Siamese Network for Learning Robust Feature of Hippocampi

Samsuddin Ahmed*, Ho Yub Jung**

Abstract

Hippocampus is a complex brain structure embedded deep into the temporal lobe. Studies have shown that this structure gets affected by neurological and psychiatric disorders and it is a significant landmark for diagnosing neurodegenerative diseases. Hippocampus features play very significant roles in region-of-interest based analysis for disease diagnosis and prognosis. In this study, we have attempted to learn the embeddings of this important biomarker. As conventional metric learning methods for feature embedding is known to lacking in capturing semantic similarity among the data under study, we have trained deep Siamese convolutional neural network for learning metric of the hippocampus. We have exploited Gwangju Alzheimer's and Related Dementia cohort data set in our study. The input to the network was pairs of three-view patches (TVPs) of size 32 × 32× 3. The positive samples were taken from the vicinity of a specified landmark for the hippocampus and negative samples were taken from random locations of the brain excluding hippocampi regions. We have achieved 98.72% accuracy in verifying hippocampus TVPs.

Keywords: Hippocampus|Feature Representation|Siamese Network

I. INTRODUCTION

The Hippocampus, a structure of the brain's limbic system, is responsible to be playing key roles in memory and learning-process[1]. Extensive studies of this part revealed that atrophy of this region has clinical consequences as it is the earliest and most severely affected structure in terms of volume and shape by several neuropsychiatric disorders such as neurodegenerative diseases, epilepsy, etc.[2-4]. Hippocampus looks like sea-horses as its name suggests[5]. In the coronal section, the shape is like a peninsula of gray matter surrounded by white matter appearing both the hemispheres. We have depicted three different views (axial, sagittal, and coronal) of hippocampus in the figure I. This structure under study is known to be an important biomarker for Alzheimer's disease and other neurodegenerative diseases. So, this is of great importance to learn a metric for hippocampus embeddings which can be further utilized in medical image processing tasks starting from localization, registration, etc. to diseases prognosis and diagnosis.

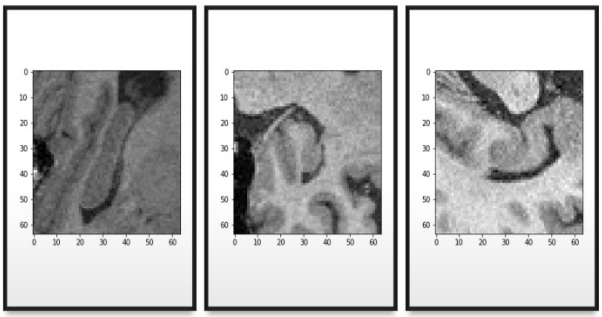

Fig.1. Hippocampus in sagittal, axial and coronal view (from left to right)

The task is to devise a model that transforms an

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

* PhD student, Department of Computer Engineering, Chosun University.
** Assistant Professor, Department of Computer Engineering, Chosun University.

Manuscript: 2020. 03. 11
Revised: 2020.06.09, 2020.06.12
Confirmation of Publication: 2020. 06.17
Corresponding Author: Ho Yub Jung
e-mail: hoyub@chosun.ac.kr

input pattern into an intermediary representation such that a preliminary metric in that intermediary representation space (ex. the Euclidean distance) approximates the semantic distance in the input space. In this study, a convolutional neural network (CNN) model was trained and tested on Gwangju Alzheimer's and related dementia (GARD) dataset. The Siamese architecture along with contrastive loss function was considered for learning the representation of the hippocampus.

We have organized the paper as follows: In section II, we have described preliminary concepts of a metric followed by the limitation of conventional metric learning and the reason for relying on deep based approaches. In section III, we briefly introduced the dataset. Section IV illustrates the structural details and design parameters of the Siamese network. Section V elaborately represents the experimental setups, data preparation processes, training criteria, and performances. Testing performance is presented in section VI. Section VII concludes the paper.

II. PRELIMINARY CONCEPTS

Suppose, we are given a dataset X. Two instances of the dataset are x_i and x_j. If we want to measure the similarity or dissimilarity, we need to measure the distance, d of these data points. To measure the distances, we use a distance metric. Any distance measure needs to satisfy the following four properties to be a metric [6].

(a) Nonnegativity:

$$d(x_i, x_j) \geq 0 \quad (1)$$

(b) Symmetry:

$$d(x_i, x_j) = d(x_j, x_i). \quad (2)$$

(c) Triangular inequality:

$$d(x_i, x_j) \leq d(x_i, x_k) + d(x_k, x_j) \quad (3)$$

(d) Identity of indiscernible:

$$d(x_i, x_j) = 0 \; for \; i = j \quad (4)$$

The commonly used distance metrics are variants of Chebyshev distance, cosine similarity, bilinear similarity, geodesic distance, etc. But these primitive metrics are sensitive to the scale and dimensions of the features. Furthermore, these cannot use contextual side information for similarity calculation [7,8]. As a result, most of the applications which use these metrics do not provide accurate results.

As an example, in figure 2, the conventional metrics are not capable of concluding that the semantically same objects are similar to each other as the semantically different objects are dissimilar. So, we need metric learning algorithms that will incorporate the internal properties of the data set as well as consider the user perspectives to find similarity and/or dissimilarity. Facing the limitation of these primitive metrics which do not consider the human perception of similarity/dissimilarity concepts, metric learning algorithms are developed.

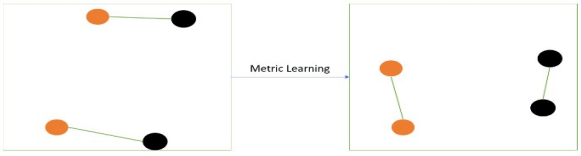

Fig.2. Illustrating the metric learning concept; the same color bubbles are semantically similar while different colors indicate that bubbles are semantically dissimilar. Metric Learning Algorithm bringing similar objects nearer while pushing the semantically different object away

The first metric learning algorithm, developed by Xing et all in 2002 [9], basically learns the Mahalanobis matrix. The distance is defined by:

$$d(x_i, x_j) = \sqrt{(x_i - x_j)^T M^{-1} (x_i - x_j)} \quad (5)$$

where M−1 is the Mahalanobis distance, which is a positive semi−definite matrix that satisfies the metric conditions. The M^{-1} parameterizes the distance. When M^{-1} is the identity matrix, the distance is equivalent to Euclidean distance. The Mahalanobis matrix M^{-1} scales the features and utilizes their correlations to compute distances between data more effectively [6].

The main task of conventional distance metric learning algorithms is to learn M^{-1} to minimize a constraint cost function. These methods are not powerful enough to capture the nonlinear relationship among data points [10]. Kernel trick can overcome the problem and are being widely used to implicitly transform the sample data points into a high dimensional feature subspace. Metric learning methods then obtain a metric in the projected feature subspace. Despite getting feasible solutions to some extent, these methods suffer from the scalability

problems as it is difficult to get the explicit nonlinear mapping functions.

Rather than conventional metric learning approaches, we prefer deep metric learning (DML) for a couple of reasons. Firstly, DML can learn similarity measures without an explicit description of features[7]. Secondly, DML methods do not require data to be heavily pre-processed. Thirdly, it is very easy to implement and deploy a deep-net framework for a wide area of applications ranging from face verification to diseases prediction[11-14]. Fourthly, zero-shot, and one-shot learning require very small or no dataset for training the network[15-16]. Finally, most of the machine vision problems solved by deep neural networks are showing state-of-the-art performance [17]. For example, Face Net[18,19], Deep Face[20], etc. Machine vision community concentrating on deep distance metric learning for the last few years[21,22], and a lot of methods have been devised.

However, DML algorithms are good at addressing the nonlinearity and scalability problems which are the main limitations suffered by conventional metric learning algorithms. The common mechanism of deep based algorithms is to train a CNN for producing characteristic description i,e, a higher level of abstraction for each input vector so that a loss function related to object distance is minimized. Several state-of-the-art deep networks are being used for metric learning. In this study, the Siamese network [23] has been used for learning the desired metric.

III. DATASET

We have exploited Gwangju Alzheimer's and Related Dementia (GARD) cohort data set in this study[25-27]. The imaging was performed at Chosun University Hospital. These contiguous 0.9 mm MPRAGE images of the whole brain were acquired using a 1.5T magnetic resonance scanner (Magnetom Avanto, Siemens) with the following parameters: relaxation time (TR) = 1800 ms; echo time(TE) = 3.43 ms; TI = 1100 ms; flip angle: 15; field of view = 224x224; matrix = 320x320; number of slices = 178.

There are 326 sMRI in the dataset among which 20 sMRI were randomly selected for training, and the rest was used for testing. The intensities of each sMRI voxels were normalized so that the mean is zero and variance is one. After normalizing the intensities, we have generated 16 positive three-view patches (TVPs) and 16 negative TVPs from each sMRI. From the training TVPs, we have randomly selected 32 of them (16 positives and 16 negatives) for creating a database to be used for evaluating the model with the test set.

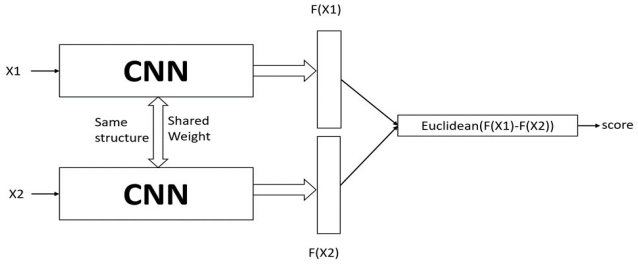

Fig.3. Siamese network for metric learning

IV. METHODOLOGY

In this study, we have deployed the Siamese network as depicted in figure 3 to learn the deep metric which can differentiate TVPs of hippocampus from non-hippocampi TVPs. Our CNN consists of a pair of networks sharing the same weights and loss function. Siamese network learned a function that maps input TVPs into a target space such that the Euclidean distance in the target space approximates the semantic distance between the TVPs. The learning process minimizes contrastive [14] loss function which ensures that the similarity metric is small for a pair of hippocampus-TVPs and large for distinct-region TVPs. The CNN works as the mapping function from input to target space. In each channel of the twin, (as depicted in figure 4) there are four convolution layers and one fully connected layer. There is a batch normalization layer after each convolution layer. The last layer of the twin-CNN is the Euclidean distance between the feature embedding of the two different networks.

Fig.4. One channel in the twin of the Siamese network for metric learning

The input to the network is a pair of TVPs (x_i,x_j) and a label y_{ij}. (x_i,x_j) are passed to the CNNs and each CNN work as a mapping function. The pair of CNNs yield representation $F(x_i)$ and $F(x_j)$ for TVP xi and TVP xj respectively. The cost module which is the Euclidean distance operator generates the distance $\hat{y}$ between $F(x_i)$ and $F(x_j)$.

We have used a contrastive loss function for training DML network. The loss function is defined in equation 6.

$$loss(y,\hat{y}) = y\hat{y}^2 + (1-y)[\max(\lambda - \hat{y}, 0)]^2 \quad (6)$$

Here, y is the actual distance (0 or 1) and $\hat{y}$ is predicted distance between the input pairs. λ=2 is used as a distance margin constraint. The constraint defines a radius in target space around Euclidean distance. Unlikely pairs have a contribution to the loss if their distance is within the defined margin.

V. EXPERIMENTAL SETUP

5.1. Platform

The experiment was performed with a python 3.6 environment. We used the TensorFlow GPU 1.8, keeping Keras as the backend. The operating system was Windows 10 installed on an "Intel(R) Xeon (R) Silver 4114 @ 2.20 GHz, 10 cores and 20 logical processors with a 32 GB RAM" machine. The GPU was NVIDIA Quadro P4000. Multi-image analysis GUI (Mango) was used for viewing and navigating throw the neuroimaging informatics technology initiative (NIFTI) images.

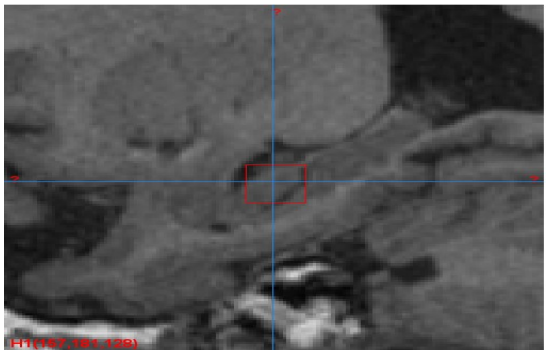

Fig.5. Landmark used for hippocampus TVP generation

5.2. Dataset Preparation

The data set preparation process is summarized in algorithm I. At first, we have manually marked the scans for hippocampus landmarks. The sagittal view of the considered landmark position of the left hippocampus is illustrated in figure 5. Then, two different sets of TVPs are generated. Step 3 to step 5 of algorithm 1, generates the hippocampal TVPs (i,e. positive sample) while step 6 to step 8 generates non-hippocampi TVPs (i,e. negative sample). The positive samples were randomly produced from 4×4×4 cube centering at a manually labeled hippocampi location. The negative samples are produced from other regions of the brain. We have generated 320 TVPs for each set.

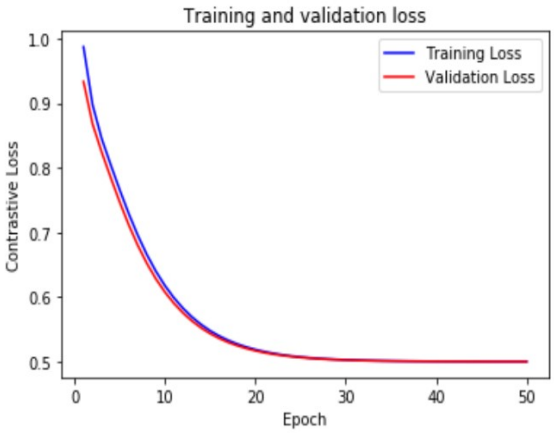

Fig.6. Training and validation loss of deep metric learning network

The pair construction process is described in step 9 to step 13 of algorithm I. The pair selection steps ensure keeping an equal number of similar pairs and dissimilar pairs for both training and testing. Input data, consist of a pair of 32×32×3 TVPs, taken from positive and negative samples. A sample for training

can be denoted as ($[x_i, x_j]$, y_{ij}), where x_i, x_j are TVPs and y_{ij} is the label. y_{ij}=1 if x_i is from the same regions as x_j, 0 (zero) otherwise. To avoid bias in training, we randomly shuffled the data at the end of the preparation process.

Algorithm I: Data Preparation Algorithm

Input: I: A sequence of preprocessed structured magnetic resonance images; H: a sequence of hippocampus locations for the images in I

Output: D: A data set consisting of pairs ($[x_i, x_j]$, y_{ij}) where x_i and x_j are two TVPs and y_{ij} is the label

1 *H= {set of hippocampus locations for i ε I}*
2 *Repeat step 2 to 14 for each mri ε I*
3 *R_h= {set of random voxel points from the reference cube formed centering locations in H}*
4 *Repeat step 5 for each point (x,y,z) ε R_h*
5 *axial= mri[x, y−16:y+16, z−16:z+16];*
sagittal= [x−16:x+16, y, z−16:z+16];
coronal= [x−16:x+16, y−16:y+16, z];
TVP= [axial sagittal coronal]
X_1=append (X_1, TVP)
6 *R_{nh}= {set of random voxel point from the mri excluding hippocampus locations}*
7 *Repeat step 8 for each point (x, y, z) ε R_{nh}*
8 *axial= mri[x, y−16:y+16, z−16:z+16];*
sagittal= [x−16:x+16, y, z−16:z+16];
coronal= [x−16:x+16, y−16:y+16, z]
TVP= [axial sagittal coronal]
X_2=append (X_2, TVP)
9 *Repeat step 10 to 13 for each tvp_i ε X_1*
10 *Repeat step 11 for each tvp_j ε X_1*
11 *sample=([tvp_i, tvp_j], 1)*
D.append(sample)
12 *Repeat step 13 for each tvp_j ε X_2*
13 *sample=(tvp_i, tvp_j,0)*
D.append(sample)
14 *shuffle(D)*
15 *return D*

For testing the trained model, we have generated 16 random TVPs for each class i, e, positive and negative class from the training set to form the database. The TVPs in the database is compared with the test TVPs for finding the dissimilarity scores. We have taken the minimum distance score among all the scores for all TVPs in the database and the label of the minimum scored database TVP was considered as the label of test TVP.

5.3. Training

We have used a contrastive loss function (equation 6) for training DML network. The distance margin constraint in the contrastive loss was kept 2. The constraint defines a radius in target space around Euclidean distance. Unlikely pairs have a contribution to the loss if their distance is within the defined margin.

We have initialized the weight by a normal distribution with zero mean and a standard deviation of 0.01. The biases were initialized for this network from a normal distribution with different mean (0.5) and the same standard deviation (0.01). For the fully connected layers, we have initialized biases differently i, e. the mean of the normal distribution was kept zero with standard deviation 0.2.

The Adam optimizer is used with a mini−batch size of 32 and an initial learning rate of 0.001. The decay of the learning rate was kept uniform (0.1) if there is no update in the loss for consecutive three epoch. The model has used a grid search to perform the hyper−parameter selection.

We have trained the presented models for 150 epochs with a batch size of 32. 10−fold cross−validation on the training TVPs was performed. The training performance of the metric learning network is depicted in figure 5.

The reason for the better validation performance than the training is that at validation time the dropout layer and regularizers in different layers are turned off. Another reason is that training loss is calculated as an average of all batch−wise losses in each epoch. On the other hand, the validation loss is calculated at the end of each epoch. So, in our case, the validation loss is lower than the training loss.

VI. RESULT

6.1 Evaluation Criteria

For interpreting the score, we consider the multiplicative inverse of the Euclidean distances yielded by the model. The additive factor 1 (one) prevents divide by zero error. If the model output is y for any pair (x_i, x_j), we have transformed y to Y according to equation 7.

$$Y = \frac{1}{1+y} \quad (7)$$

Here, y is the un-normalized Euclidean distance of (i.e., the dissimilarity between) two TVPs in the target space and Y is the normalized similarity score in the open interval [0,1]. This transformation makes sure the range of similarity is between the interval [0,1] while do not alter the inverse relation between similarity and dissimilarity.

For testing the DML network, we consider the accuracy in verifying whether the TVPs are containing the hippocampus or not. If the minimum score found with the positive class database, we consider the test TVP was taken from the hippocampus region, and if the minimum score is found for the negative class database then the TVP under observation is considered non-hippocampi.

The accuracy of the model is calculated based on equation 8.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

Here, TP is the number of TVPs that are drawn from hippocampus regions and matched with the positive class database, TN is the number of TVPs that are drawn from the non-hippocampus regions and matched with the negative class database, FP is the number of TVPs drawn from the non-hippocampi region but matched with the positive class database, FN is the number of TVPs drawn from hippocampus regions but matched with the negative class database.

6.2 Result Analysis

Table I presents class label verification results along with a confidence score of eight TVPs from different sMRI scans from the GARD data set. The TVPs generated from hippocampus voxels were classified as hippocampus TVPs. And the TVPs generated from other locations were classified as non-hippocampus TVPs. The total accuracy we have achieved is 98.72% in finding similar TVPs. We have provided a normalized similarity score for 8 test TVPs from eight different test sMRI. MRI ID is the identification number of each scan. Voxel positions indicate the reference locations for generating TVP. Actual and verified regions define the regions from where the sample was generated and the classification result for model respectively. D is the normalized similarity score of the sample compared with the stored database. The distance scores are normalized by equation 7 to get the similarity score.

Table 1. Performance evaluation of the metric learning method on GARD data. D is similarity scores (normalized using equation 7) for eight different TVPs of different sMRI from the GARD dataset. The similarity indicates that the presented TVP is similar to the related class in the stored database with given a confidence score

MRI ID	Voxel Position	Actual Region?	Verified Region?	D
14071906	77,127,82	Y	Y	0.97
14051804	92,157,84	Y	Y	0.89
14080210	66,159,95	N	N	0.96
17101603	82,141,80	Y	Y	0.86
14051110	122,76,145	N	N	0.93
17092001	145,88,133	N	N	0.87
15031904	124,72,153	N	N	0.91
15031904	85,152,90	Y	Y	0.82

In our study, we have avoided 3D image computation by considering TVPs. This helps to generate training data so that overfitting can be avoided. We have trained and tested the Siamese CNN model (figure 4) for robust feature learning. The findings in this study might be useful in localization of cerebral landmarks and other areas of diagnosis and prognosis of diseases from sMRI.

VII. CONCLUSION

In this work, we have learned hippocampus features using deep CNN. Rather than using 3D information, we have generated TVPs from the vicinity of a specific landmark of the hippocampus. The model was trained and tested based on the TVPs. The proposed Siamese network architecture provides robust accuracy in learning hippocampus features. We have observed 98.72% accuracy in verifying hippocampus TVPs by the proposed model. This achievement has further application in sMRI processing and analysis.

REFERENCES

[1] M.-R. Siadat, H. Soltanian Zadeh, and K. V Elisevich, "Knowledge-based localization of hippocampus in human brain MRI," *Comput. Biol. Med.*, vol. 37, no. 9, pp. 1342-1360, 2007

[2] K. S. Anand and V. Dhikav, "Hippocampus in health and disease: An overview," *Ann. Indian Acad. Neurol.*, vol. 15, no. 4, pp. 239-246, 2012

[3] A. M. Kälin *et al.*, "Subcortical Shape Changes, Hippocampal Atrophy and Cortical Thinning in Future Alzheimers Disease Patients," *Front. Aging Neurosci.*, vol. 9, no. 38, Jul. 2017

[4] S. Ahmed *et al.*, "Ensembles of Patch-Based Classifiers for Diagnosis of Alzheimer Diseases," *IEEE Access*, vol. 7, pp. 73373-73383, 2019

[5] J. Milos and P. Mihovil, "A note on the sea-horse in the human brain," *Transl. Neurosci.*, vol. 1, no. 4, pp. 335-337, Dec. 2010

[6] J. Han and M. Kamber, *Data mining: concepts and techniques*, 3rd ed. Morgan Kaufmann Publishers Inc., 2012

[7] Kaya and Bilge, "Deep Metric Learning: A Survey," *Symmetry (Basel).*, vol. 11, no. 9, pp. 1066, 2019

[8] S. Ahmed, A. Basher, A. Reja, and H. Y. Jung, "A brief Review on Deep Metric Learning," *Korea Next Generation Computing Society Spring Conference 2018*, 2018

[9] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance Metric Learning, with Application to Clustering with Side-information," *Proceedings of the 15th International Conference on Neural Information Processing Systems*, pp. 521-528, 2002

[10] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27-48, 2016

[11] N. Torosdagli, D. K. Liberton, P. Verma, M. Sincan, J. S. Lee, and U. Bagci, "Deep Geodesic Learning for Segmentation and Anatomical Landmarking," *IEEE Trans. Med. Imaging*, vol. 38, no. 4, pp. 919-931, 2019

[12] D. Chitradevi and S. Prabha, "Analysis of brain sub regions using optimization techniques and deep learning method in Alzheimer disease," *Appl. Soft Comput. J.*, vol. 86, pp. 105857, 2020

[13] D. Lu, K. Popuri, G. W. Ding, and R. Balachandar, "Multimodal and Multiscale Deep Neural Networks for the Early Diagnosis of Alzheimer's Disease using structural MR and FDG-PET images," *Sci. Rep.*, vol. 8, no. October 2017, pp. 1-13, 2018

[14] M. Wang and W. Deng, "Deep Face Recognition: A Survey," *CoRR*, vol. abs/1804.0, pp. 1-26, 2018

[15] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele, "Latent Embeddings for Zero-Shot Classification," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 69-77, 2016

[16] R. Donner and H. Bischof, "One-shot learning of anatomical structure localization models," *10th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 222-225, 2013

[17] P. Moutafis, M. Leng, and I. A. Kakadiaris, "An Overview and Empirical Comparison of Distance Metric Learning Methods," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 612-625, 2017

[18] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, pp. 815-823, 2015

[19] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," *32nd IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, no. 1, 2019

[20] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “DeepFace: Closing the Gap to Human-Level Performance in Face Verification,” *27th IEEE Conference on Computer Vision and Pattern Recognition, (CVPR) 2014*, pp. 1701-1708, 2014

[21] W. Yin, H. Schütze, B. Xiang, and B. Zhou, “ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs,” *Trans. Assoc. Comput. Linguist.*, vol. 4, pp. 259-272, 2016

[22] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, “Signature Verification Using a Siamese Time Delay Neural Network,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 7, no. 4, pp. 669-688, 1993

[23] S. Chopra, R. Hadsell, and Y. LeCun, “Learning a Similarity Metric Discriminatively, with Application to Face Verification,” *28th Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, pp. 539-546, 2005

[24] N. T. Duc *et al.*, “3D-Deep Learning Based Automatic Diagnosis of Alzheimer’s Disease with Joint MMSE Prediction Using Resting-State fMRI,” *Neuroinformatics*, vol. 18, no. 1, pp. 71-86, 2019

[25] M. Naveed, I. Qureshi, S. Ryu, J. Song, and J. H. Cole, “Evaluation of Functional Decline in Alzheimer’s Dementia Using 3D Deep Learning and Group ICA for rs-fMRI Measurements,” *Front. Aging Neurosci.*, vol. 11, no. February, pp. 1-9, 2019

[26] K. Y. Choi *et al.*, “APOE Promoter Polymorphism-219T / G is an E ff ect Modifier of the Influence of APOE ε 4 on Alzheimer ’ s Disease Risk in a Multiracial Sample,” 2019

[27] K. Y. Choi *et al.*, “APOE Promoter Polymorphism-219T / G is an Effect Modifier of the Influence of APOE ε 4 on Alzheimer ’ s Disease Risk in a Multiracial Sample,” vol. 8, no. 8, pp. 1-12, 2019

[28] Abol Basher, Samsuddin Ahmed, Ho Yub Jung, "One Step Measurements of hippocampal Pure Volumes from MRI Data Using an Ensemble Model of 3-D Convolutional Neural Network ," *Smart Media Journal*, vol. 9, no. 2, pp. 22-32, 2020

[29] Tien Duong Vu, Hyung-Jeong Yang , Luu Ngoc Do, Thao Nguyen Thieu, "Classifying Instantaneous Cognitive States from fMRI using Discriminant based Feature Selection and Adaboost," *Smart Media Journal*, vol. 5, no. 1, pp. 30-37, 2016

[30] Seo jeong Kim, Jae Su Lee, Hyong Suk Kim, "Deep learning-based Automatic Weed Detection on Onion Field," *Smart Media Journal*, vol. 7, no. 3, pp. 16-21, 2018

Authors

Samsuddin Ahmed

He received a bachelor's degree in computer science and engineering from the University of Chittagong, Bangladesh in 2010. In 2020, he received his master's degree in computer engineering from Chosun University, South Korea. Since 2010, he served as faculty member of computer science and engineering at different universities in Bangladesh. His research interests include artificial intelligence, computer vision and image processing, medical imaging, machine learning, etc.

E-mail: sambd86@gmail.com

Ho Yub Jung

He received the B.S. degree in electrical engineering from The University of Texas at Austin, in 2002, and the M.S. and Ph.D. degrees in electrical engineering and computer science from Seoul National University, in 2006 and 2012, respectively. He was with Samsung Electronics for two years as a Senior Engineer. Since 2017, he has been an Assistant Professor with the Department of Computer Engineering, Chosun University. His research interests include computer vision, machine learning, and medical imaging.

E-mail: hoyub@chosun.ac.kr