J Inf Process Syst, Vol.16, No.6, pp.1293~1308, December 2020
https://doi.org/10.3745/JIPS.04.0194

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Mid-level Feature Extraction Method Based Transfer Learning to Small-Scale Dataset of Medical Images with Visualizing Analysis

Dong-Ho Lee*, Yan Li*, and Byeong-Seok Shin*

Abstract

In fine-tuning-based transfer learning, the size of the dataset may affect learning accuracy. When a dataset scale is small, fine-tuning-based transfer-learning methods use high computing costs, similar to a large-scale dataset. We propose a mid-level feature extractor that retrains only the mid-level convolutional layers, resulting in increased efficiency and reduced computing costs. This mid-level feature extractor is likely to provide an effective alternative in training a small-scale medical image dataset. The performance of the mid-level feature extractor is compared with the performance of low- and high-level feature extractors, as well as the fine-tuning method. First, the mid-level feature extractor takes a shorter time to converge than other methods do. Second, it shows good accuracy in validation loss evaluation. Third, it obtains an area under the ROC curve (AUC) of 0.87 in an untrained test dataset that is very different from the training dataset. Fourth, it extracts more clear feature maps about shape and part of the chest in the X-ray than fine-tuning method.

Keywords

Feature Extraction, Medical Imaging, Transfer Learning

1. Introduction

Convolutional neural networks (CNN) [1,2] have shown outstanding performance in body detection including facial recognition [3-7] and disease classification [8-10]. Several studies [11,12] have demonstrated that the pre-trained model in ImageNet can be transferred to medical images. Transfer learning is a widely used method for training datasets [13]. By using a model that has been pre-trained with sufficiently large-scale datasets during the initialization process of features, small-scale datasets can be trained more efficiently than they could have been from scratch. However, recent CNN models have very deep structures, with more than hundreds of layers, so it is not easy to retrain these models with small-scale datasets.

Generally, it is difficult to collect large-scale medical imaging datasets due to patient privacy or ethics-related concerns. In the case of rare diseases, small-scale datasets are inevitable. Furthermore, the networks for medical images are applied locally (e.g., a local community). In other words, it is sometimes unnecessary to collect large-scale datasets, such as those collected by ImageNet [14].

For example, a hospital in one country does not need much information about patients from other

Manuscript received January 29, 2019; first revision August 19, 2019; accepted September 22, 2019.
Corresponding Author: Byeong-Seok Shin (bsshin@inha.ac.kr)
* Dept. of Computer Science and Engineering, Inha University, Incheon, Korea (dongholab13@gmail.com, leeyeon@inha.ac.kr, bsshin@inha.ac.kr)

countries. In addition, local hospitals only need information from local residents. Furthermore, it is often the case that the necessary infrastructure for collecting medical images is not well established. Therefore, it is necessary to consider an effective deep learning method for the training of small-scale medical imaging datasets.

Fine-tuning is a method that retrains all layers and classifiers within a network. It is a method that is typically used in datasets from various fields within transfer learning. However, the possibility of overfitting is very high when fine-tuning the pre-trained model to a small-scale dataset. To address this limitation, we utilized feature extraction (FE). FE is a method that modifies last fully connected layer to fit the number of output classes of dataset and maintain existing parameters of convolutional layers without updating them [15]. The FE method for transfer learning can be thought of in two ways: (1) to prevent updates of all convolutional layers, to only extract features, and then to train only the linear classifiers, and (2) to divide the convolutional layer into levels and then to train each level selectively. The convolutional layers can be divided into a low level, a middle level, and a high level, with deeper-level extraction displaying features that are more complex and closer to the shape of the object [16]. Because layers at each level have different features, it is necessary to determine which level is the most effective at retraining small-scale medical imaging datasets. Studies using transfer-learning for medical imaging [11,12,17,18] all have fine-tuning method to study all layers. All studies show good performance but fine-tuning to small-scale medical imaging dataset can lead to overfitting, and updating all layers consumes unnecessary computing power.

Therefore, we devised a method that consumes less computing power without overfitting in learning of small-scale medical image dataset. We propose that mid-level FE, which retrains features such as the texture of the image and parts of the object, is the most effective model. We compared the transfer-learning model, implemented with fine-tuning, and the FE of each level to the proposed method. And we visualize and compare all convolutional layers in mid-level FE and fine-tuning model. As a result, we confirmed that our model has the ability to conduct efficient, small-scale medical image analysis and it clearly extracts the features of the medical image.

In this paper, we present contributions as follows:

- For training a small-scale medical image dataset, we propose mid-level FE method that only retrain middle level layers. Our proposed method shows good classification accuracy and reduce computation power by showing the fastest convergence than another baseline. This method also is robust to unseen dataset.
- Through visualizing layers of network, we confirmed that our network train valid features of lesion area.

The rest of the paper is organized as follows: in Section 2, we provide some related works in structure of transfer-learning, several studies applied transfer-learning to medical imaging, and visualizing method for CNN. We describe detailed design of our proposed network which layers are updated and rationale of that based on visualization studies in Section 3. A comparison of the performances of the networks efficiency and results of visualizing all layers in the network in Section 4. And we present the conclusions in Section 5.

2. Related Works

Before CNN have shown tremendous performance, most studies applied pattern recognition or

machine learning approach to computer aided diagnosis (CAD). There are studies that apply local binary pattern (LBP) [19] and bag-of-visual-words (BoVW) [20,21] to the disease classification, and there is also a disease classification study using SVM [22] method. After the advent of AlexNet [2], CNN has been used in most CAD applications, many researchers are studying CNN's excellent FE capabilities.

2.1 Transfer Learning

CNN has a significant number of parameters, so it needs large-scale datasets to learn it. Transfer learning [23] is a means of training new datasets that are limited in size. Rather than randomly initializing a new network, the parameters of the pre-trained model are called up and used during the initialization process. In this case, the pre-trained model should be learned in a very large dataset in order to guarantee good performance. In one study, researchers [23] conducted an experiment to transfer the parameters of AlexNet [2], as learned in ImageNet [24], to Pascal VOC 2007 and 2012 datasets [25]. Their results demonstrated that the transfer-learning model from ImageNet is much better than a random initialization model. In other words, instead of random initializing, transfer learning can be used effectively for learning a small-scale dataset by transferring the model trained with a huge dataset. In terms of implementation, transfer learning can be implemented through the use of fine-tuning, which updates all the layers to adjust to a new dataset, and FE, which extracts input features [15] and learns only the fully connected layer. When implemented with FE, in addition to learning only the fully connected layer, FE can specify which layer to learn. This is a very important process. Fig. 1 shows the transfer-learning structure as fine-tuning and FE.

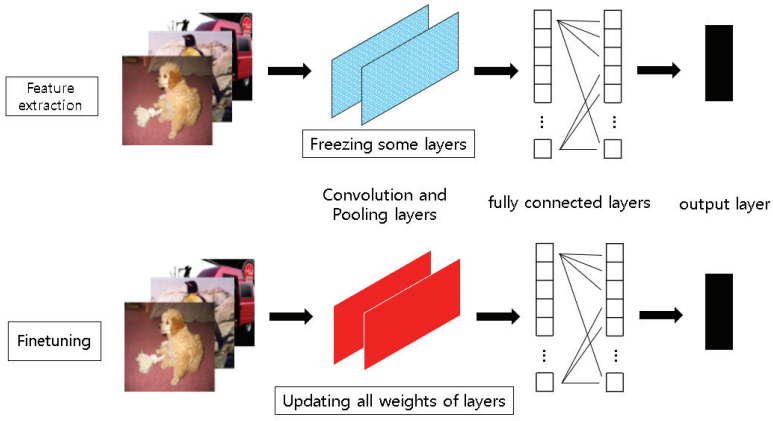

Fig. 1. Transfer-learning structure as FE and fine-tuning.

2.2 Transfer Learning for Medical Imaging

Research applying transfer learning to the learning of medical images has become increasingly common. Bar et al. [11] applied AlexNet [2] within ImageNet to chest X-rays. They compared the CNN structure with other methods, namely, GIST [26] and BoVW [27]. Of these, CNN performance was the best. They thus demonstrated that it is possible to transfer non-medical datasets to medical imaging datasets. In addition, CheXNet [17] trained pneumonia pathology in chest X-rays with better accuracy than doctors could produce using transfer learning. They chose a DenseNet [28] model pre-trained with

ImageNet and updated all the layers. Various studies have also been carried out on datasets other than X-ray datasets. Tan et al. [18] studied classification in bronchoscopy images using sequential fine-tuning based on DenseNet, obtaining good results. Shin et al. [12] showed that CifarNet [29], AlexNet, and GoogLeNet [30] learned in ImageNet can be used as pre-trained models for transfer learning to CT datasets. However, all the aforementioned studies have implemented transfer learning through fine-tuning, and they have not been compared with each convolutional layer level. Therefore, we should consider whether fine-tuning is appropriate for small-scale medical image sets and compare the training results according to the characteristics of each convolutional layer level.

2.3 Visualizing a CNN Network

AlexNet is an early deep CNN model, but its structure is simple and its performance is good, so it is actively used in various studies. The aforementioned studies have also conducted transfer learning using AlexNet. AlexNet has a structure simpler than the latest deep CNN network, consisting of five convolution layers, three max-pooling layers, and three fully connected layers.

However, compared to non-CNN models like general neural network models, AlexNet is not overly simple. The CNN model has a complex structure, which makes it difficult to understand intuitively the process of learning CNN. For example, FE, the operations between layers, and the very high dimensional tensor make it hard to see if the network is being learned properly. Therefore, to interpret the learning process of CNN networks, many efforts have been made to visualize inside the network [11,12,16,31]. Many researchers [16,31,32] used AlexNet in CNN visualizing studies. Through these studies, we can see that the higher-level layers in the network learn object-specific features, and the lower-level layers in the network learn more general features, such as lines, edges, corners, etc.

Chitra et al. [10] have proposed a deconvolution structure that visualized the feature map by unpooling methods. Unpooling is the process of returning the rectified feature map through the rectified linear units (ReLU) function back to the original image dimensions. Specifically, unpooling proceeds by the following two passes. First, the pooled location is stored in the max-pooling process (the variable that stores the location is called the switch). Second, the feature map is restructured by arranging the stored switches to the appropriate locations. Once the unpooled feature maps are obtained, they need to be rectified through ReLU. This process of rectification ensures the activation map. Finally, the refined features undergo filtering. Filtering is performed using the reversed version of the convolutional layer and the transposed version of the filter of the existing convolutional layer [31]. The result of visualizing each convolutional layer can be obtained by performing all the unpooling, rectification, and filtering processes described above. In this paper, we note the mid-level FE capabilities of CNN and visualize our model through the deconvolution structure of Zeiler et al. [16].

3. Mid-level FE for Medical Image Transfer Learning

The CNN model effectively extracts mid-level image representation. However, as the parameters in training this type of network are very large, learning from an insufficient number of small-scale datasets can lead to overfitting [6]. This problem can be similarly applied to fine-tuning, which retrains all layers. Medical images are limited in their dataset size; furthermore, their features are completely different from

real-life images, such as are present in ImageNet. As a result, we determined to explore a method that extracts mid-level image features effectively and prevents the overfitting of small-scale datasets.

First, we considered utilizing the construct transfer-learning network with the FE method to prevent overfitting by reducing the number of parameters to retrain. Second, we selected which convolutional layer we planned to retrain and update. The low-level layers showed the edge and color of the image, the mid-level layers represented the texture of the image or the specific parts of the objects, and the high-level layers showed the larger part of the objects or the entire objects [8,10,16]. Therefore, we reasoned that the mid-level layers could represent both common and more class-specific features. As a result, we determined to use the mid-level FE method.

Fig. 2 shows the entire network structure of the mid-level FE. The network is based on using AlexNet [2] in ImageNet, which has a simple structure that allows for the comparison of the effects of training the mid-level convolutional layer (CL). According to [9], AlexNet's CL3 and CL4 refer to textures, such as a mesh pattern, and more complex and class-specific features of the objects in the image. We only trained these two layers. That is, we chose CL3 and CL4 as the mid-level layers from the five CLs of AlexNet. We updated our network by retraining only the mid-level layers, CL3, CL4, and the last fully connected layer. CL1 and CL2, which are low-level layers, and CL5, which is a high-level layer, were frozen, and they kept the parameters of AlexNet within ImageNet. In addition, the classifiers (FC1 and FC2) were frozen, and only FC3 was updated.

The benefit of layer freezing also appears in computation. Table 1 shows the detailed CL architecture of our mid-level FE. The number of parameters of CL3 and CL4 occupy 62% of the total CL parameters. By training the mid-level layers, we can greatly reduce the number of parameters to update.

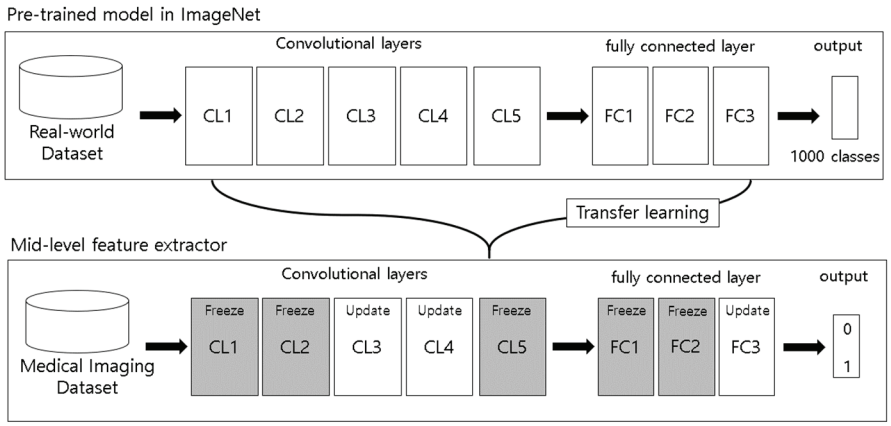

Fig. 2. Mid-level FE structure. Our network is trained on ImageNet. We only train mid-level layers (CL3, CL4) and the last fully connected layer.

Table 1. The architecture of convolutional layers (CL) in mid-level feature extractor

Layer	Input size	Parameters	Parameters to update	Kernel size
CL1	224×224×3	23,296	0	11×11
CL2	56×56×96	307,392	0	5×5
CL3	27×27×196	663,936	663,936	3×3
CL4	13×13×384	884,992	884,992	3×3
CL5	13×13×256	590,080	0	3×3

3.1 Dataset and Training

Our dataset included frontal chest X-ray image data, labeled pulmonary tuberculosis (TB) or non-TB from [33]. There were two datasets, the Shenzhen dataset, from Shenzhen No. 3, People's Hospital (Guangdong, China), and the Montgomery dataset, from the Department of Health and Human Services of Montgomery County (Rockville, MD, USA). While the Shenzhen dataset consisted of 336 TB and 326 non-TB, the Montgomery dataset consisted of 103 TB and 296 non-TB. In addition, the Shenzhen dataset was used as a training and validation set. It was randomly split into training (80%) and validation (20%) sets. In addition, the Montgomery dataset was used only as a test set to examine the possibility of overfitting because the features of the Montgomery dataset are very different from those of the Shenzhen dataset. These two datasets have very different distributions, as shown in Fig. 3. On the basis of the AlexNet's input size, all input images in the dataset were converted to a size of 224×224. We also normalized all the images using average and standard deviation from the ImageNet dataset.

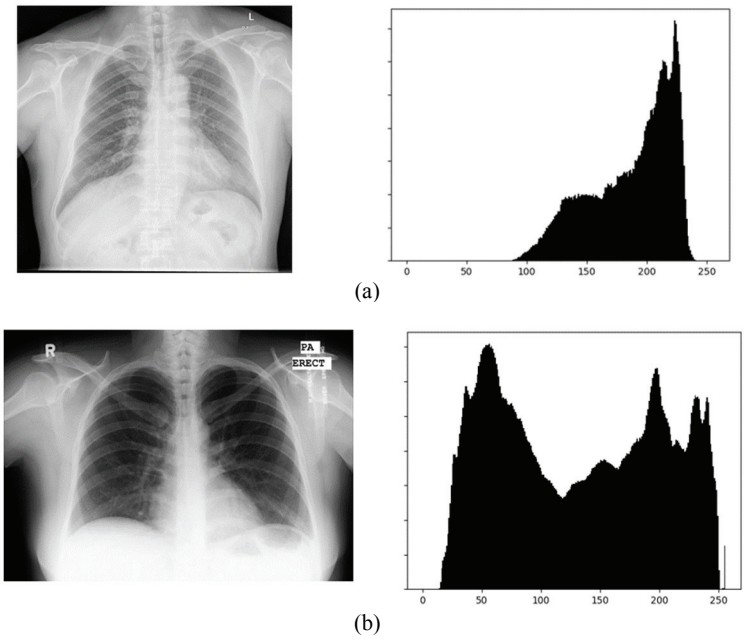

Fig. 3. Gray-scale histograms of (a) Shenzhen dataset and (b) Montgomery dataset.

We experimented with the performance of our methods, mid-level FE, by implementing a classification model of TB. This is a binary classification that registers the existence or absence of TB as 1 or 0 labels for each input datum. Thus, the output of our model should be one class of TB. Our loss function is un-weighted binary cross-entropy for our binary classification problem, and it is defined as

$$l(x, y) = [\sum_{n=1}^{N} \{ y_n \cdot \log x_n + (1 - y_n) \cdot \log(1 - x_n) \}] / N, \text{ where } y_n \in \{0,1\} \quad (1)$$

Here x_n is frontal chest X-ray data, y_n is output that only be 0 or 1 and N is the batch size by using the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Our initial learning rate was 0.001, decaying by 0.1 every 7 epochs.

The experiment was separated into nine cases, depending on whether each CL was frozen. Specifically, it was divided into each level FE and each layer FE. The low-level FE was up to CL2, mid-level was up to CL4, and high-level was up to CL5. Table 2 shows which layers were updated or not updated during the training. For example, mid-level FE, our method, involved updating only the CL3 and CL4 layers. The term CL3-FE in the table means that only the CL3 layer was updated, the term high-level FE means that only the CL5 layer was updated, and the term H-M level FE means that the high- and mid-level layers, from CL3 to CL5, were updated. Dividing the experiment cases into levels and layers shows which levels and layers are most effective for learning a small-scale medical images dataset. All cases were designed to allow for the updating of only the 3rd (last) fully connected layer, assuming the possibility of similar experiments in more restrictive situations.

Table 2. The nine experiment cases, indicating update/freeze status

Case	CL1	CL2	CL3	CL4	CL5
Mid-level FE	Freeze	Freeze	Update	Update	Freeze
CL3-FE	Freeze	Freeze	Update	Freeze	Freeze
CL4-FE	Freeze	Freeze	Freeze	Update	Freeze
Fine tuning	Update	Update	Update	Update	Update
Low-level FE	Update	Update	Freeze	Freeze	Freeze
CL1-FE	Update	Freeze	Freeze	Freeze	Freeze
CL2-FE	Freeze	Update	Freeze	Freeze	Freeze
High-level FE	Freeze	Freeze	Freeze	Freeze	Update
H-M level FE	Freeze	Freeze	Update	Update	Update

3.2 Visualizing Network

Fig. 4 shows our feature visualization structure. First, convolution proceeds according to the structure of AlexNet. The features generated from 1st CL and 2nd CL are passed consecutively through the ReLU function and max pooling. Similarly, features created in the 3rd, 4th, and 5th CLs passed through ReLU. However, last 3rd max pooling is performed at the end of the convolution path, after the last 5th CL. The features created in the last max-pooling layer are entered into the deconvolution path through max unpooling. The CLs of the deconvolution path are the transposed layers of the existing CLs. We call this the deconvolution layer [16]. For example, if the input of the 1st CL is 3 and the output is 63, the 1st deconvolutional layer has the converse inputs and outputs of 63 and 3. Thus, the deconvolution path visualizes the feature map by reversing the convolution process. This deconvolution process can be considered to be symmetrically constructed as a U-Net [34] and SegNet [35]. However, the deconvolution structure implemented in this paper is not a symmetric model. The convolution path and the deconvolution path are performed independently. Because training both convolutional and deconvolution path requires a lot of computing power, we simplified deconvolution structure for visualizing network. Our deconvolution layers are shared by transposing the weight of the paired convolutional layers without any training.

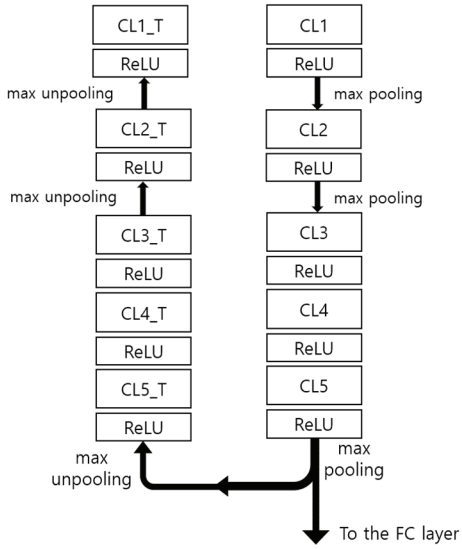

Fig. 4. The structure of simplified deconvolution network for visualizing.

4. Evaluation and Analysis

The experiment of this paper is divided into three parts using nine cases in Table 2. First, comparison of validation loss, second, accuracy and overfitting test through comparative receiver operating characteristic (ROC) curves, third, visualization of each layer.

4.1 Validation Loss

Fig. 5 compares the validation losses for each level's FE and fine-tuning, and Fig. 6 compares the validation losses for each layer's FE. The mid-level FE (ours) shows a stable convergence at 80 epochs; as a result, all the graphs are compared to epoch 80. The loss was the lowest and most stable for FE in the mid-level. Mid-level FE (in Fig. 5), CL3-FE, and CL4-FE (in Fig. 6), which updated in the middle layer and extracted class-specific and generic features, show good performance, compared to other methods. Mid-level FE shows the lowest losses, demonstrating a maximum of 0.4 and a minimum of 0.02. Furthermore, it displays a stable tendency to converge after 60 epochs. CL3-FE and CL4-FE show low losses in Fig. 6. In particular, CL3-FE demonstrates a minimum of 0.024, a small difference from mid-level FE. However, CL3-FE was less stable than mid-level FE. Its loss values rose to the 0.5 in training. This result shows that mid-level learning is more effective than learning each single layer in the middle level. Meanwhile, the fine-tuning case that updates all layers (in Fig. 5) demonstrates that its learning is unstable and that its loss is high. Because the dataset was small, it lacked an epoch. We increased the epoch to 200, but they did not converge.

This shows that fine-tuning is not suitable for the learning of small-scale medical images datasets. High-level FE (in Figs. 5 and 6) and H-M level FE (in Fig. 5), which updated the upper layer in the network, tends in general to be trained stably, except for the fact that loss values fluctuate at the beginning of the training in the high-level FE case. In addition, their loss values stay higher than those of mid-layer FE. This result shows that training high-level layers that extract entire objects of input is useful for small-

scale medical image sets but not as good as training the mid-level layers. Low-level FE (in Fig. 5), CL1-FE, and CL2-FE (in Fig. 6) were poorly trained cases; their losses were too high, and they could not be considered to be stable. That is, small-scale medical image datasets are insufficient in the training of low-level layers. Because low-level features from one medical image (such as lines or patterns) are too general, they are not distinguishable from the features of other images.

Therefore, we confirmed that learning mid-level layers is more effective than learning other levels. This shows that the mid-level FE has the smallest number of epochs to converge, in other words, has lowest computing costs, has the best loss performance, and has a stable learning tendency.

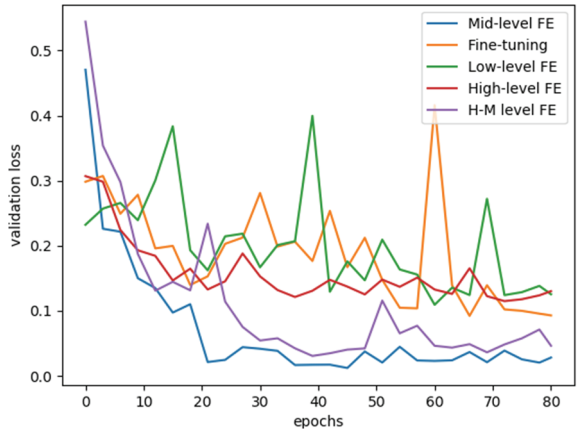

Fig. 5. Validation loss graph for each level's FE.

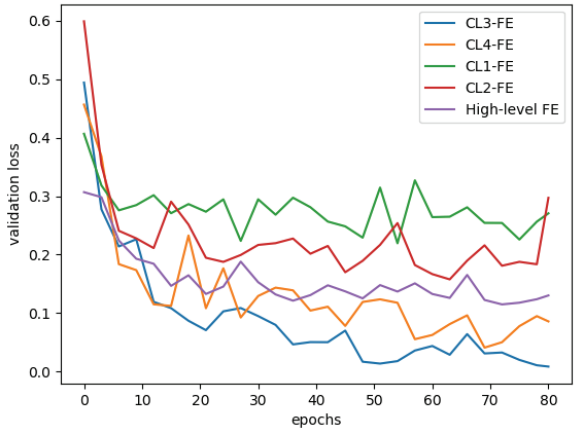

Fig. 6. Validation loss graph for each layer's FE.

4.2 Overfitting Test with ROC Curve

Fig. 7 presents ROC curves performed on the Montgomery test set. In addition, Table 3 shows the AUC values of each case. The area under the ROC curve (AUC) of the mid-level FE is 0.87, and the AUC of the fine-tuning is 0.77; in other words, the difference is 0.1, and the accuracy of mid-level FE is higher than the fine-tuning. This demonstrates that our method outperforms the fine-tuning method. In addition, it can be observed that mid-level FE does not overfit the original training dataset. As shown in

the previous validation loss experiment, low-level FE performance is very low. This implies that overfitting has occurred. The H-M level FE performance is better than fine-tuning. However, it is lower than the mid-level FE. Therefore, our proposed method, mid-level FE, showed the best performance to prevent overfitting.

We found the mid-level FE to be the most effective method in implementing transfer learning for small-scale medical images. The mid-level FE trained more stably at the mid-level than did the model learned at the other levels. In addition, it demonstrated the lowest loss performance and fastest convergence in comparison with the other cases. Therefore, the mid-level FE can be an effective means of training small-scale medical images.

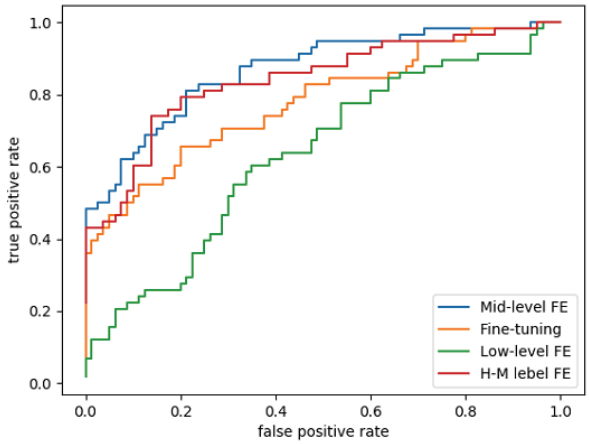

Fig. 7. Receiver operating characteristic curves of mid-FE and fine-tuning on test set.

Table 3. AUC of each level FE

Case	AUC
Mid-level FE	0.87
Fine-tuning	0.77
H-M level FE	0.84
Low-level FE	0.63

4.3 Visualizing the Mid-level Feature Extractor

Through two previous experiments, we found the mid-level FE to be the most effective method in implementing transfer learning for small-scale medical images. The mid-level FE trained more stably at the mid-level did than the model learned at the other levels. In addition, it demonstrated the lowest loss performance and fastest convergence in comparison with the other cases. Therefore, this result demonstrates that the mid-level FE can be an effective means of training small-scale medical images. We visualized the feature map inside our network to see which features the network was learning. It explains why our method performs well.

We visualized input images through the deconvolution structure. We arbitrarily selected nine input images and randomly selected one of the feature maps extracted from the input image. Figs. 8 and 9 show the results of visualization in the mid-level FE. The features of low-level layers (CL1, CL2) are shown as the lines of the ribs and vertebrae. In addition, we can see contours of the lungs. Without the layer

update, it catches the features well because the low-level layers learn general features, such as lines, edges, and corners (in Fig. 8). The mid-level layer (CL3, CL4) with updating of parameters finds more object-specific features than the low-level layer does. The visualization results below show that the right and left lungs are activated. This shows that transferring the parameters from the real-world dataset (ImageNet) to the medical imaging dataset was successfully performed (in Fig. 9). In the case of the high-level layer (CL5), the entire lung is caught. Even though layer updating is not performed, it shows that the feature is properly activated (in Fig. 9).

Fig. 10 is a visualization result of the CL3 and CL4 of the fine-tuning model. Unlike the visualization results of the proposed method, the features did not appear correctly on all layers in the fine-tuning. In other words, the performance of the pre-trained model has been lost through inadequate fine-tuning. This result shows that fine-tuning performed on a small-scale medical image dataset is very dangerous. Fine-tuning can be the worst approach unless a dataset capable of updating all the parameters of the model is guaranteed.

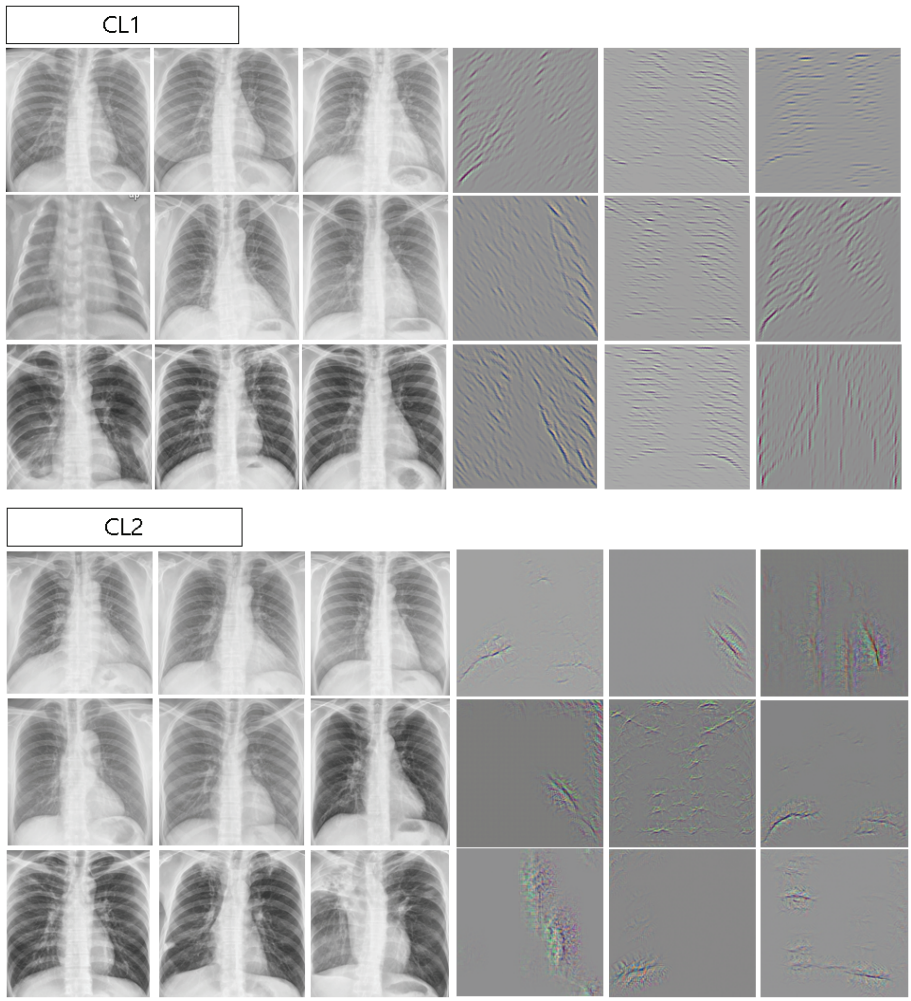

Fig. 8. Visualization results for low-level convolutional layers (CL1, CL2) in the mid-level feature extractor.

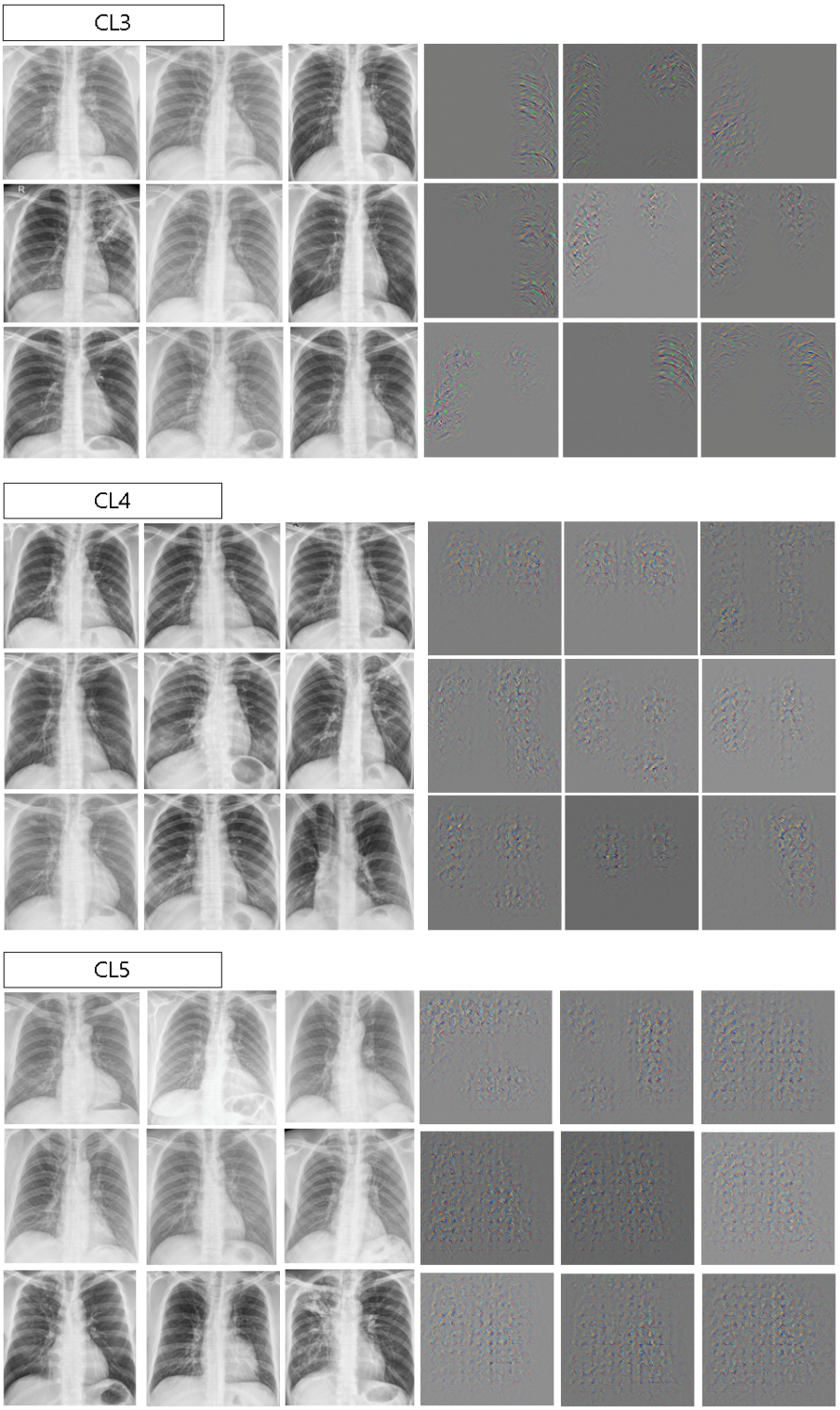

Fig. 9. Visualization results for mid-level convolutional layers (CL3, CL4) and high-level convolutional layer (CL5) in the mid-level feature extractor.

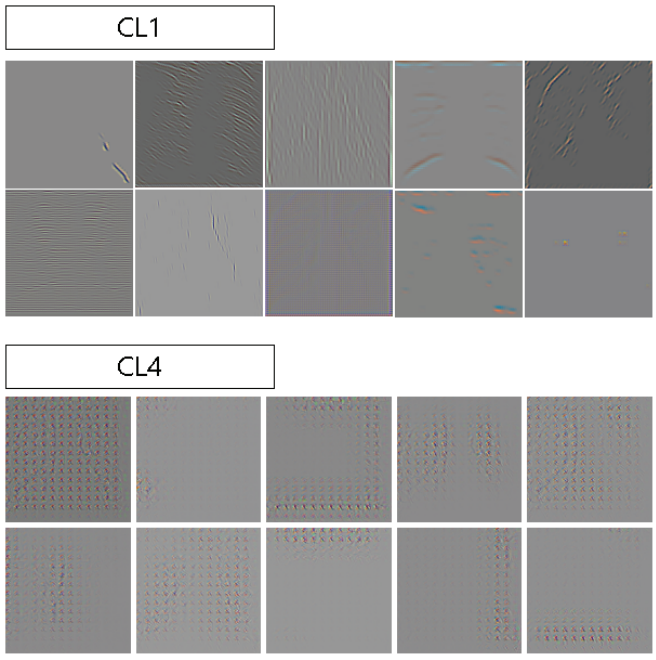

Fig. 10. Visualization results of CL 1 and CL 4 in the fine-tuning model.

5. Conclusion

We propose a mid-level FE approach to the training of small-scale medical imaging datasets. To evaluate the performance of this method, we compared it with low-level FE, high-level FE, and fine-tuning methods. Compared with the other methods, our proposed method shows the lowest amount of loss between 0.4 and 0.02, the most stable training tendency, and the lowest computing costs for convergence. In the experiment pertaining to overfitting, we used different datasets from the training set; the AUC obtained from the test is 0.87. Our method also prevents overfitting; our results are 0.1 higher than the fine-tuning method. We also conducted a visualization experiment of convolution layers in our method using a deconvolution structure. Our method can be verified to extract meaningful features throughout the network. On the other hand, the fine-tuning method did not extract the features correctly on all layers. Thus, our method is shown to be an efficient alternative to the classification of small-scale medical imaging datasets through its prevention of overfitting, its maintenance of accuracy, and reduction in computing costs. For future work, we need to research the training of neurons inside the layer. If we can selectively train valid neurons inside the mid-level layers, more efficient learning is possible.

Acknowledgement

This work was supported by the Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korean government (MSIT) (No. 2017-0-018715, Development of AR-based Surgery Toolkit and Applications).

References

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[3] C. Li, M. Liang, W. Song, and K. Xiao, "A multi-scale parallel convolutional neural network based intelligent human identification using face information," *Journal of Information Processing Systems*, vol. 14, no. 6, pp. 1494–1507, 2018.

[4] S. Zhou and S. Xiao, "3D face recognition: a survey," *Human-centric Computing and Information Sciences*, vol. 8, article no. 35, 2018.

[5] K. M Koo and E. Y. Cha, "Image recognition performance enhancements using image normalization," *Human-centric Computing and Information Sciences*, vol. 7, article no. 33, 2017.

[6] A. Sun, Y. Li, Y. M. Huang, Q. Li, and G. Lu, "Facial expression recognition using optimized active regions," *Human-centric Computing and Information Sciences*, vol. 8, article no. 33, 2018.

[7] J. Zhang, X. Jin, Y. Liu, A. K. Sangaiah, and J. Wang, "Small sample face recognition algorithm based on novel Siamese network," *Journal of Information Processing Systems*, vol. 14, no. 6, pp. 1464–1479, 2018.

[8] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[9] S. Sarraf, G. Tofighi, and Alzheimer's Disease Neuroimaging Initiative, "DeepAD: Alzheimer's disease classification via deep convolutional neural networks using MRI and fMRI," 2016 [Online]. Available: https://doi.org/10.1101/070441.

[10] R. Chitra and V. Seenivasagam, "Heart disease prediction system using supervised learning classifier," *International Journal of Software Engineering and Soft Computing*, vol. 3, no. 1, pp. 1-7, 2013.

[11] Y. Bar, I. Diamant, L. Wolf, S. Lieberman, E. Konen, and H. Greenspan, "Chest pathology detection using deep learning with non-medical training," in *Proceedings of 2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, New York, NY, 2015, pp. 294-297.

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

[13] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*. Cambridge, MA: MIT Press, 2016.

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, et al., "ImageNet large scale visual recognition challenge," 2015 [Online]. Available: https://arxiv.org/abs/1409.0575

[15] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," 2014 [Online]. Available: https://arxiv.org/abs/1403.6382.

[16] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision – ECCV 2014*. Cham, Switzerland: Springer, 2014, pp. 818-833.

[17] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, et al., "CheXnet: radiologist-level pneumonia detection on chest x-rays with deep learning," 2017 [Online]. https://arxiv.org/abs/1711.05225.

[18] T. Tan, Z. Li, H. Liu, F. G. Zanjani, Q. Ouyang, Y. Tang, et al., "Optimize transfer learning for lung diseases in bronchoscopy using a new concept: sequential fine-tuning," 2018 [Online]. Available: https://arxiv.org/abs/1802.03617.

[19] J. M. Carrillo-de-Gea and G. Garcia-Mateos, "Detection of normality/pathology on chest radiographs using LBP," in *Proceedings of the 1st International Conference on Bioinformatics*, Valencia, Spain, 2010, pp. 167-172.

[20] U. Avni, H. Greenspan, E. Konen, M. Sharon, and J. Goldberger, "X-ray categorization and retrieval on the organ and pathology level, using patch-based visual words," *IEEE Transactions on Medical Imaging*, vol. 30, no. 3, pp. 733-746, 2011.

[21] U. Avni, H. Greenspan, and J. Goldberger, "X-ray categorization and spatial localization of chest pathologies," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2011*. Heidelberg: Springer, 2011, pp. 199-206.

[22] J. Ramirez, J. M. Gorriz, D. Salas-Gonzalez, A. Romero, M. Lopez, I. Alvarez, and M. Gomez-Rio, "Computer-aided diagnosis of Alzheimer's type dementia combining support vector machines and discriminant set of features", *Information Sciences*, vol. 237, pp. 59-72, 2013

[23] M. Oquab, L. Bottou, I. Laptev, and J. Sovic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, 2014, pp. 1717–1724.

[24] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "ImageNet: a large-scale hierarchical image database," in *Proceedings of 2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, 2009, pp. 248-255.

[25] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303-338, 2010.

[26] A. Oliva and A. Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145-175, 2001.

[27] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proceedings of the 8th European Conference on Computer Vision: Workshop on Statistical Learning in Computer Vision*, Prague, Czech Republic, 2004.

[28] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2016 [Online]. Available: https://arxiv.org/abs/1608.06993.

[29] H. R. Roth, L. Lu, J. Liu, J. Yao, A. Seff, K. Cherry, L. Kim, and R. M. Summers, "Improving computer-aided detection using convolutional neural networks and random view aggregation" *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1170-1181, 2016.

[30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, 2015, pp. 1-9.

[31] Z. Qin, F. Yu, C. Liu, and X. Chen, "How convolutional neural network see the world: a survey of convolutional neural network visualization methods," *Mathematical Foundations of Computing*, vol. 1, no. 2, pp. 149-180, 2018.

[32] W. Yu, K. Yang, Y. Bai, T. Xiao, H. Yao, and Y. Rui, "Visualizing and comparing AlexNet and VGG using deconvolutional layers," in *Proceedings of the 33rd International Conference on Machine Learning*, New York, NY, 2016.

[33] S. Jaeger, S. Candemir, S. Antani, Y. X. J. Wang, P. X. Lu, and G. Thoma, "Two public chest X-ray datasets for computer-aided screening of pulmonary diseases," *Quantitative Imaging in Medicine and Surgery*, vol. 4, no. 6, pp. 475-477, 2014.

[34] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *Medical Image Computing And Computer-Assisted Intervention – MICCAI 2015*. Cham, Switzerland: Springer, 2015, pp. 234-241.

[35] V. Badrinarayanan, A. Handa, and R. Cipolla, "SegNet: a deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," 2015 [Online]. Available: https://arxiv.org/abs/1505.07293.

Dong-Ho Lee **https://orcid.org/0000-0001-9934-146X**

She has been in the Department of Computer Engineering of Inha University as a M.S. candidate since 2018. She received a B.S. degree in Industrial Engineering from Inha University in 2018. Her research interests include machine learning and Deep Learning for medical imaging.

Yan Li **https://orcid.org/0000-0003-2886-3572**

She is a professor in the Department of Computer Engineering, Inha University, Korea. She received M.S. and Ph.D. degrees from the Department of Computer Science and Information Engineering of Inha University. Her current research interests include spatial databases, GIS, deep learning, and medical imaging.

Byeong-Seok Shin **https://orcid.org/0000-0001-7742-4846**

He is a professor in the Department of Computer Engineering, Inha University, Korea. His current research interests include volume rendering, real-time graphics, virtual reality, and medical imaging. He received his B.S., M.S., and Ph.D. in Computer Engineering from Seoul National University, Korea.