Fuzzy Systems and Data Mining VI
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA200689

The Diagnosis of Alzheimer's Disease: An Ensemble Approach

Jingyan Qiu[a], Linjian Li[a], Yida Liu[a], Yingjun Ou[a], Yubei Lin[a,1]

[a] *School of Software Engineering, South China University of Technology, Guangzhou, China*

Abstract. Alzheimer's disease (AD) is one of the most common forms of dementia. The early stage of the disease is defined as Mild Cognitive Impairment (MCI). Recent research results have shown the prospect of combining Magnetic Resonance Imaging (MRI) scanning of the brain and deep learning to diagnose AD. However, the CNN deep learning model requires a large scale of samples for training. Transfer learning is the key to enable a model with high accuracy by using limited data for training. In this paper, DenseNet and Inception V4, which were pre-trained on the ImageNet dataset to obtain initialization values of weights, are, respectively, used for the graphic classification task. The ensemble method is employed to enhance the effectiveness and efficiency of the classification models and the result of different models are eventually processed through probability-based fusion. Our experiments were completely conducted on the Alzheimer's Disease Neuroimaging Initiative (ADNI) public dataset. Only the ternary classification is made due to a higher demand for medical detection and diagnosis. The accuracies of AD/MCI/Normal Control (NC) of different models are estimated in this paper. The results of the experiments showed that the accuracies of the method achieved a maximum of 92.65%, which is a remarkable outcome compared with the accuracies of the state-of-the-art methods.

Keywords. Alzheimer's disease, Deep learning, Ensemble learning, Transfer learning, Convolutional Neural Network

1. Introduction

Alzheimer's Disease is the most common cause of dementia [1], with the symptoms of memory loss, difficulty in speaking and execution barrier [2]. The crude prevalence of AD in China has been found to range between 7 per 1000 people to 66 per 1000 individuals and the estimation is that there are 9.5 million patients [3].

These days, deep learning and transfer learning accelerate the development of computer vision, thus boosting the efficiency and effectiveness of the diagnosis of Alzheimer's disease based on MRI images. Glozman et al. [4] used transfer learning that fine-tuned AlexNet architecture. 3D MRI images are essentially a stack of 2D images. Hon et al. [5] used an intelligent method to select slices on each subject's MRI image to select data with an entropy-based mechanism. They employed two

[1] Corresponding Author: Yubei Lin, South China University of Technology, Guangzhou, China; Email: yupilin@scut.edu.cn

architectures: VGG16 [6] and Inception V4 [7] through transfer learning and compared the performance of the two architectures. Jain R et al. [8] used a similar method to select slices on Alzheimer's Disease Neuroimaging Initiative (ADNI), which is established with the idea of enhancing the prevention and treatment of Alzheimer's disease (AD) [9] and used VGG16 for training analysis. It is worth noting that the previous two research groups ultimately focus on the performance of the models based on slice analysis instead of the analysis of a definite case of an individual, which may cause a test performance being too high.

In our study, unlike Hon et al. [5] and Jain R et al. [8], we divide our data set in the form of medical cases (in .nii format) ahead of pre-processing. This paper is organized as follows: At first, the pre-processing part will be introduced. After that, classification models and a probability-based fusion [10] will be laid out. Following the methodology and algorithm are the experiments conducted respectively on DenseNet [11] and Inception V4, as well as the one on the fusion. We also compare our maximal result with other researchers' results on ternary classification. And eventually, we conclude our work.

2. Methodology and Algorithm

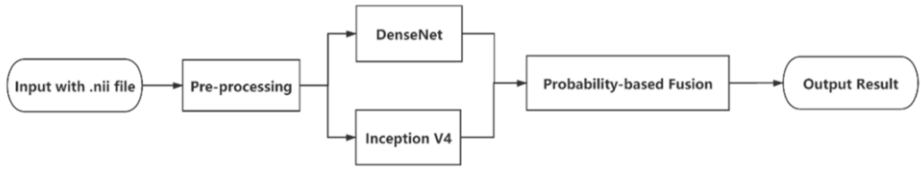

Figure 1. Illustration of the ensemble model

Shown in Fig. 1, at the very beginning, the 3D images are divided into a training set, a validation set, and a test set. Then the 3D images undergo pre-processing through FreeSurfer [12] and are sliced into 256 slices, among which the most informative 32 slices are selected for the following process. The slices will be processed by DenseNet and Inception V4 respectively. During the testing, the judgment given by each base classifier will be passed to the probability-based fusion to figure out the final judgment of the classification.

2.1. Pre-processing and Data Partitioning

In the pre-processing part, we employ a pre-processing model named P_FS_E, which is part of the model made by Jain R et al. [8]. In our method, following the 80-20 rule [13], we divide the data set in the ".nii" form into two parts: One is for the training process (80%) and the other is test set (20%). Within the data set for the training process, the data set is again divided into the training set (72%) and validation set (8%). Compared with the paper composed by Jain R et al. [8], their division is based on the slice, which means one person's 32 slices could appear on more than one set. This could be a trick to enhance accuracy, which fails to show the generalization capability of the model.

2.2. Classification Models

2.2.1. DenseNet

In the MRI detection, we select DenseNet instead of VGG16 or Inception V3 because in the experiment given in the next part indicates the higher accuracy and robustness of DenseNet. For DenseNet, the output part from the top layer was removed, and only input layers, dense blocks and transition layers are reserved. The last dense block is connected to the Global Average Pooling, after which the new output layer and fully-connected layers are added. Softmax activates the classification layer and ternary classification is accomplished.

Convolution	112 x 112	7 x 7 conv, stride 2
Pooling	56 x 56	3 x 3 max pool, stride 2
Dense Block (1)	56 x 56	$\begin{bmatrix} 1\ x\ 1\ conv \\ 3\ x\ 3\ conv \end{bmatrix}$ x 6
Transition Layer (1)	56 x 56	1 x 1 conv
	28 x 28	2 x 2 average pool, stride 2
Dense Block (2)	28 x 28	$\begin{bmatrix} 1\ x\ 1\ conv \\ 3\ x\ 3\ conv \end{bmatrix}$ x 12
Transition Layer (2)	28 x 28	1 x 1 conv
	14 x 14	2 x 2 average pool, stride 2
Dense Block (3)	14 x 14	$\begin{bmatrix} 1\ x\ 1\ conv \\ 3\ x\ 3\ conv \end{bmatrix}$ x 24
Transition Layer (3)	14 x 14	1 x 1 conv
	7 x 7	2 x 2 average pool, stride 2
Dense Block (4)	7 x 7	$\begin{bmatrix} 1\ x\ 1\ conv \\ 3\ x\ 3\ conv \end{bmatrix}$ x 16
Classification Layer	1 x 1	7 x 7 global average pool
		3D fully-connected, softmax

Figure 2. Structure of the DenseNet classification model

Softmax in Fig. 2. is a function that takes as input a vector of K real numbers, and normalizes it into a probability distribution consisting of K probabilities proportional to the exponentials of the input numbers. Each component after softmax will be in the interval (0,1), and the components will add up to 1, so that they can be interpreted as probabilities. Softmax is defined as:

$$\sigma(\boldsymbol{z})_i = \frac{e^{z_i}}{\sum_{j=0}^{k} e^{z_j}} \text{ for } i = 1 \ldots K \text{ and } \boldsymbol{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K \quad (1)$$

The performance of a classifier, the cross-entropy loss, is calculated as:

$$L(y, p) = -\sum_{c=1}^{M} y_{o,c} \log (p_{o,c}) \quad (2)$$

where y is the actual value, p is the predicted value, M is the class no.

Mini-Batch Gradient Descent is used as the optimization function. The learning rate is set as 0.01. The weight θ used for updating the neuron is calculated as:

$$\theta = \theta - \alpha \nabla_\theta J(\theta; x^{i:i+n}; y^{i:i+n}) \quad (3)$$

where α is the learning rate, ∇ is the gradient operand, J is loss function and x, y are sample labels.

Transfer learning is adapted for the size of the data set is so limited. In the model, we employ the DenseNet trained on ImageNet to enhance the classification capability of the model.

After training the model trained on the slice level, majority voting is introduced to accurately classify the case hierarchy. By selecting some specific slices on the sample,

the input model is used to predict the type of the slice, and the category in which the votes are selected by voting is the category of the sample stage.

2.2.2. Inception V4

The Inception architecture is a deep CNN invented by Google. Based on Inception V1, Inception V4 is an architecture brought up by [7]. The Inception V4 discards the fully-connected layer in preference of a global average pooling and connects with a softmax layer to reduce the large parameter number and overfitting [5].

Due to the limitation of the computational devices of our research group, we do not train the whole Inception V4 model by ourselves. Instead, we use the pre-trained model from the internet trained with the dataset from the ImageNet, which contains over 14 million images. The kernel of the pre-trained model can extract the features from general pictures very well, and we do not re-train the convolutional layers and pooling layers of it. Fixing the convolutional layers and pooling layers, we only re-train the last fully-connected layer. This approach can not only facilitate the computation but reserve the generalization capability of the middle convolutional and pooling layers as well.

2.2.3. A Probability-based Fusion

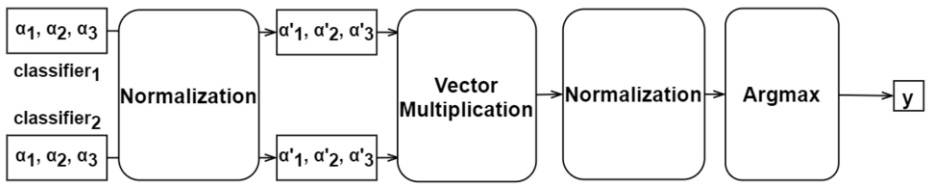

Figure 3. Structure of the probability-based fusion

Shown in Fig. 3., there are 2 classifiers in our model. The classifier_1 is DenseNet and classifier_2 is Inception V4. and α_1, α_2, α_3 are three probabilities of a prediction given by a classifier, which indicates the probabilities of AD, MCI and NC respectively when a sample is given. The probabilities have such a relationship, which is $\alpha_1 + \alpha_2 + \alpha_3 = 1$. Each prediction will first be normalized as below:

$$(\alpha_1', \alpha_2', \alpha_3') = \frac{(\alpha_1, \alpha_2, \alpha_3)}{max(\alpha_1, \alpha_2, \alpha_3)} \tag{4}$$

After normalization, the predictions of different classifiers will be multiplied to vote. Then, the multiplied vectors will be normalized and the argmax is the final prediction of class. The calculation process is shown below:

$$y = \arg max\,(\prod_{i=1}^{m} {\alpha_1}^i, \prod_{i=1}^{m} {\alpha_2}^i, \prod_{i=1}^{m} {\alpha_3}^i) \tag{5}$$

3. Experiment

3.1. Ternary Classification of the MRI Slices

We built the model with Keras [14], which is written in Python with TensorFlow [15].

380 medical cases (342 for the training set and 38 for the validation set) were employed for training the model, and 95 cases were used for the test set. Our model for ternary classification was conducted on 12160 slices in a batch size of 16 in 120 epochs. At each epoch, parameter values are updated.

Accuracy is calculated as:

$$Accuracy = \frac{\sum_{i=1}^{n} B^i}{n} \tag{6}$$

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

$$B^i = \begin{cases} 0 & y^i_{true} \neq y^i_{predicted} \\ 1 & y^i_{true} = y^i_{predicted} \end{cases} \tag{7}$$

For further performance evaluation, Precision, Recall, and F1-Score are also presented, which are calculated as:

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{10}$$

where TP is true positive predictions, FP is false positive predictions, and FN is false negative predictions.

3.1.1. Selection of DenseNet

We compared VGG16, DenseNet and Inception V3 on the classification. The confusion matrices are presented in Table 1.

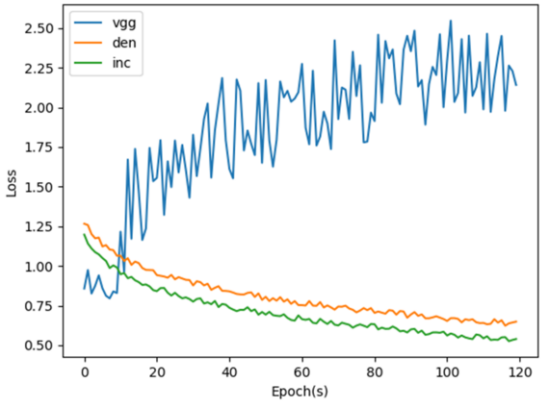

Figure 4. Validation loss of VGG16 (blue line), DenseNet (orange line) and Inception V3 (green line)

From the confusion matrices, we can see DenseNet is more accurate and robust than VGG16. Based on the experiments conducted by Hon and Khan [5], the average accuracy of the leveraged and fine-tuned VGG16 achieved is 0.92296 in the binary classification using 6400 MRI images. Moreover, Fig. 4 also illustrates the loss function of VGG16 does not converge. Therefore, the high accuracy (0.9706) fails to indicate the fabulous performance of VGG16. In contrast, DenseNet could be the most reliable and stable algorithm when it comes to the classification. The connection of the DenseNet guarantees that the overfitting will hardly occur. Shown in Table 2, the accuracy of DenseNet, 0.9118, is still delightful with high accuracy. DenseNet can extract as many features as possible without overfitting. That is why DenseNet is eventually selected as one of the classification models. Obviously, due to the low accuracy of Inception V3 in Table 6, we do not consider it as a selection.

Table 1. Confusion matrix of DenseNet, VGG16 and Inception V3

Actual Value - Prediction	DenseNet	VGG16	Inception V3
AD - AD	0.2059	0.1912	0.1471
AD – MCI	0.0147	0.0294	0.0294
AD - NC	0.0147	0	0.0147
MCI - AD	0.0147	0.0294	0
MCI - MCI	0.3824	0.2941	0.3235
MCI - NC	0.0294	0.0147	0.0735
NC - AD	0.0147	0.0588	0.0882
NC - MCI	0	0.1324	0.0441
NC - NC	0.3235	0.2500	0.2794

Table 2. Performance of DenseNet, VGG16 and Inception V3

Indices	DenseNet	VGG16	Inception V3
Precision of AD	0.8750	1.0000	0.7692
Precision of MCI	0.8966	0.9310	0.8148
Precision of NC	0.9565	1.0000	0.6786
Recall of AD	0.8750	1.0000	0.6250
Recall of MCI	0.9630	1.0000	0.8148
Recall of NC	0.8800	0.9200	0.7600
F1-score of AD	0.8750	1.0000	0.6897
F1-score of MCI	0.9286	0.9643	0.8148
F1-score of NC	0.9167	0.9583	0.7170
Accuracy	0.9118	0.9706	0.7500

3.1.2. Inception V4

We also employ Inception V4 as the enhancement of the classification. Shown in Table 3 and Table 4, the accuracy reaches 0.8235 and the confusion matrix indicates that Inception V4 is comparatively robust.

Table 3. Performance of Inception V4

	Precision	Recall	F1-score
AD	0.8000	0.7500	0.7742
MCI	0.8696	0.8000	0.8333
NC	0.8000	0.8889	0.8421
Accuracy		0.8235	

Table 4. Confusion matrix of Inception V4

Labels	AD	MCI	NC
AD	0.1765	0.0147	0.0294
MCI	0.0294	0.3529	0.0147
NC	0.0147	0.0588	0.2941

3.1.3. The Result after the Fusion

Shown in Table 5 and Table 6, during the fusion process, the performance of our diagnosis model is enhanced and the ensemble model achieves an accuracy of 0.9256.

Table 5. Performance after Fusion

	Precision	Recall	F1-score
AD	1.0000	0.8750	0.9333
MCI	0.9000	1.0000	0.9474
NC	0.9167	0.8800	0.8980
Accuracy		0.9256	

Table 6. Confusion matrix of Fusion

Labels	AD	MCI	NC
AD	0.2059	0	0.0294
MCI	0	0.3971	0
NC	0.0147	0.0441	0.3235

3.2. Comparison with Others' Research

We compared our work with some others' which employed ternary classification. The volume in Table 7 indicates the number of cases instead of the slices.

Table 7. Comparison with Others' Work

Methods	Source	Volume	Accuracy
Gupta A et al. [16]	ADNI	843	0.8500
Payan et al. [17]	ADNI	2265	0.8553
H.A. E et al. [18]	ADNI + CADDementia	210	0.8910
Jain R et al. [8]	ADNI	150	0.9573
This paper	ADNI	475	0.9265

It is worth mentioning again that although Jain R et al. [8], whose method shows high accuracy in Table 7, have already achieved a satisfying accomplishment, our approach to splitting the data set guarantees that nobody's MRI slices can exist in more than one set among the training set, validation set, and test set. Merely based on ADNI, it is delightful that our method reaches a maximum of 0.9265 with such a limited volume of data.

4. Conclusion

In this paper, we proposed an ensemble method that combines DenseNet (transfer learning) with pre-trained Inception V4. In the classification, we reached a maximum of 0.9265 of accuracy in ternary classification with each classification module surpassing 0.8 of accuracy. In our future work, we will continue striving for the enhancement of the probability-based fusion to achieve higher accuracy.

Acknowledgment

This work is supported by Ministry of Education-Microsoft Industry-University Cooperation Project for Educating Teaching Contents and Curriculum System Reform under Grant 201802011006, National Innovation and Entrepreneurship Training Program for College Students under Grant S201910561249, Innovation and Entrepreneurship Training Program of School of Software in South China University of Technology and Student Research Project of South China University of Technology.

References

[1] A Wilson R S, Segawa E, Boyle P A, et al., The natural history of cognitive decline in Alzheimer's disease[J], Psychology and aging (2012), 27(4): 1008.
[2] Iliffe S, Burns A, Alzheimer's disease, BMJ (2009), 338: b158
[3] Chan KY, Wang W, Wu JJ, et al., Epidemiology of Alzheimer's disease and other forms of dementia in China, 1990–2010: a systematic review and analysis, Lancet (2013), 381(9882): 2016–2023.
[4] Glozman T, Azhari H, A method for characterization of tissue elastic properties combining ultrasonic computed tomography with elastography, Journal of Ultrasound in Medicine (2010), 29(3): 387-398.
[5] Hon M, Khan N M, Towards Alzheimer's disease classification through transfer learning[C], 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) (2017), 1166-1169
[6] Simonyan K, Zisserman A, Very deep convolutional networks for large-scale image recognition, International Conference on Learning Representations (2015)
[7] Szegedy C, Ioffe S, Vanhoucke V, et al., Inception-v4, inception-resnet and the impact of residual connections on learning, Thirty-First AAAI Conference on Artificial Intelligence (2017)
[8] Jain R, Jain N, Aggarwal A, et al., Convolutional neural network-based Alzheimer's disease classification from magnetic resonance brain images, Cognitive Systems Research (2019)
[9] Fukushima, K., Neocognitron, Scholarpedia (2007), 2 (1): 1717.
[10] G. Wen, Z. Hou, H. Li, D. Li, L. Jiang, E. Xun, Ensemble of deep neural networks with probability-based fusion for facial expression recognition, Cognit. Comput. 9 (5) (2017) 597–610.
[11] Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4700-4708.
[12] Fischl B, FreeSurfer, Neuroimage (2012), 62(2): 774-781
[13] Newman, MEJ, Power laws, Pareto Distributions, and Zipf's law, Contemporary Physics (2005). 46 (5): 323–351.
[14] Chollet, F, et al., Keras(2015)
[15] Abadi M, Agarwal A, Barham P, et al. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow. org[J]. URL http: tensorflow. org, 2015
[16] Gupta A, Ayhan M, Maida A. Natural image bases to represent neuroimaging data[C]. International conference on machine learning. 2013: 987-994.
[17] Payan A, Montana G. Predicting Alzheimer's disease: a neuroimaging study with 3D convolutional neural networks[J]. arXiv preprint arXiv:1502.02506, 2015
[18] Hosseini-Asl E, Gimel'farb G, El-Baz A, Alzheimer's disease diagnostics by a deeply supervised adaptable 3D convolutional network, Computational and Mathematical Methods in Medicine (2016)