IPSJ Transactions on Mathematical Modeling and Its Applications Vol.11 No.3 74–83 (Dec. 2018)

Regular Paper

Feature Representation Analysis of Deep Convolutional Neural Network using Two-stage Feature Transfer —An Application for Diffuse Lung Disease Classification

Aiga Suzuki[1,2,a)] Hidenori Sakanashi[1,b)] Shoji Kido[3,c)] Hayaru Shouno[1,4,d)]

Received: April 11, 2018, Revised: August 3, 2018,
Accepted: September 10, 2018

Abstract: Transfer learning is a machine learning technique designed to improve generalization performance by using pre-trained parameters obtained from other learning tasks. For image recognition tasks, many previous studies have reported that, when transfer learning is applied to deep neural networks, performance improves, despite having limited training data. This paper proposes a two-stage feature transfer learning method focusing on the recognition of textural medical images. During the proposed method, a model is successively trained with massive amounts of natural images, some textural images, and the target images. We applied this method to the classification task of textural X-ray computed tomography images of diffuse lung diseases. In our experiment, the two-stage feature transfer achieves the best performance compared to a from-scratch learning and a conventional single-stage feature transfer. We also investigated the robustness of the target dataset, based on size. Two-stage feature transfer shows better robustness than the other two learning methods. Moreover, we analyzed the feature representations obtained from DLDs imagery inputs for each feature transfer models using a visualization method. We showed that the two-stage feature transfer obtains both edge and textural features of DLDs, which does not occur in conventional single-stage feature transfer models.

Keywords: deep convolutional neural networks, transfer learning, image recognition, textural recognition, medical imaging

1. Introduction

In the field of computer vision and image recognition, deep convolutional neural networks (DCNNs) have been the primary model, owing to AlexNet [14] having had great success during the ImageNet competitions in 2012. DCNNs are thus becoming the de facto solution for image recognition tasks. The DCNN is a multi-layered neural network that has the same architecture as Neocognitron [6], [8], inspired by biological human visual systems. The brain's vision center has a hierarchical mechanism that understands visual stimulus [11]. The DCNN uses a similar hierarchical structure to extract features by using stacks of "convolution" and "spatial pooling" operations. The distinctive feature of a DCNN is its automation of obtaining feature representations, which suits the given tasks. Whereas DCNNs provide significant performance with image recognition tasks, they require massive amounts of training data compared to conventional machine learning models. The deep network structure exhibits higher expressive power than shallow models, which have the same complexity [2]. Alternatively, most deep models have a large number of free parameters. Han et al. reported that deep neural networks require one-tenth of the number of free parameters training data needed to obtain a good generalization capability [10]. However, when the acquisition of a training dataset is difficult (e.g., medical imagery), the amount of data will sometimes be insufficient. Generally, for learning approaches, the amount of training data has a strong effect on model performance. Deficient training data sometimes causes generalization problems such as overfittings.

A conventional approach for overcoming data deficiency is transfer learning [16]. This is a learning technique that reutilizes knowledge gained from other learning tasks, called the "*source domain*," to improve model performance in the desired task, called the "*target domain*." In the case of transfer learning for an image classification task, the model will first be trained to classify the source domain. Then, it will be trained for the target domain. In the case of DCNNs, we expect feature extraction to be improved by reutilizing its feature extraction capability. Note that this paper distinguishes two common styles of transfer learning. One is "fine-tuning," which retrains only the classification part while maintaining the feature extraction part. In other words, the fine-tuning style assumes that the feature extraction part has sufficient ability to represent input signals. The other is "feature transfer," which retrains the entire DCNN, containing the feature extraction layers, to adopt the feature extraction part for a target task. This paper focuses on the latter case of transfer learning.

[1] National Institute of Advanced Industrial Science and Technology, Tsukuba, Ibaraki 305–8560, Japan
[2] University of Tsukuba, Tsukuba, Ibaraki 307–8577, Japan
[3] Yamaguchi University, Ube, Yamaguchi 755–8505, Japan
[4] University of Electro-Communications, Chofu, Tokyo 182–8585, Japan
[a)] ai-suzuki@aist.go.jp
[b)] h.sakanashi@aist.go.jp
[c)] kido@ai.csse.yamaguchi-u.ac.jp
[d)] shouno@uec.ac.jp

In most transfer learning approaches for image recognition tasks, massive natural image datasets, such as ImageNet [5], are used as the source domain [22]. The reason a natural image dataset is usually adopted is that of the availability of pre-trained models and their known performance. However, the effectiveness of utilizing a natural image dataset when the target domain greatly differs from the natural images is slightly questionable, because features of the source domain do not appear in the target domain. Azizpour et al. suggested that the possibility of knowledge transfer is affected by similarities between the source and target domains. They reported that it is preferable that transfer learning takes in similar data [1]. However, only a few studies have focused on model performance variation by changing source and target domains, and their scope of tasks was limited to object recognition.

This paper proposes a two-stage feature transfer method that focuses on textural image recognition. By this method, a DCNN is successively trained with natural and textural images as an initial state. Subsequently, the entire DCNN, which includes not only the classification part but also the feature extraction part, is trained again with the textural target domain. We show that this type of successive and multi-domain feature transfer improves the generalization performance of the model and provides robustness with a decrease in the size of the training dataset. Moreover, we discuss the why feature transfer on DCNNs works so well. We visualize how feature representations of DCNNs derive from different feature transfer processes, and reveal that feature transfer improves feature representations of DCNNs corresponding to both source domains.

In our experiment, we applied two-stage feature transfer to a classification task of textural X-ray high-resolution computed tomography (HRCT) images of diffuse lung diseases (DLDs) and show performance improvements.

2. Related Works and Contributions

References [9], [19] applied feature transfer to the classification of DLDs and used a conventional single-staged feature transfer, which uses a natural image dataset. They reported that feature transfer improves the classification performance over learning from scratch. However, the effectiveness of the source domain was not discussed, despite noting that the targets were textural. Reference [3] proposed an ensemble method that used multiple models trained with different domains for lung disorder classification. The term, "transfer learning," in this study references fine-tuning. The essence of this method entails ensemble modeling, rather than an actual transfer process. A notable study of transfer learning in the field of medical image analysis [22], systematically surveyed and analyzed the effects of transfer learning for various types of medical images, including textural images. They compared transfer learning from natural images and several modern parameter initialization methods in various medical image classification tasks, which had limited amounts of training data. They concluded that transfer learning from natural images to medical images is possible and meaningful, despite the large difference between the source and target domains. Nonetheless, the reason transfer learning works in DCNNs is still not fully understood.

In this paper, we study two-stage feature transfer, focusing on diffuse lung disease classification, making the following contributions.

- We demonstrate the superiority of feature transfer over fine-tuning by comparing the model performance under the same source domains.
- We demonstrate how the source domain of feature transfer affects the performance of DCNNs by comparing learning-from-scratch, single-stage feature transfer, and our proposed method.
- We show that transfer learning provides robust performance with a decrease in the size of the training dataset.
- We analyze how feature representations in intermediate DCNN layers change according to the transfer processes of the feature visualization method. This change implies a DCNN mechanism of feature transfer that has not been fully researched.

3. Deep Convolutional Neural Networks (DCNNs)

DCNNs are well-known deep learning models, which are a type of multi-layered neural network, widely used in computer vision. The most common DCNNs consist of "convolutions" and "spatial pooling" layers, which serve as feature extractors, and fully-connected layers, which serve as classifiers. The set of convolution and pooling layers are defined as "stages," in the same manner described by Ref. [8]. The stages deform the input pattern into an intermediate representation, serving as a feature-extractor. Generally, DCNNs, which have several input channels, take 2D images and repeatedly transform them into feature maps via a stack of stages. **Figure 1** shows a schematic diagram of a typical DCNN.

To understand the feature extraction of DCNNs, let us consider the activation of i-th stage. Here, we denote $h_i(l, \mathbf{x})$ as an l-th channel activation, at the location, $\mathbf{x}$, in the i-th stage. Convolution layers provide convolutional filtering to derive feature maps (i.e., activations) from previous stages. The activation of the convolution layer is written as

$$h_i^{\mathrm{conv}}(k, \mathbf{x}) = \sum_{l,\mathbf{u}} g_i(k, l, \mathbf{u})\, h_{i-1}(l, \mathbf{x} - \mathbf{u}), \tag{1}$$

where k is the channel of the derived feature map, and $g_i(k, l, \mathbf{u})$ is the convolution kernel (i.e., a "*filter tensor*"). Equation (1) shows that the convolution layer makes a feature map as an inner product of a filter tensor, g_i, and all input regions. Most neural networks modulate responses of each layer with an activation function to provide a non-linearity. We chose the rectified linear unit (ReLU), commonly used in deep neural networks, as the activation function. Following the convolution layer, all feature maps, $h_i(k, \mathbf{x})$, are modulated with ReLU.

$$h_i^{\mathrm{relu}}(k, \mathbf{x}) = \max\left(0, h_i^{\mathrm{conv}}(k, \mathbf{x})\right) \tag{2}$$

The pooling layer gathers spatial neighbors to reduce the repercussions of local pattern deformations and the dimensionality of the feature map. The response to the pooling layer of the feature map, $h_i(l, x)$, is computed as

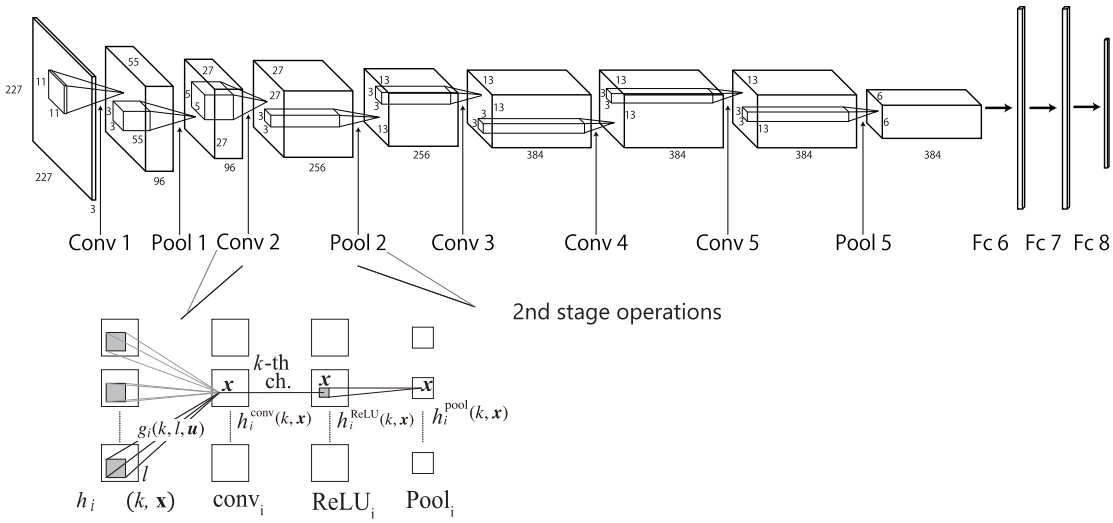

Fig. 1 Top: Schematic diagram of our DCNN, the same as Ref. [14], or *AlexNet*. Bottom: Details of the feature map construction. The DCNN acquires feature representation by repeating convolution and spatial pooling.

$$h_i^{\text{pool}}(k, \mathbf{x}) = \max_{\mathbf{r} \in N(\mathbf{x}))} (0, h_i(k, \mathbf{r})), \tag{3}$$

where $N(\mathbf{x})$ is the spatial neighbor at location, x, in the feature map. This type of pooling operation, which uses the maximum value of spatial neighbors as a representative value, is called "max-pooling."

These layers appear as early DCNN layers, which sum inputs and provide well-posed inputs for a given task. Trainable parameters of these formulations are the filter tensors, g_i.

The latter layers of DCNNs (i.e. "Fc n," in Fig. 1) are fully-connected layers. Figure 1, extracted feature representations of the input image appear as the first fully-connected layer, "Fc 6." Layers, "Fc7" and "Fc8" comprise a multi-layered perceptron, which plays the role of a classifier.

The most remarkable trait of the DCNNs is its effective feature representation, corresponding to tasks that are obtained as an intermediate representation of the feature extraction parts, consisting of convolution and spatial-pooling layers. These are obtained via a back-propagation algorithm, which minimizes classification errors.

4. Methods

4.1 Two-Stage Feature Transfer

Transfer learning is a technique that reutilizes feature expressions that come from similar tasks [16]. This paper proposes a two-stage feature transfer method focused on textural recognition tasks.

Figure 2 shows the schematic diagram of two-stage feature transfer, which, for DCNNs, means the reutilization of the feature extraction parts of the pre-trained network. These parts consist of convolution layers and no classification layers. Thus, the fully-connected layers (i.e., "Fc7" and "Fc8") are cut off from their connections, as shown in Fig. 1. After reconfiguring the network, we randomly initialize connections of the classifier part [*1]

[*1] Fully-connected weights, without a softmax layer (e.g., "Fc8") can be reused as the initial state for the transfer. In our experiment, however, the resulting performance has been worsened.

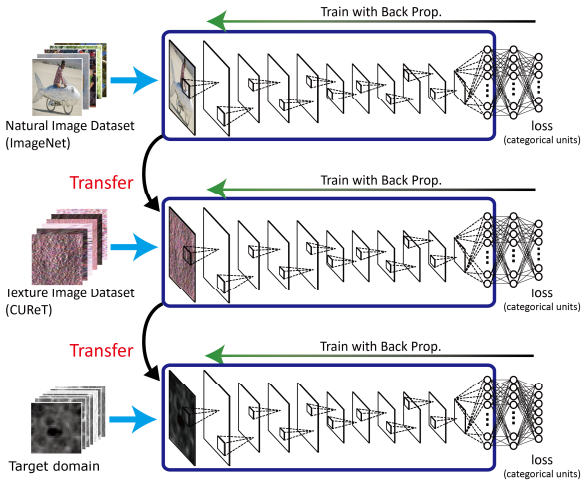

Fig. 2 Schematic diagram of two-stage feature transfer for analyzing DLD HRCT patterns. The DCNN is first trained with natural images to obtain a good feature representation as the initial state. Subsequently, it transfers to the more effective domain (i.e., texture dataset) to obtain the feature representation suited for texture-like patterns. Then, finally it trains with the target domain.

and train the entire DCNN again using back-propagation. Thus, feature transfer utilizes the feature extraction parts from other domains as its initial state.

In our proposed method, we first train the DCNN with massive natural images in the same manner as conventional feature transfer. At this stage, we expect that all connections are well-trained for extracting visual features from the input images of natural scenes, such as edge structures [14], [24]. Second, we apply feature transfer again, using the texture image dataset and natural images to acquire better feature representation and fitting for the textural images, which do not appear in the natural images.

4.2 Feature Visualization

For analysis, to understand the mechanism of feature transfer in DCNNs, and to reveal how feature transfer influences improvements, we should discuss the DCNN feature extraction process. We adopted DeSaliNet, proposed by Ref. [15], as our feature vi-

sualization method. This includes similar methods proposed by Refs. [24] and [21] as its special cases. DeSaliNet reveals which input component influences the feature representation of the feature extraction parts. **Figure 3** shows the process flow of a feature visualization using DeSaliNet.

The main idea of DeSaliNet is to propagate the feature map backward into the input space. DeSaliNet construes DCNN operations as functions and describes itself as a composite function. Let $\phi^{(i)}$ be a map to the i-th layer's feature map that we want to visualize. $\phi^{(i)}$ can thus be denoted by each layer's activation, up to the i-th layer, as

$$\phi^{(i)} = h_i^{L_i} \circ \cdots \circ h_1^{L_1}, \tag{4}$$

where L_i is the layer type, such as convolution, max-pooling, and ReLU. Here, we also denote the "backward path," $\phi^{(i)\dagger}$, which is illustrated on the left side of Fig. 3 as a pseudo-inverse map of $\phi^{(i)}$.

$$\phi^{(i)\dagger} = h_1^{L_1\dagger} \circ \cdots \circ h_i^{L_i\dagger}, \tag{5}$$

where $h_i^{L_i\dagger}$ denotes the pseudo-inverse maps associated with its corresponding layer, $h_i^{L_i}$. Details of each pseudo-inverse map are discussed in Appendix A.1.

The backward propagation of the feature map h, calculated as $\phi^{(i)\dagger}(h)$, is reconstructed as an imagery member of the input space. Components of the input image, which have a strong influence on the feature extraction, will compose the salient value of the reconstructed images, e.g., the top-left of Fig. 3. The contours of the propeller blade have salient pixels in the reconstructed input. Such saliencies help us to interpret the mechanism of the feature extraction in the DCNN.

The origin of the visualization method, based on backward propagation, is the selective attention model [7], [20]. This type of feature visualization enables us to analyze *what component is paid attention to* in input images, in contrast to GradCAM [17], which analyzes *where it is paid attention to*. Textural images are "what-based," because they do not have locality as a characteristic.

5. Materials

5.1 Target Domain

We examined the effectiveness of our proposed two-stage feature transfer method with the classification of X-ray and HRCT DLDs. DLDs is a collective term for lung disorders that can spread to large areas of the lung. X-ray HRCT is effective for finding early-stage DLDs when they are small and mild. DLD conditions are seen as textural patterns on HRCTs. In this work, these patterns are classified into seven classes: consolidations (CON), ground-glass opacities (GGO), honeycombing (HCM), reticular opacities (RET), emphysematous changes (CON), nodular opacities (NOD), and normal (NOR). These categorizations were introduced by Ref. [23]. **Figure 4** shows portions of HRCT images for each class.

The DLDs image dataset was acquired from Osaka University Hospital, Osaka, Japan. We collected 117 HRCT scans from

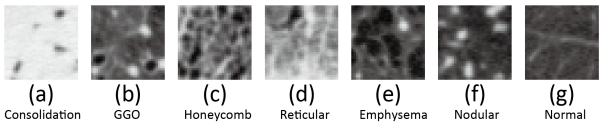

Fig. 4 Typical HRCT images of diffuse lung diseases: (a) consolidations (CON); (b) ground-glass opacities (GGO); (c) honeycombing (HCM); (d) reticular opacities (RET); (e) emphysematous changes (CON); (f) nodular opacities (NOR); and (g) normal (NOR).

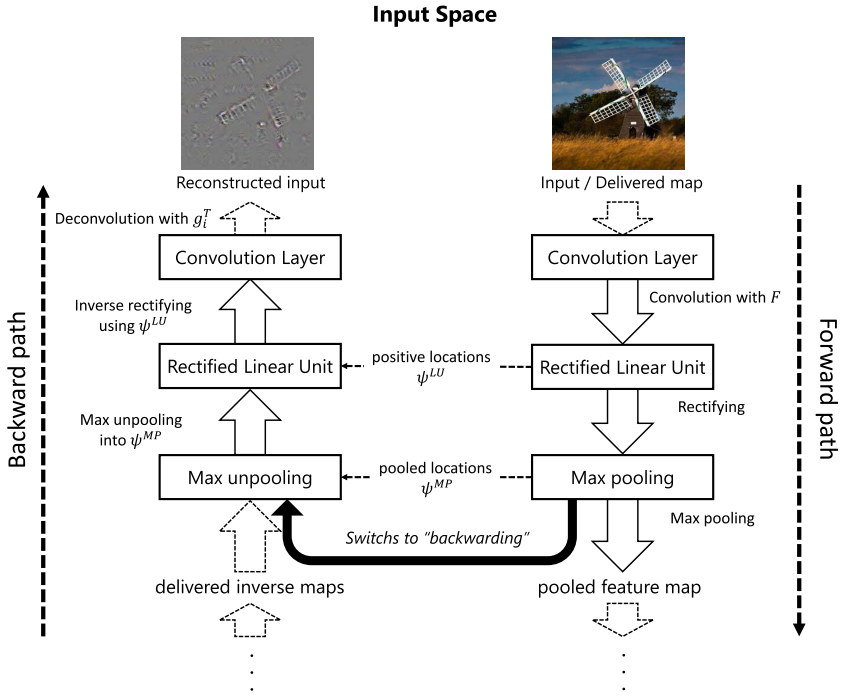

Fig. 3 A feature visualization flow using DeSaliNet. The feature map to visualize is calculated during the forward propagation stage (right). When visualizing neuronal activations, the feature map is switched to backward visualization path (left), which consists of inverse maps of each forward layers, and is backpropagated into the input space as a saliency image.

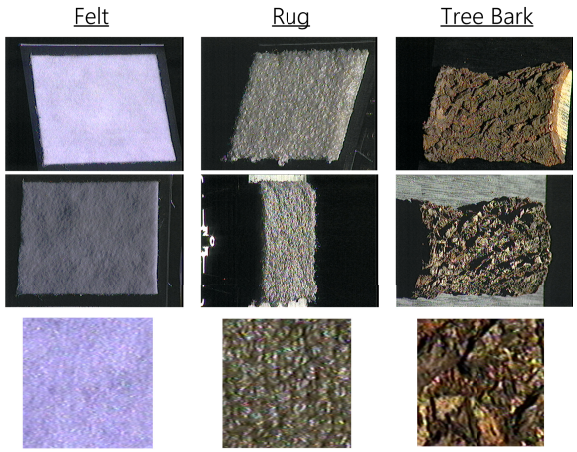

Fig. 5 Examples of textural images comprising the CUReT database. Top and middle rows: entire images of "Felt," "Rug," and "Tree Bark" classes. Bottom: cropped and resized images used as input for the DCNN.

different subjects. Each slice was converted to gray-scale images with a resolution of 512 × 512 pixels and slice-thickness of 1.0 [mm]. Lung region slices were annotated for their seven types of patterns by experienced radiologists. The annotation region shapes and their labels were the results of diagnoses by three physicians. The annotated CT images were partitioned into regions of interest (ROI) patches, which were 32 × 32 pixels, corresponding to about 4 cm^2. This is a small ROI size for DCNN input. Thus, we magnified them by 224 × 224 pixels using bicubic interpolation. Therefore, from these operations, we collected 169 patches for CON, 655 for GGO, 355 HCM, 276 for RET, 4702 for RET, 827 for NOD, and 5726 for NOR. We then divided these patches for DCNN training and evaluation, because each class does not contain patches from the same patients. For the training, we used 143 CONs, 609 GGOs, 282 HCMs, 210 RETs, 4406 EMPs, 762 NODs, and 5371 NORs. The remaining 26 CONs, 46 GGOs, 73 HCMs, 66 RETs, 296 EMPs, 65 NODs, and 355 NORs were used for the evaluation.

5.2 Source Domains

Two-stage feature transfer uses both natural image and texture datasets. We used the ILSVRC 2012 dataset, which is a subset of ImageNet [5], as the natural image dataset in the same manner as most conventional feature transfer studies [13], [18], [19]. We also used the Columbia-Utrecht Reflectance and Texture Database (CUReT) [4] as the texture dataset, as provided by Columbia University and Utrecht University. **Figure 5** shows examples of textural images in the CUReT database. The database contains macro photographs of 61 classes of real-world textures. Each class has approximately 200 samples, and each sample was imaged under various combinations of illumination and viewing angles. This database could be preferable for the textural source domain due to the various samples from coarse to fine and from simple to complex. To train DCNNs, we cropped the textured regions and resized them into 224 × 224 (Fig. 5: bottom) to accommodate the network input.

6. Experiments

The network structure used in this work is exactly the same as AlexNet [14], illustrated in Fig. 1. Because [13] reported that the models designed to classify ImageNet, including AlexNet, are capable of obtaining good performance for general tasks with feature transfer. We trained the network using momentum stochastic gradient descent with a momentum of 0.9 and a dropout rate of 0.5. When the network was trained for the first time, we set the learning rate to 0.05. Otherwise, when the feature transfer was used, we set the learning rate to 0.0005 because it was reported that small learning rate is preferable for pre-trained networks in Ref. [22]. We trained the network until obtaining training loss plateaus, as to steadily converge the network parameters.

In our experiment, we demonstrated two different ability of the feature transfer method. One was an eventual classification performance for DLDs images, discussed in Section 6.1, which measures the ability of generalization improvement for the task. The other is performance robustness with the size of the training data, discussed in Section 6.2, which measures how feature transfer improves the dynamics of the learning process.

For evaluation metrics, we used accuracy, recall, precision, and F1-score. Accuracy is the proportion of correct predictions to the total number of predictions. Recall is the fraction of samples collectively classified over the number of samples of its class. Precision is the fraction of samples correctly classified as a class, c, over all samples classified as a class c. Recall is an index of oversights, whereas precision is an index of over-detection. The F1-score is a harmonic mean between precision and recall: $\frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$. To minimize the effect of extraordinary good results, the 75th percentile values of the learning process was used as representative values for each evaluation metrics.

In our experiment, we compared models from different learning processes, as follows.

(1) Learning a randomly initialized model from scratch in the most naive way (i.e., no feature transfer)
(2) Feature transfer from textural images, i.e., CUReT database
(3) Feature transfer from natural images, i.e., ILSVRC 2012 dataset
(4) Two-stage feature transfer, training the DCNN from ILSVRC 2012 and CUReT, sequentially (*proposed*)

6.1 Classification Performance

First, we compared the classification performance of each model (1)–(4). To reveal the effectiveness of feature transfer, we also compared with fine-tuning models, which only the classification part of the DCNNs was retrained, as follows:

(a) Fine-tuning from natural images (ILSVRC 2012 dataset)
(b) Fine-tuning from textural images (CUReT database)

Results are shown in **Table 1**. All metrics were averaged over 10 different realizations of the random conditions and were calculated with their standard deviations.

Feature transfer models (1)–(4) surpass fine-tuning models (a) and (b) at all classification performances. This suggests that the feature representation as is obtained in natural and textural images is not suitable for DLDs classification. In other words, the

Table 1 Classification performance comparison for test data. ± following each metrics means the standard deviation.

	(1)	(2)	(3)	(4)	(a)	(b)
Transfer	None	single-stage (conventional)		two-stage *(proposed)*	fine-tuning	
Accuracy	0.9392 ± 0.0017	0.9247 ± 0.0030	0.9501 ± 0.0041	0.9537 ± 0.0016	0.7735	0.8263
Precision	0.9206 ± 0.0025	0.9111 ± 0.0052	0.9472 ± 0.0052	0.9547 ± 0.0026	0.7842	0.8345
Recall	0.9269 ± 0.0021	0.9153 ± 0.0040	0.9465 ± 0.0046	0.9523 ± 0.0018	0.7735	0.8263
F1-score	0.9270 ± 0.0021	0.9153 ± 0.0040	0.9465 ± 0.0046	0.9522 ± 0.0019	0.7675	0.8228

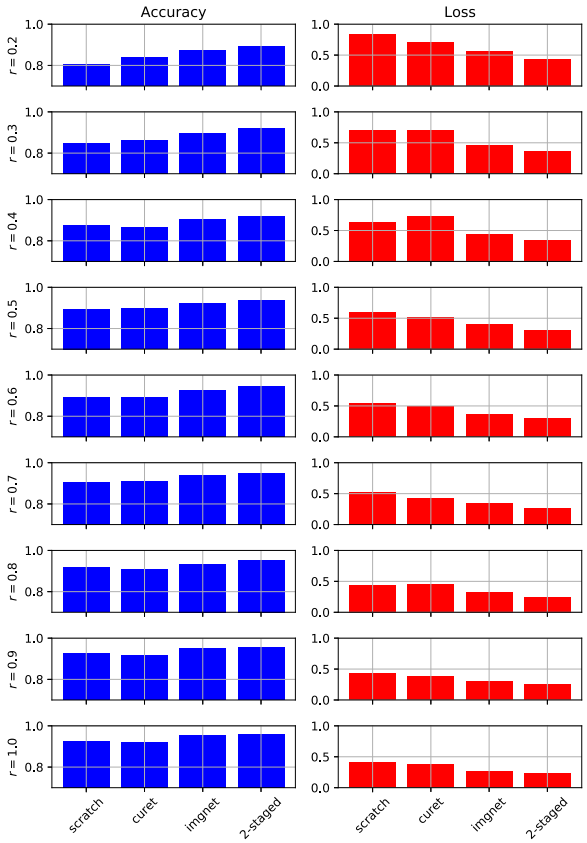

Fig. 6 Performance comparisons of each amount of training data: (Left) classification accuracies of DLDs; (Right) cross-entropy losses of "Fc8" in Fig. 1. Each bar, from left to right, shows the learning processes: (1) learning from scratch; (2) single-staged feature transfer with CUReT; (3) single-staged feature transfer with ImageNet; and (4) our proposed two-stage feature transfer.

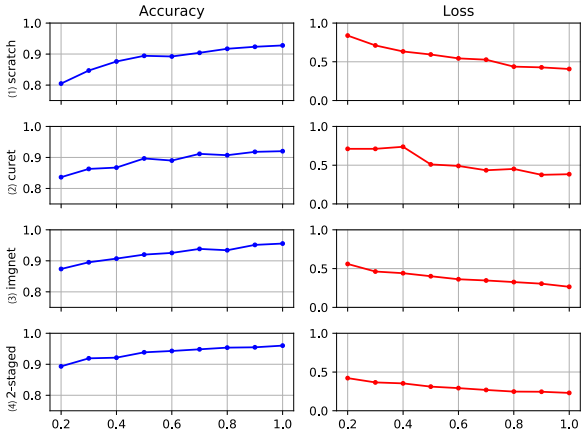

Fig. 7 Fluctuation comparisons of each learning process: (Left) classification accuracies for validation data; (Right) softmax losses of validation data. Each row (1)–(4), from top to bottom, shows the learning processes.

Table 2 Variations of model performances in each process.

	(1)	(2)	(3)	(4)
Slopes of accuracies	1.3560	0.9920	0.9475	0.7479
Slopes of losses	−0.5054	−0.4938	−0.3221	−0.2230

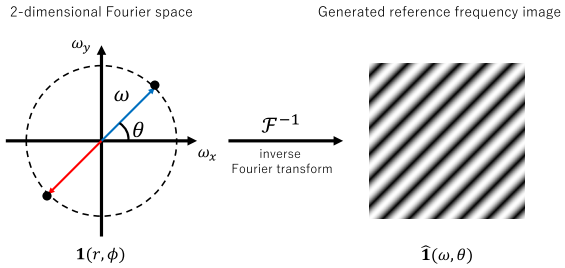

Fig. 8 Mechanism for generating the reference frequency image. In 2-dimensional Fourier space, set the value 1 to two opposite points on the circle of radius ω, which are (ω,θ) and $(\omega,-\theta)$ in polar coordinate. The reference frequency image is given as its inverse Fourier transform. In this example, $\omega = 5$ and $\theta = \pi/4$.

feature extraction part ought to be retrained with the target domain. By comparing each feature transfer model (1)–(4), the two-stage feature transfer (4) displays significantly best performances in all evaluation metrics ($p < 0.01$, non-parametric Wilcoxon signed-rank test, $n = 10$).

On the other hand, despite feature transfer, the single-stage feature transfer model (2) using only CUReT dataset performed worse than learning from scratch (1). This implies that CUReT, by itself, is useless as the source domain for conventional feature transfer. Such deterioration demonstrates that an inappropriate choice of the source domain could results in a worse performance than no feature transfer one. However, as we will discuss later, feature transfer may improve the robustness of the model performance with the size of the training data, even if the choice of the source domain was wrong.

6.2 Model Robustness for the amounts of Training data

In addition to the performance comparison, we demonstrated how the robustness of each model, with respect to the decrease in the amount of target domain data, improved. We transitioned the accuracies and losses of the softmax layer (i.e., "Fc8" in Fig. 1), which represent the classification performance and model fitness for the true predictions, respectively, by changing the amount of DLDs training samples by the ratio, r, from 20% to 100% [*2]. Here note that the amount of source domain's data, i.e., ILSVRC 2012 dataset and CUReT dataset, have not changed.

All samples were evaluated only one time and represented by the 75th percentile of a plateau. **Figure 6** shows the models' performance comparison. In all cases, two-stage feature transfer showed the best robustness for both accuracy and loss, especially

[*2] For example, when $r = 1.0$ and $r = 0.5$, the amounts of training DLDs examples are 927 and 434, respectively. Proportions of each class are retaining.

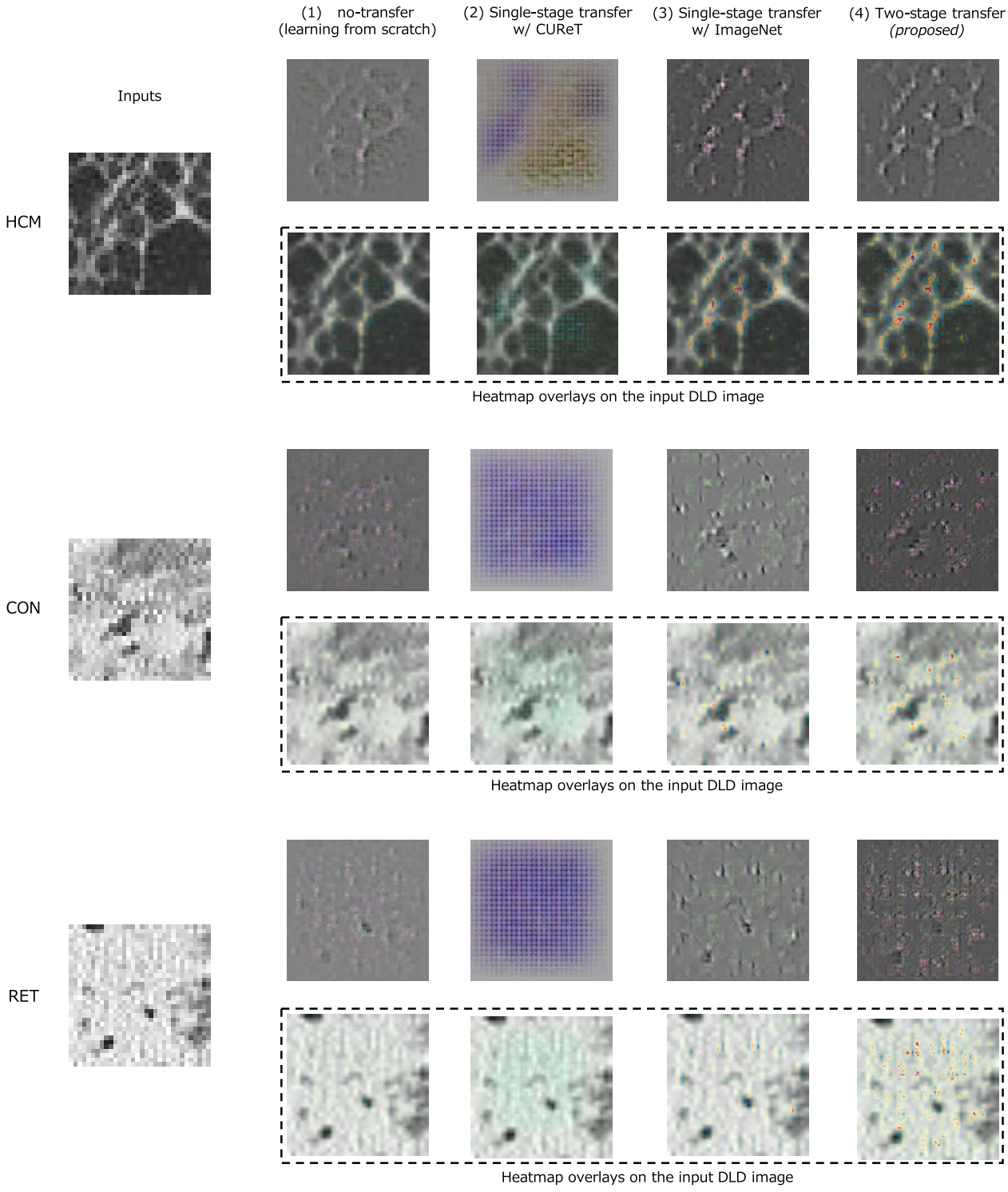

Fig. 9 Visualization results of extracted feature maps from DLDs images. The leftmost figures show the DCNN inputs. Each row represents the input DLDs images, which are in the class HCM, CON, and RET, respectively. Each column represents the DCNN learning processes as described in Section 6. For each visualization result, the above ones show the normalized reconstructed input. Bright regions indicate that the corresponding components of inputs have a strong effect on feature maps. The below ones, surrounded by a dotted line, show the saliency heatmap overlay on the input DLD images.

in the case of a small training dataset.

Figure 7 shows the fluctuation of model performance with a decline in the amount of DLDs images. To quantify the degree of model robustness, we assumed that these variations have linearity to the amount of data, and compared the slopes A of the linear regression model: Accuracy $= Ar + b$, where r is the percentage of data, and b is the intercept coefficient. Clearly, a small absolute value of slope indicates that the model is more robust with r. All feature transfer models show better results than learning from scratch, as shown in **Table 2**. Two-stage feature transfer showed

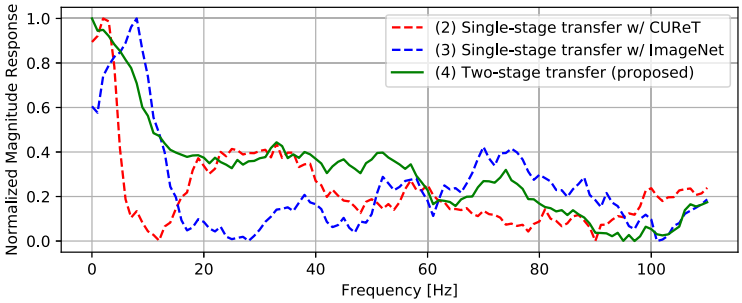

Fig. 10 The frequency response of each feature transfer models. Red-dotted, blue-dotted and green-solid lines represent feature transfer models, as described in Section 6, (2), (3) and (4), respectively. The two-stage feature transfer model (4) have peaks that appeared in both models (2) and (3).

the best robustness, both with accuracy and with loss.

7. Analysis of the Feature Extraction

7.1 Feature Visualization

The main question we address in this section is how feature transfer improves the feature extraction part of DCNNs. First, we analyzed the feature maps extracted in the DCNN using the visualization method, that is DeSaliNet explained in Section 4.2. **Figure 9** shows the visualization results of extracted features (i.e., the input of feature extraction part obtained as an input for layer "Fc6" in Fig. 1) for each model, from (1) to (4). To visualize raw reconstucted input, each activation was normalized into closed interval $[0, 255]$ by minimum and maximum value.

Model (1), learned from scratch, did not show salient activities in any region of the input, as seen in the heatmap overlays. This suggests that the model could not extract meaningful features from the inputs because of the lack of training data. Model (2), transferred from textural images, showed activation in the regions where the textural structure appeared (e.g., in raw visualization results, entire of CON and RET or bottom right of HCM). Alternatively, model (3), transferred from natural images, showed activations in the regions where edge structures appeared (e.g., pits of CON or cyst wall contours of HCM, clearly in the heatmap). It is intuitive, considering that the models trained for natural images show an activation for edge structures (e.g., the object contours and lines), as reported by most studies on the visualization of DCNNs [15], [21], [24]. Interestingly, Model (4), which came from two-stage feature transfer, responded to both edge and textural structures. The models (2) and (3) show the strong responses to the edge and textural regions respectively. In contrast, we can see that these models show weak responses to the opposite regions. Given the results of (2) and (3), such feature representations seem to be additively obtained from both natural images and textural domains during two-stage feature transfer. Performance improvements occur because the DCNNs obtain better feature representation, which suits textural patterns with the two-stage feature transfer.

7.2 Numerical Evaluation

To support our analysis, we have compared the frequency response of each feature transfer model, because textural and shape features can be separated in Fourier space [12]. We denote by $\mathbf{1}(\omega, \theta)$ the reference frequency image which only has a spatial frequency component of ω and a phase component of θ. The reference frequency image $\mathbf{1}(\omega, \theta)$ is given by inverse Fourier transform as

$$\mathbf{1}(\omega, \theta) = \mathcal{F}^{-1}\left[\hat{\mathbf{1}}(r, \phi)\right] \Big/ \left\|\mathcal{F}^{-1}\left[\hat{\mathbf{1}}(r, \phi)\right]\right\|_2^2 \tag{6}$$

where $\hat{\mathbf{1}}(r, \phi)$ is a polar image in Fourier space

$$\hat{\mathbf{1}}(r, \phi) = \begin{cases} 1 & (r = \omega, \phi = \theta) \\ 0 & (\text{otherwise}) \end{cases} \tag{7}$$

$\|\cdot\|_2^2$ denotes the Frobenius norm of image and $\mathcal{F}[\cdot]$ denotes the two-dimensional Fourier transformation. **Figure 8** illustrates the process of generating reference frequency images.

We define the frequency response of the extracted feature by a Frobenius norm gain. Let i be a layer, where the extracted feature appears, the frequency response can be denoted by the notations in Eqs. (4), (5), and (6) as:

$$G(\omega) = \sum_{\theta \in [0, \pi)} \left\| \left(\psi^{(i)\dagger} \circ \psi^{(i)}\right)(\mathbf{1}(\omega, \theta)) \right\|_2^2 \tag{8}$$

Equation (8) represents how salient the fixed-norm input $\mathbf{1}(\omega, \gamma)$ in the feature extraction layer, thus this metric may be suitable to evaluate the frequency responses.

Figure 10 shows the frequency renponse of each feature transfer model (2)–(4). To emphasize the peak structures, each response was normalized into an interval $[0, 1]$ and was smoothed by a second-order Savitzky-Golay filter. Model (2), transferred from textural images, has peak responses at low-frequencies near DC ($\omega \simeq 0$ Hz) and mid-frequencies ($\omega = 20$–40 Hz), which are essential for textural images [12]. Model (3), transferred from natural images, has strong peak responses at low-frequencies near 10 Hz and mid-high-frequencies ($\omega \simeq 70$ Hz). Similarly to Section 7.1, model (4), transferred from both textural and natural images, has a peak response at low-frequencies near DC to mid-frequencies like model (2); however, it also has a peak at mid-high-frequencies near 70 Hz, similar to model (3). This result accords with the visualization result that the two-stage feature transfer model can additively obtain both textural and structural feature representation from multiple source domains.

8. Conclusion

We proposed a two-stage feature transfer, which improved the

performance of DCNNs for classification tasks of textural images, as an extension of conventional transfer learning methods, which use a single domain as the source. We applied two-stage feature transfer to the classification of HRCT images of lung diseases and demonstrated that two-stage feature transfer improves classification performance and robustness while decreasing the amount of training data, compared to learning from scratch and conventional transfer learning. To assess these improvements, we analyzed and compared each feature representation using a feature visualization method. Two-stage feature transfer seems to have provided appropriate feature representations for both edge and textural structures transferred from natural images and textural images, respectively. These results indicate the consequence of source domain selection.

Acknowledgments This work is partially supported by Grant-in-Aids for Scientific Research KAKENHI (C) 16K00328, and Innovative Areas 16H01452, MEXT, Japan. We thank Dr. H. Nosato (AIRC, AIST) for constructive discussion for our numerical evaluation ways. We also appreciate Prof. Honda and Osaka University Hospital for providing the HRCT images of DLDs.

References

[1] Azizpour, H., Razavian, A.S., Sullivan, J., Maki, A. and Carlsson, S.: Factors of transferability for a generic convnet representation, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.38, No.9, pp.1790–1802 (2016).

[2] Ba, J. and Caruana, R.: Do deep nets really need to be deep?, *Advances in neural information processing systems*, pp.2654–2662 (2014).

[3] Christodoulidis, S., Anthimopoulos, M., Ebner, L., Christe, A. and Mougiakakou, S.: Multisource Transfer Learning With Convolutional Neural Networks for Lung Pattern Analysis, *IEEE Journal of Biomedical and Health Informatics*, Vol.21, No.1, pp.76–84 (online), DOI: 10.1109/JBHI.2016.2636929 (2017).

[4] Dana, K.J., Van Ginneken, B., Nayar, S.K. and Koenderink, J.J.: Reflectance and texture of real-world surfaces, *ACM Transactions on Graphics* (*TOG*), Vol.18, No.1, pp.1–34 (1999).

[5] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. and Fei-Fei, L.: Imagenet: A large-scale hierarchical image database, *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009*, pp.248–255, IEEE (2009).

[6] Fukushima, K.: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, *Biological Cybernetics*, Vol.36, No.4, pp.193–202 (1980).

[7] Fukushima, K.: Neural network model for selective attention in visual pattern recognition and associative recall, *Applied Optics*, Vol.26, No.23, pp.4985–4992 (1987).

[8] Fukushima, K.: Neocognitron: A hierarchical neural network capable of visual pattern recognition, *Neural networks*, Vol.1, No.2, pp.119–130 (1988).

[9] Gao, M., Bagci, U., Lu, L., Wu, A., Buty, M., Shin, H.-C., Roth, H., Papadakis, G.Z., Depeursinge, A., Summers, R.M., et al.: Holistic classification of CT attenuation patterns for interstitial lung diseases via deep convolutional neural networks, *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, pp.1–6 (2016).

[10] Han, S., Pool, J., Tran, J. and Dally, W.: Learning both weights and connections for efficient neural network, *Advances in Neural Information Processing Systems*, pp.1135–1143 (2015).

[11] Hubel, D.H. and Wiesel, T.N.: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex, *The Journal of physiology*, Vol.160, No.1, pp.106–154 (1962).

[12] Julesz, B. and Caelli, T.: On the limits of Fourier decompositions in visual texture perception, *Perception*, Vol.8, No.1, pp.69–73 (1979).

[13] Kornblith, S., Shlens, J. and Le, Q.V.: Do Better ImageNet Models Transfer Better?, arXiv preprint arXiv:1805.08974 (2018).

[14] Krizhevsky, A., Sutskever, I. and Hinton, G.E.: Imagenet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems*, pp.1097–1105 (2012).

[15] Mahendran, A. and Vedaldi, A.: Salient deconvolutional networks, *European Conference on Computer Vision*, pp.120–135, Springer (2016).

[16] Pan, S.J. and Yang, Q.: A survey on transfer learning, *IEEE Trans. Knowledge and Data Engineering*, Vol.22, No.10, pp.1345–1359 (2010).

[17] Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D.: Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization, *ICCV*, pp.618–626 (2017).

[18] Shin, H.-C., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D. and Summers, R.M.: Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning, *IEEE Trans. Medical Imaging*, Vol.35, No.5, pp.1285–1298 (2016).

[19] Shouno, H., Suzuki, S. and Kido, S.: A transfer learning method with deep convolutional neural network for diffuse lung disease classification, *International Conference on Neural Information Processing*, pp.199–207 (2015).

[20] Shuono, H. and Fukushima, K.: Connected character recognition in cursive handwriting using selective attention model with bend processing, *Systems and Computers in Japan*, Vol.26, No.10, pp.35–46 (1995).

[21] Simonyan, K., Vedaldi, A. and Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps, arXiv preprint arXiv:1312.6034 (2013).

[22] Tajbakhsh, N., Shin, J.Y., Gurudu, S.R., Hurst, R.T., Kendall, C.B., Gotway, M.B. and Liang, J.: Convolutional neural networks for medical image analysis: Full training or fine tuning?, *IEEE Trans. Medical Imaging*, Vol.35, No.5, pp.1299–1312 (2016).

[23] Uchiyama, Y., Katsuragawa, S., Abe, H., Shiraishi, J., Li, F., Li, Q., Zhang, C.-T., Suzuki, K., et al.: Quantitative computerized analysis of diffuse lung disease in high-resolution computed tomography, *Medical Physics*, Vol.30, No.9, pp.2440–2454 (2003).

[24] Zeiler, M.D. and Fergus, R.: Visualizing and understanding convolutional networks, *European Conference on Computer Vision*, pp.818–833, Springer (2014).

Appendix

A.1 DeSaliNet's Inverse Maps

This appendix provides details of the inverse maps used in DeSaliNet [15]. In Eq. (5), each $\phi_i^{(L_i)\dagger}$ denotes the inverse map of each forward operation, $\phi_i^{(L_i)}$, where the L_i is a layer type of the i-th stage. DeSaliNet considers only the case where $L_i \in$ {convolution, max-pooling, ReLU}, otherwise the layers be ignored. This results in an identity map.

Convolution layer

Let $h_i(l, \mathbf{x})$ be a feature map of the l-th channel, where it is in the position, $\mathbf{x}$. The inverse map of the convolution layer $\phi_i^{\text{conv}\dagger}$, called "deconvolution," is denoted as

$$\phi^{\text{conv}\dagger}\left(h_i(l,\mathbf{x})\right) = \sum_{l,\mathbf{u}} g_i(k,i,S(\mathbf{u}))\, h_i(l,\mathbf{x}-\mathbf{u}), \qquad \text{(A.1)}$$

where

$$\begin{array}{ccccc} S: & \mathbb{Z}^2 & \longrightarrow & \mathbb{Z}^2 \\ & \cup & & \cup \\ & \begin{pmatrix} x \\ y \end{pmatrix} & \longmapsto & \begin{pmatrix} y \\ x \end{pmatrix}. \end{array} \qquad \text{(A.2)}$$

Equations (A.1) and (A.2) indicate that the deconvolution layer is a convolution for feature maps having a transposed filter tensor, $\mathbf{g}_i$.

Max-pooling layer

The inverse map of the max-pooling layer, $\phi_i^{\text{MP}\dagger}$, is denoted as

$$\phi^{\text{MP}\dagger}(h_i(l,\mathbf{x})) = \begin{cases} h_i(l,\mathbf{x}) & (x \in \psi_i^{\text{MP}}) \\ 0 & \text{otherwise} \end{cases}, \qquad \text{(A.3)}$$

where ψ_i^{MP} contains the stored maximum value locations of forward calculation in max-pooling. The pooled map is sparsely restored into a maximum position.

Rectifying layer

The inverse map of the ReLU layer, $\phi_i^{\mathrm{ReLU}\dagger}$, is denoted as

$$\phi^{\mathrm{ReLU}\dagger}(h_i(l,\mathbf{x})) = \begin{cases} h_i(l,\mathbf{x}) & (x \in \psi_i^{\mathrm{LU}}) \\ 0 & \text{otherwise,} \end{cases} \tag{A.4}$$

where ψ_i^{LU} is the stored positive locations in the forward ReLU caculation, where zero was not modulated.

Aiga Suzuki received his B.Eng. degree from the University of Electro-Communications, Japan, in 2017. He is currently working towards his M.Eng. degree at the University of Tsukuba and Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology (AIRC, AIST). His research interests include artificial neural networks and computer visions. He is a student member of IPSJ, IEICE and IEEE.

Hidenori Sakanashi received his Ph.D. in information engineering from Hokkaido University, in 1996. He currently leads Artificial Intelligence Application Research Team, Artificial Intelligence Research Center (AIRC), National Institute of Advanced Industrial Science and Technology (AIST). He is also an associate professor (cooperative graduate school program) at the University of Tsukuba and is a visiting professor at Toho University. His research interests include pattern recognition, machine learning and computer-aided diagnosis. He is a member of IPSJ.

Shoji Kido received his M.D. degree from Osaka University in 1988. He received his Ph.D. degrees in Medicine and Information Science from Osaka University in 1992 and 1999, respectively. He was a Professor in the Department of Computer Science and Systems Engineering at Yamaguchi University from 1999 to 2004, and in the Department of Applied Medical Engineering Science, Graduate School of Medicine, Yamaguchi University from 2004 to 2016. He is currently a Professor in the Graduate School of Sciences and Technology for Innovation, Yamaguchi University.

Hayaru Shouno was born in 1968. He received M.E. and Ph.D. from Osaka University in 1994 and 1999. He is a professor at the University of Electro-Communications. His current research interest is in the deep neural network and its applications, e.g. the medical image processing. He is a member of IEEE, IEICE, JNNS, and IPSJ.