Journal of Multimedia Information System VOL. 9, NO. 3, September 2022 (pp. 177-182): ISSN 2383-7632 (Online)
https://doi.org/10.33851/JMIS.2022.9.3.177

Analysis of Weights and Feature Patterns in Popular 2D Deep Neural Networks Models for MRI Image Classification

Bijen Khagi[1], Goo-Rak Kwon[1*]

Abstract

A deep neural network (DNN) includes variables whose values keep on changing with the training process until it reaches the final point of convergence. These variables are the co-efficient of a polynomial expression to relate to the feature extraction process. In general, DNNs work in multiple 'dimensions' depending upon the number of channels and batches accounted for training. However, after the execution of feature extraction and before entering the SoftMax or other classifier, there is a conversion of features from multiple N-dimensions to a single vector form, where 'N' represents the number of activation channels. This usually happens in a Fully connected layer (FCL) or a dense layer. This reduced 2D feature is the subject of study for our analysis. For this, we have used the FCL, so the trained weights of this FCL will be used for the weight-class correlation analysis. The popular DNN models selected for our study are ResNet-101, VGG-19, and GoogleNet. These models' weights are directly used for fine-tuning (with all trained weights initially transferred) and scratch trained (with no weights transferred). Then the comparison is done by plotting the graph of feature distribution and the final FCL weights.

Key Words: Deep Neural Network, Activation Channels, MRI, Fully Connected Layer, Weights Correlation.

I. INTRODUCTION

Learning by supervision means the input is being considered to map some output so that certain characteristics (here class characteristics) are identified/stored in the model. The stored identifying capability can be used for other application (not the one in scratch learning) tasks, which we usually call transfer of knowledge or transfer learning [1-2]. Transfer learning has been a 'de facto' savior for most of the state-of-the-art deep learning applications [3-5]. The pretrained weights in the Convolutional Neural Network (CNN) models are finely tuned for the model to be applied in other processes. The major process for this is done by removing the final fully connected layer of 1000 output (for IMAGENET trained models like AlexNet [6], ResNet [7], VGG [8], GoogleNet [9]) into k numbers of output, where k represents the number of labels to be trained in the new task. This implication of the pretrained model into another application not only requires the transfer of weights but also requires the transfer of connection i.e., the whole model trained model is not transferred but just readjusted in its tail part for the new application. On the contrary, with scratch training, we need lots of supervising ground truth and at the same time since the whole network needs to learn from the training material (no external source for learning) it all starts from a 'zero-level'. And to reach from zero-level to an acceptable 'fitness', we need a lot of time, and material and still the acceptable fitness may not be as good as from one already trained model [10].

Krizhevsky et al. [6] successfully utilized CNN in natural image classification (ImageNet Database of 1,000 image types of class) with a minimum error rate in 2012. Later various variants of CNN were proposed by different researchers for image classification and object recognition tasks; the famous ones being Resnet, GoogleNet, and R-CNN [11], etc. Tajbaksh et al. [1] tested CNN in medical images for poly detection and Pulmonary embolism detection, where they highlighted pretrained or fined-tuned CNN performed well as scratch-trained CNN and suggested layer-wise tuning for practical performance. Similarly, Hoo-Chang Shin et al. [12] tested CNN architecture for Lymph-Node detection and Interstitial Lung disease classification, where they also tested a pretrained CNN network (AlexNet, GoogLeNet and CifarNet) and also used the transfer learning technique.

GoogLeNet (also called Inception V1) achieved a top-5 error rate of 6.67% in the ILSVRC competition for the ImageNet classification challenge in 2014 which was very close to the human-level performance. It's an architecture

Manuscript received August 31, 2022; Revised September 23, 2022; Accepted September 23, 2022 (ID No. JMIS-22M-08-029)

Corresponding Author (*): Goo-Rak Kwon, +82-62-230-7707, grkwon@chosun.ac.kr

[1]Dept. of Information and Communication Engineering, Chosun University, Gwangju, Korea, grkwon@chosun.ac.kr, bijenkhagi@gmail.com

developed by Szegedy et al. [9] at Google Inc. It has 22 layers and adopts multiple parallel convolution layer concatenation which is called the Inception module. The network used a CNN inspired by LeNet but implemented a novel element which is dubbed an inception module. It uses batch normalization, image distortions, and RMSprop. This module is based on several very small convolutions in order to drastically reduce the number of parameters. Their architecture consisted of a 22-layer deep CNN but reduced the number of parameters from 60 million (AlexNet) to 4 million. The key point is that the architecture uses a 1×1 convolution for the ensemble of features. The runner-up at the ILSVRC 2014 competition is VGGNet developed by Simonyan and Zisserman from Oxford University [8]. Out of the six VGG models, VGG16 and VGG19 are frequently used. VGGNet consists of 16 convolutional layers and is very interesting because of its uniform architecture using all 3x3 convolutional filters with stride size 1. The weight configuration of the VGGNet is publicly available and has been used in many other applications and challenges as a baseline feature extractor. However, it consists of 138 million parameters, which can be a bit challenging to handle. This structure is notable for its very simple methodology and has performed well. At last, at the ILSVRC 2015, the so-called Residual Neural Network (ResNet) by He et al. [7] presented a novel architecture with "skip connections" and substantial batch normalization. Such skip connections are also known as gated units or gated recurrent units and have a strong similarity to recent successful elements applied in RNNs. These skip connections also work as a residual connection to preserve the image features by working as an identity function. It achieved a top-5 error rate of 3.57%.

In this research, we are going to reinvestigate the transfer learning process mainly through the weights and feature analysis in the FCL layer. Here the reason for FCL selection is mainly because, of its simplicity and importance for the final decision. Additionally, this is the layer where all the channels/dimension rearranges into two-dimensional feature values [13]. In Section I we discuss the background of research, architecture details of CNN models, and some related work. In section II we discuss the methodology and the user data along with some training procedures. We present our result in section III and provide concluding remarks in section IV.

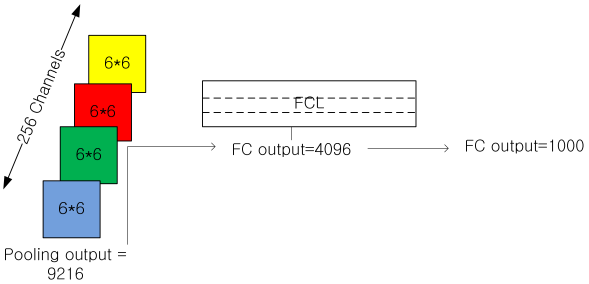


Fig. 1. Showing the problem of high dimensional pooling output being flattened/condensed when used as input for a fully connected layer. Here we are trying to show, that the spatial information of each channel or activated output of the channel might get lost when flattened in the FCL layer (from AlexNet architecture).

II. METHODOLOGY AND TRANSFER LEARNING REQUIREMENT

We have used the OASIS dataset brain MRI scans for the experiment. The available MRI scans are in 3D format since the pre-trained models available are only 2D architecture, hence we need to use the 2D images as inputs, for which each MRI scan was converted from analyze format to jpeg format using MRIcon software. Around 30 mid slices were extracted from each MRI scan so in total it produced 840 images each for NC and AD classes. These images were later split randomly in a 5:2:3 ratio for training, validation, and testing as shown in Table 1. All the used MRI scans are made publicly available to download at https://github.com/xen888/Dataset. The reported accuracy is for the 30% test set images. Table 2 shows out of all 3 models, one with freezed weights from the pre-trained model has the lowest accuracy, whereas the accuracy is highest with either fine-tuned or scratch trained. With fine-tuned we can save time, but might not get the best result, with scratch training, we need to train for higher epochs and might get the best result. However, with scratch training, we always have a chance of overfitting the model.

Fig. 2 shows the feature plot of output FCL values (each with 2 scores one for AD and the other for MCI), being plotted in its class label. This plot is not the weights, but the generated value as output from the model. Here each colored dot represents the feature property of an induvial class i.e., blue color for AD MRI and red color for NC MRI.

III. RESULT

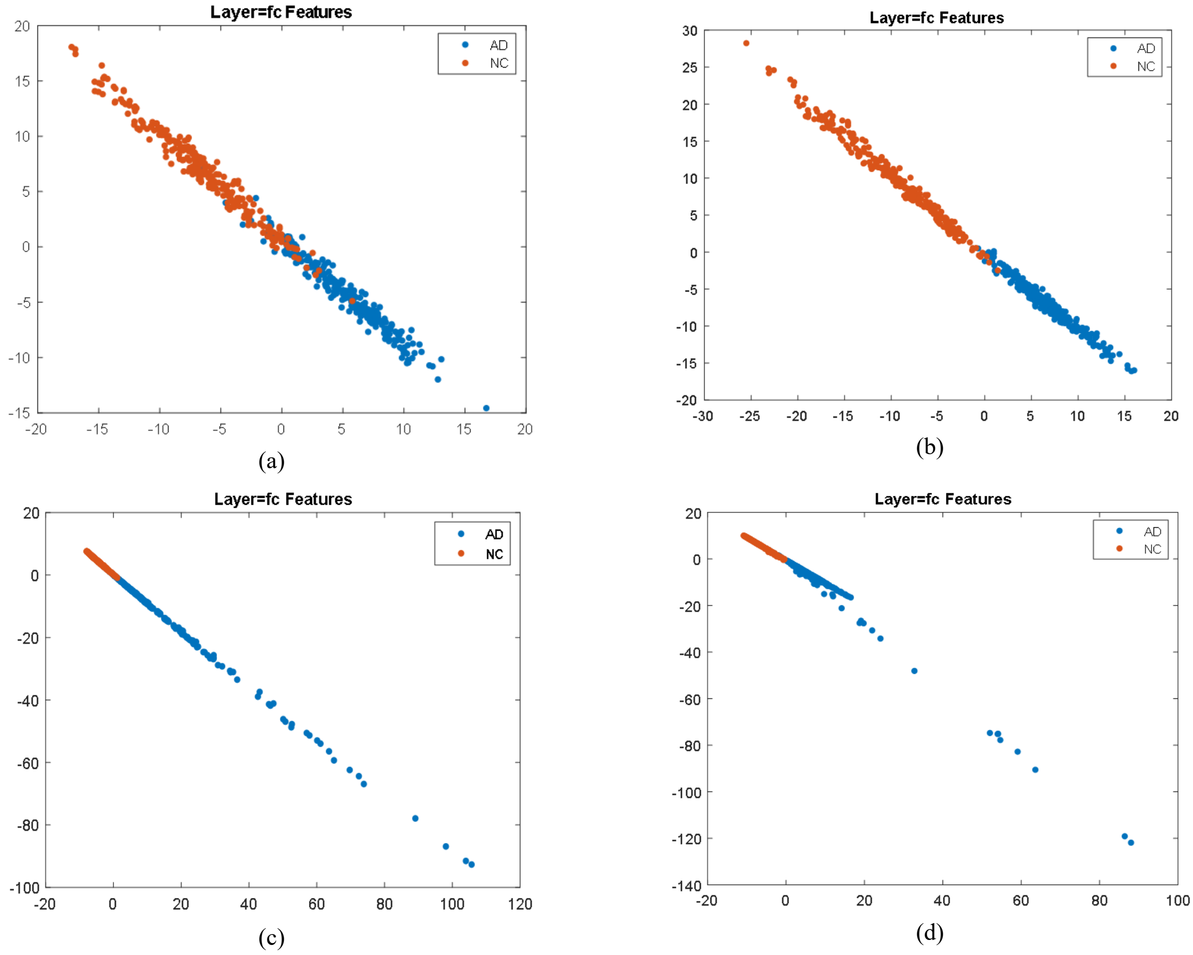
Simply, the training and testing result shows that a fine-tuned model works better than a freezed model. However, if we spend more training time, the training will improve the test accuracy at the cost of a higher training epoch.

Table 1. MRI scans used in experiments.

Class	Total number of scans	Number of scans from single MRI	Total training scans	Total testing scans
AD	840	30	420	252
NC	840	30	420	252

Table 2. The result of freezed weights, fine-tuned weights, and scratch-trained weights from Resnet-101, GoogleNet, and VGG-19 models. These results are analyzed specifically for their patterns as in Fig. 2. ‘l’ beneath each DNN model name denotes the initial learning rate of the model, which drops by 10% after every 10 epochs.

DNN model	Fine or scratch	Final training accuracy	Final validation accuracy (%)	Test accuracy (%)	Training time (min:sec)
ResNet-101 (l=0.001)	Freezed weight of ImageNet (20 epoch)	100	92.6	93.2	4:04
	Fine-tuned (20 epoch)	100	97.2	98.1	7:48
	Scratch-weightless (20 epoch)	100	95.8	96.5	8:29
	Scratch-weightless (50 epoch)	100	97.8	98.2	22:48
GoogleNet (l=0.001)	Freezed weight of ImageNet (30 epoch)	82	76.5	78.6	2:12
	Fine-tuned (30 epoch)	100	96.7	97.2	3:21
	Scratch-weightless (30 epoch)	99	96.4	96.8	3:10
	Scratch-weightless (60 epoch)	100	97.6	97.6	6:27
VGG-19 (l=0.0001)	Freezed weight of ImageNet (30 epoch)	80	79.46	76.59	2:24
	Fine-tuned (30 epoch)	100	98.21	99.12	3:55
	Scratch-weightless (30 epoch)	88	83.63	88.69	3:52
	Scratch-weightless (60 epoch)	99	96.43	97.82	7:46



(a)

(b)

(c)

(d)

Fig. 2. The demonstration of features yielded from variously trained models, (a) Freezed weights, (b) fine-tuned weights, (c) scratch-trained weights for 20 epochs, (d) scratch-trained weights for 50 epochs. Here each feature is obtained from the final FCL, and since all four models have very good test classification accuracy, the distribution of the class-wise feature (represented as colored dots) is highly discriminant. The x-axis and y-axis represent the weights value of the FCL layer for classes AD and NC respectively. Here, each color dots represent a single patient so the error might have resulted from the areas of overlapping between blue (AD) and red dots (NC).

Table 2 shows the final result of 2D MRI images classification using all three DNN models and trained under three different conditions as below:

a. Freezed: Here the whole network uses the final weights of the pre-trained models, trained on the IMAGENET dataset and as it is available/stored. These weights of all layers are not changed at all during training. A fully connected layer with 2 outputs is replaced with the original final FCL with 1,000 outputs, the input number being the same.
b. Fine-tuned: Here the models have the original weights of pre-trained models as a freezed model. However, during training, these weights are slightly updated using Stochastic gradient descent optimization during backpropagation. Eventually, the weights are slightly tuned for our MRI classification task.
c. Scratch-trained: Here we use the layers of the pre-trained model, but the weights are not transferred at all. It means the model has completely no weights (or say zero weights) before training. Once training starts depending upon the initialization algorithms the weight of each layer gets the value and updates via SGD optimization. Since the layer is weightless at the beginning, its value needs to be learned properly with input values during training hence called scratch training. Here we have used two versions of scratch training, one with a lower epoch e.g., scratch_20 denotes scratch training with only 20 epochs it is done to compare the value of the weight with its freezed and finetuned version. Other is one with a higher epoch to reach full convergence i.e., 100% training accuracy.

The classification performance is shown in Table 2. Here, it is interesting to note that the feature is sparsely dispersion in the case of freezed model and starts to be densely populated in scratch trained model. This might suggest that weights try to converge to a smaller range during scratch training or fine-tuning which is supported by the fact, that the accuracy of freezed model is comparatively lower than other fine-tuned and scratch-trained models (see Table 2). It means when sparsely dispersed the features are difficult to be classified. Also note the difference in the number of parameters i.e., weight values, e.g., in VGG-19 has 4,096×2, here it means 4,096 weights supporting for AD (1st dimension or x-axis) features and other 4,096 weights supporting for NC (2nd dimension or y-axis) features. Here each dimension in the x-y axis corresponds to each class, i.e., the x vs. y plot can be considered as the AD vs. NC plot because the first row of FCL is responsible for making decisions for the AD class and 2nd row is responsible for making decisions for the NC class.

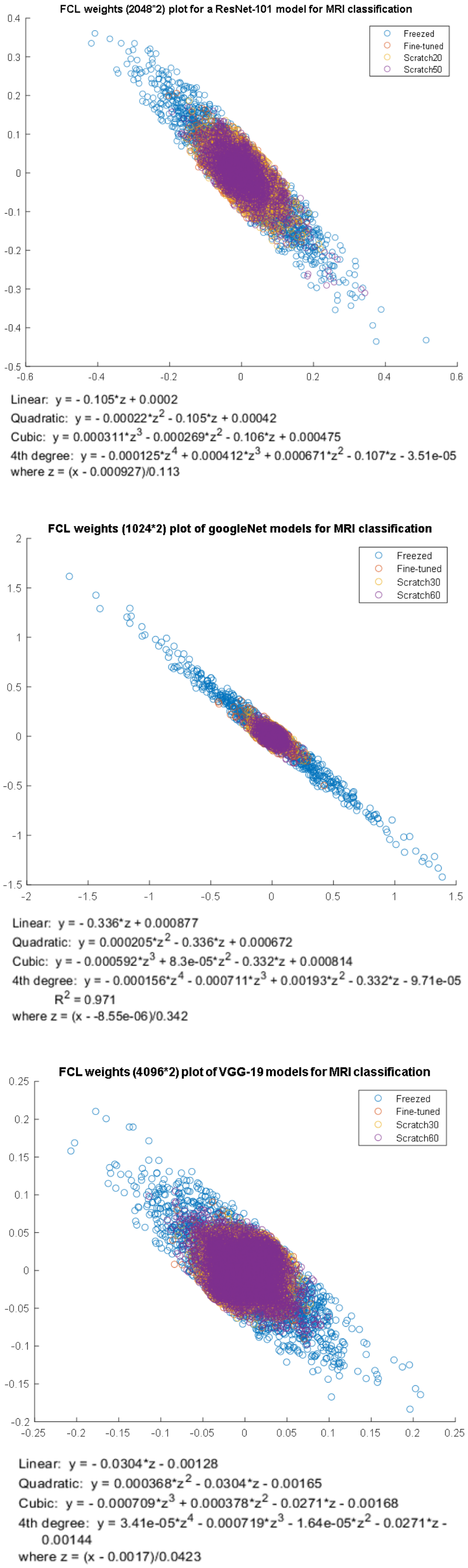


Fig. 3. Weight dispersion pattern of 3 DNN models from top to bottom ResNet101, GoogleNet, and VGG-19. Each of these figures shows the plot of FCL weights of a trained model.

Here more important is the feature distribution pattern as shown in Fig. 2. Fig. 2 shows how the feature distribution varies from complete weight transfer to no weight transfer. Since the freezed model uses pre-trained weights obtained by training from IMAGENET images [14], it is not very supportive for MRI classification (Fig. 2(a)) so, it requires fine-tuning to change the weights to converge into 2 classes properly which is as shown in Fig. 2(b). Fig. 2(c) and Fig. 2(d) show the result of scratch training, features are not properly distinguished and seems condensed with under-training i.e., for only 20 epoch. However, features start to be sparsely distributed under full training i.e., 50 epochs. Similarly, Fig. 3 shows the weight plots of each model. The number of weights for input for FCL is 1,024, 2,048, and 4,096 for GoogleNet, ResNet101, and VGG-19 respectively. With the higher number of inputs being encoded to the smaller output (i.e., 2), we might lose lots of spatial information due to a tremendous reduction in dimension. When using a pre-trained model along with FCL weights we need to condense the high number of input variables into very small output variables equal to the number of classes. As well as the congestion of high input to low output also raises a bottleneck problem, which brings difficulty in encoding and reduces the variability of outputs.

IV. CONCLUSION

In this work, we tried to analyze the class correlation of weights from FCL of various CNN-trained models. This also shows how the architecture plays role in giving classification accuracy along with its length and depth. Consequently, this work is just an attempt to understand how the flattening process works. We tried to analyze the 2D feature distribution process in DNN and besides tried to analyze the weights dispersion pattern. This is an initial work; we hope we can endeavor more understanding of these phenomena in the future.

ACKNOWLEDGMENT

This study was supported by research funds from Chosun University, 2022.

REFERENCES

[1] N. Tajbakhsh, J. Y. Shin, R. Suryakanth, R. Gurudu, R. T. Hurst, and C. B. Kendall, et al., "Convolutional neural networks for medical image analysis: Full training or fine tuning?," *arXiv*, vol. 35, no. 5, pp. 1299-1312, 2017.

[2] S. Bozinovski, "Reminder of the first paper on transfer learning in neural networks, 1976," *Informatica*, vol. 44, no. 3, pp. 291-302, 2020.

[3] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," *IEEE Computure Society Confernce Computer Visioin Pattern Recognition Work*, pp. 512-519, 2014.

[4] B. Cheng, M. Liu, D. Shen, Z. Li, and D. Zhang, "Multi-domain transfer learning for early diagnosis of Alzheimer's disease," *Neuroinformatics*, vol. 15, no. 2, pp. 115-132, 2017.

[5] B. Khagi, C. G. Lee, and G. R. Kwon, "Alzheimer's disease classification from brain MRI based on transfer learning from CNN," in *BMEiCON 2018 - 11th Biomed. Engineering International Confernece*, 2019.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing System*, vol. 25, pp. 1097-1105, 2012.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770-778.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv Prepr. arXiv1409.1556*, 2014.

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, and D. Anguelov, et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1-9.

[10] A. Labatie, "Characterizing weil-behaved vs. pathological deep neural networks," in *36th International Confernece Machine Learning ICML 2019*, Jun. 2019. pp. 6396-6406,

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 2017.

[12] H. C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, and I. Nogues, et al., "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Transaction on Medical Imaging*, vol. 35, no. 5, pp. 1285-1298, 2016.

[13] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249-256.

[14] L. F. FEI and J. DENG, "Where have we been ? Where

are we going ? The beginning: CVPR 2009," *Imagenet Work*, 2017.

AUTHORS

Bijen Khagi received his Ph.D. from the Department of Information and Communication Engineering, Chosun University in 2022. Currently, he has been working as a Post-Doc at Chosun University. His research interests include Artificial Neural Networks, Artificial Intelligence systems, Deep Learning, and Machine Learning on Image Processing, especially in Medical Image Analysis.

Goo-Rak Kwon received a Ph.D. from the Department of Mechatronic Engineering, Korea University, in 2007. He served as Chief Executive Officer and the Director of Dalitech Co. Ltd. from 2004 to 2007. He joined the Department of Electronic Engineering, Korea University, from 2007 to 2008, where he was a Postdoctoral Researcher supporting the BK21 Information Technique Business. He has been a Professor at Chosun University, since 2017. He has also been an Associate Dean with the Industry-academic Cooperation Foundation, since 2018. He has contributed 66 and 91 articles to journals and conference proceedings, respectively. He also holds 34 patents on medical image analysis and the security of multimedia contents for digital rights management. He was a member of the IEICE and IS&T international institutes. In domestic institutes, he was a member of the signal processing society in the IEIE, KMMS, KIPS, and KICS. His research interests include medical image analysis, A/V signal processing, video communication, and applications.