voxel selection and support vector machines for mapping and classification of fMRI spatial patterns. *NeuroImage*, 43(1):44–58, 2008.

[6] M. Jenkinson, P. Bannister, M. Brady, and S. Smith. Improved Optimization for the Robust and Accurate Linear Registration and Motion Correction of Brain Images. *NeuroImage*, 17(2):825–841, 2002.

[7] M Jenkinson and S Smith. A global optimisation method for robust affine registration of brain images. *Medical image analysis*, 5(2):143–56, 2001.

[8] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. 2014.

[9] A. Krizhevsky, I. Sutskever, and G.E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[11] T.M. Lehmann, C. Gonner, and K. Spitzer. Survey: interpolation methods in medical image processing. *IEEE Transactions on Medical Imaging*, 18(11):1049–1075, 1999.

[12] S. Liu, S. Liu, W. Cai, H. Che, S. Pujol, R. Kikinis, D. Feng, M.J. Fulham, and ADNI. Multimodal Neuroimaging Feature Learning for Multiclass Diagnosis of Alzheimer's Disease. *IEEE Transactions on Biomedical Engineering*, 62(4):1132–1140, 2015.

[13] S. Liu, S. Liu, W. Cai, S. Pujol, R. Kikinis, and D. Feng. Early diagnosis of Alzheimer's disease with deep learning. In *2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI)*, pages 1015–1018. IEEE, 2014.

[14] Mathworks. Neural Network Toolbox - MATLAB: (R2016b), https://www.mathworks.com/products/neural-network.html, 2016.

[15] M.M. Monti. Statistical Analysis of fMRI Time-Series: A Critical Review of the GLM Approach. *Frontiers in Human Neuroscience*, 5:28, 2011.

[16] National Institute on Aging. Alzheimer's Disease Fact Sheet, https://www.nia.nih.gov/health/alzheimers-disease-fact-sheet.

[17] A. Payan and G. Montana. Predicting Alzheimer's disease: a neuroimaging study with 3D convolutional neural networks. *arXiv preprint arXiv:1502.02506*, 2015.

[18] S.M. Plis, D.R. Hjelm, R. Salakhutdinov, E.A. Allen, H.J. Bockholt, J.D. Long, H.J. Johnson, J.S. Paulsen, J.A. Turner, and V.D. Calhoun. Deep learning for neuroimaging: a validation study. *Frontiers in neuroscience*, 8:229, 2014.

[19] C. Roden. NITRC: dcm2nii: File Release Notes and Changelog, https://www.nitrc.org/frs/shownotes.php?release_id=3336.

[20] S. Sarraf and J. Sun. Functional Brain Imaging: A Comprehensive Survey. *arXiv preprint arXiv:1602.02225*, 2016.

[21] S. Sarraf and G. Tofighi. Classification of Alzheimer's Disease using fMRI Data and Deep Learning Convolutional Neural Networks. *arXiv preprint arXiv:1603.08631*, 2016.

[22] S. Sarraf and G. Tofighi. Deep Learning-based Pipeline to Recognize Alzheimers Disease using fMRI Data. *bioRxiv*, 2016.

[23] J. Shen. Tools for NIfTI and ANALYZE image - File Exchange - MATLAB Central, https://www.mathworks.com/matlabcentral/fileexchange/8797-tools-for-nifti-and-analyze-image, 2014.

[24] S.M. Smith. Fast robust automated brain extraction. *Human Brain Mapping*, 17(3):143–155, 2002.

[25] R.A. Stelzma, H.N. Schnitzlein, and F.R. Murtagh. An English Translation of Alzheimer's 1907 Paper, "Uber eine eigenartige Erlranliung der Hirnrinde". *Clinical Anatomy*, 8:429–431, 1995.

[26] H-I. Suk, S-W. Lee, D. Shen, ADNI, et al. Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis. *NeuroImage*, 101:569–82, 2014.

[27] H-I. Suk and D. Shen. Deep learning-based feature representation for AD/MCI classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 583–90, 2013.

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

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