arXiv:1804.03830v1 [cs.CV] 11 Apr 2018

Unsupervised Segmentation of 3D Medical Images Based on Clustering and Deep Representation Learning

Takayasu Moriya[a], Holger R. Roth[a], Shota Nakamura[b], Hirohisa Oda[c], Kai Nagara[c], Masahiro Oda[a], and Kensaku Mori[a]

[a]Graduate School of Informatics, Nagoya University
[b]Nagoya University Graduate School of Medicine
[c]Graduate School of Information Science, Nagoya University

ABSTRACT

This paper presents a novel unsupervised segmentation method for 3D medical images. Convolutional neural networks (CNNs) have brought significant advances in image segmentation. However, most of the recent methods rely on supervised learning, which requires large amounts of manually annotated data. Thus, it is challenging for these methods to cope with the growing amount of medical images. This paper proposes a unified approach to unsupervised deep representation learning and clustering for segmentation. Our proposed method consists of two phases. In the first phase, we learn deep feature representations of training patches from a target image using joint unsupervised learning (JULE) that alternately clusters representations generated by a CNN and updates the CNN parameters using cluster labels as supervisory signals. We extend JULE to 3D medical images by utilizing 3D convolutions throughout the CNN architecture. In the second phase, we apply k-means to the deep representations from the trained CNN and then project cluster labels to the target image in order to obtain the fully segmented image. We evaluated our methods on three images of lung cancer specimens scanned with micro-computed tomography (micro-CT). The automatic segmentation of pathological regions in micro-CT could further contribute to the pathological examination process. Hence, we aim to automatically divide each image into the regions of invasive carcinoma, noninvasive carcinoma, and normal tissue. Our experiments show the potential abilities of unsupervised deep representation learning for medical image segmentation.

Keywords: Segmentation, Micro-CT, Representation Learning, Unsupervised Learning, Deep Learning

1. PURPOSE

The purpose of our study is to develop an unsupervised segmentation method of 3D medical images. Most of the recent segmentation methods using convolutional neural networks (CNNs) rely on supervised learning that requires large amounts of manually annotated data.[1] Therefore, it is challenging for these methods to cope with medical images due to the difficulty of obtaining manual annotations. Thus, research into unsupervised learning, especially for 3D medical images, is very promising. Many previous unsupervised segmentation methods for 3D medical images are based on clustering.[2] However, most unsupervised work in medical imaging was limited to hand-crafted features that were then used with traditional clustering methods to provide segmentation.

In our study, we investigated whether representations learned by unsupervised deep learning aid in the clustering and segmentation of 3D medical images. As an unsupervised deep representation learning, we adopt joint unsupervised learning (JULE)[3] based on a framework that progressively clusters images and learns deep representations via a CNN. Our main contribution is to combine JULE with k-means[4] for medical image segmentation. To our knowledge, our methods are the first to employ JULE for unsupervised medical image segmentation. Moreover, our work is the first to conduct automatic segmentation for pathological diagnosis of micro-CT images. This work demonstrates that deep representations can be useful for unsupervised medical image segmentation.

There are two reasons why we chose JULE for our proposed method. The first reason is that JULE is robust against data variation (e. g., image type, image size, and sample size) and thus can cope with a dataset composed of 3D patches cropped out of medical images. Moreover, the range of intensities is different for each medical image. Thus, we need a learning method that works well with various datasets. The second reason is that JULE can learn representations that work well with many clustering algorithms. This advantage allows us to learn representations on a subset of possible patches from a target image and then apply a faster clustering algorithm to the representations of all patches for segmentation.

2. METHOD

The proposed segmentation method has two phases: (1) learning feature representations using JULE and (2) clustering deep representations for segmentation. In phase (1), we conduct JULE in order to learn the representations of image patches randomly extracted from an unlabeled image. For use with 3D medical images, we extend JULE to use 3D convolutions. The purpose of this phase is to obtain a trained CNN that can transform image patches to discriminative feature representations. In phase (2), we use k-means to assign labels to learned representations generated by the trained CNN.

2.1 Deep Representation Learning

The main idea behind JULE is that meaningful cluster labels could become supervisory signals for representation learning and discriminative representations help to obtain meaningful clusters. Given a set of n_s unlabeled image patches $\boldsymbol{I} = \{I_1, \ldots, I_{n_s}\}$, cluster labels for all image patches $\boldsymbol{y} = \{y_1, \ldots, y_{n_s}\}$, and the parameters for representations $\boldsymbol{\theta}$, the objective function of JULE is formulated as

$$(\hat{\boldsymbol{y}}, \hat{\boldsymbol{\theta}}) = \underset{\boldsymbol{y},\boldsymbol{\theta}}{\arg\min}\, \mathcal{L}(\boldsymbol{y}, \boldsymbol{\theta} | \boldsymbol{I}) \tag{1}$$

where $\mathcal{L}$ is a loss function. JULE tries to find optimal $\hat{\boldsymbol{y}}$ in the forward pass and optimal $\hat{\boldsymbol{\theta}}$ in the backward pass to minimize $\mathcal{L}$. By iterating the forward pass and the backward pass, we can obtain more discriminative representations and therefore better image clusters. In the forward pass, we conduct image clustering to merge clusters using agglomerative clustering.[5] In the backward pass, we conduct representation learning via a 3D CNN using cluster labels as supervisory signals. JULE can be interpreted as a recurrent framework because it iterates merging clusters and learning representations over multiple timesteps until it obtains the desired number of clusters C. Fig. 1 shows an overview of a recurrent process at the time of step t.

2.2 Extension to 3D Medical Images

We conducted two extensions of JULE. One is the extension of the recurrent process for the CNN training in the backward pass. Originally, JULE aims to obtain the final clusters and finishes when it obtains a desired number of clusters in the forward pass.[3] In contrast, our purpose is to obtain a well-trained CNN. If we terminate the recurrent process in the final forward pass, we lose a chance to train the CNN with the final cluster labels. Therefore, we extended the recurrent process to train the CNN using the final cluster label in the backward pass. The intuitive reason is that the final clusters are the most precise of the entire process and representations learned with them become more discriminative. The other is the extension of CNN to support 3D medical images. Originally, JULE is a representation learning and clustering method for 2D images. We, however, aim to learn representations using 3D image patches. Thus, we extended the CNN architecture of the original JULE[3] to use 3D convolutions throughout the network.

2.3 Patch Extraction

Prior to learning representations, we need to prepare training data composed of small 3D image patches. These patches are extracted from the unlabeled image, which is our target for segmentation, by randomly cropping n_s sub-volumes of $w \times w \times w$ voxels. In many cases of medical image segmentation, we need to exclude the outside of a scanned object from the training data. We choose a certain threshold that can divide the scanned target region from the background and include only patches whose center voxel intensity is within the threshold. After extracting training patches, we centralize them by subtracting out the mean of all intensities and dividing by the standard deviation, following Yang et al.[3]

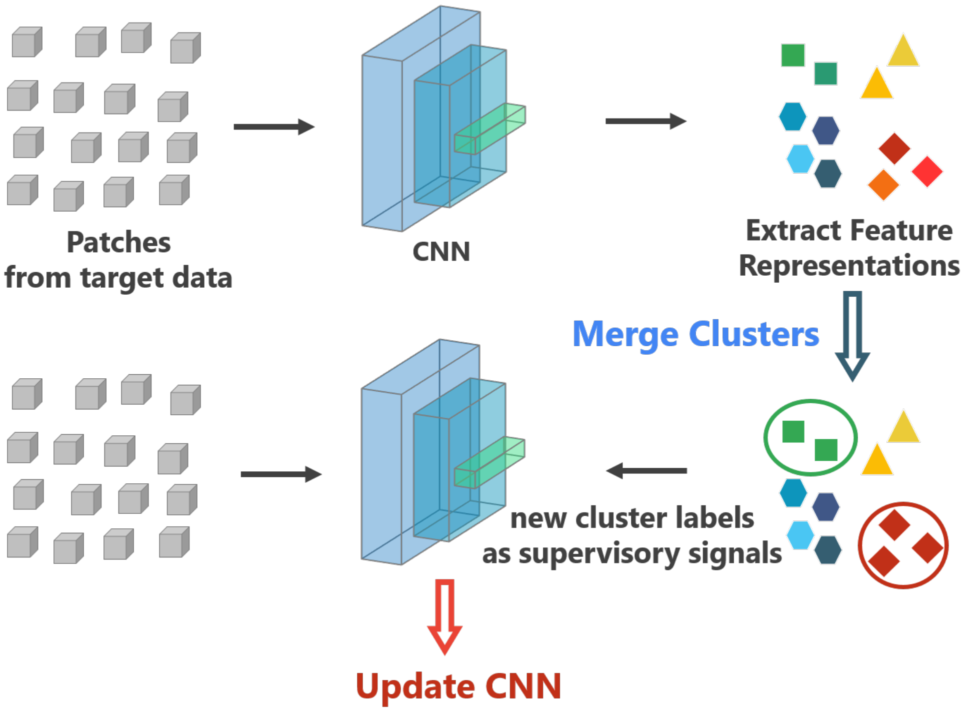


Figure 1: Illustration of a recurrent process at the time of step t. First, we extract representations $\boldsymbol{X}^t$ from training patches $\boldsymbol{I}$ via a CNN with parameter $\boldsymbol{\theta}^t$. Next, we merge them and assign new labels $\boldsymbol{y}^t$ to $\boldsymbol{X}^t$. Finally, we input $\boldsymbol{I}$ into the CNN again and update the CNN parameters $\boldsymbol{\theta}^t$ to $\boldsymbol{\theta}^{t+1}$ through back propagation from a loss calculated using $\boldsymbol{y}^t$ as supervisory signals. Note that the CNN is initialized with random weights.

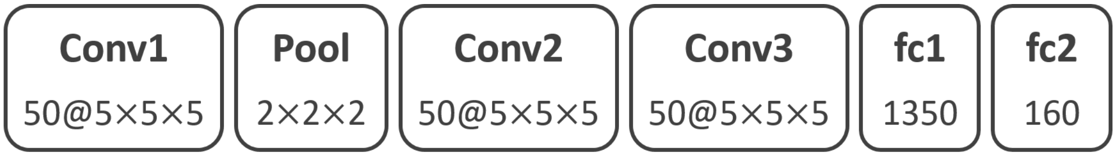


Figure 2: Our CNN architecture has 3 convolutional, 1 max pooling, and 2 fully-connected layers. All 3D convolutional kernels are $5 \times 5 \times 5$ with stride 1. Number of kernels are denoted in each box. Pooling kernels are $2 \times 2 \times 2$ with stride 2. The first fully-connected layer has 1350 neurons, and the second one has 160 neurons.

2.4 CNN Architecture

Our CNN consists of three convolutional layers, one max pooling layer, and two fully-connected layers. The kernels of the second and third convolutional layers are connected to all kernel maps in the previous layer. The neurons in the fully-connected layers are connected to all neurons in the previous layer. The max pooling layer follows the first convolutional layers. Batch normalization is applied to the output of each convolutional layer. A rectified linear unit (ReLU) is used as the nonlinearity after batch normalization. The second fully-connected layer is followed by the L2-normalization layer. All of the convolutional layers use 50 kernels of $5 \times 5 \times 5$ voxels with a stride of 1 voxel. The Max pooling layer has a kernel of $2 \times 2 \times 2$ voxels with a stride of 2 voxels. The input to the network are image patches of $27 \times 27 \times 27$ voxels. The first fully-connected layer has 1350 neurons and the second has 160 neurons. Other parameters for the CNN training, such as learning rate, are the same as proposed in the original JULE.[3] The CNN architecture is presented in Fig. 2.

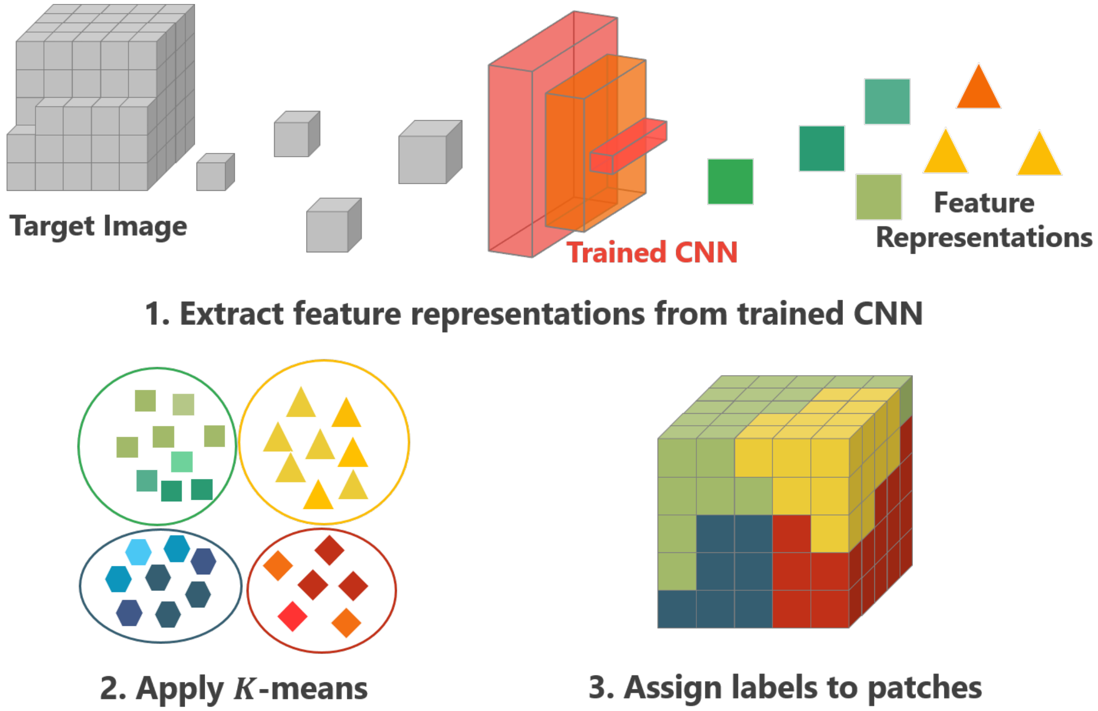

Target Image
Trained CNN
Feature Representations
1. Extract feature representations from trained CNN
2. Apply K-means
3. Assign labels to patches

Figure 3: Our segmentation process. We first obtain feature representations from a trained CNN and then apply conventional k-means to them. Finally, we assign labels to the patches based on the clustering results. (For simplification, we have drawn the figure with a stride equal to w.)

2.5 Segmentation

In the segmentation phase, we first extract a possible number of patches of $w \times w \times w$ voxels from the target image separated by s voxels each. Note that stride s is not larger than w. As with extracting training patches, we select only voxels within the scanned sample by thresholding. The trained CNN transforms each patch into a feature representation. We then divide the feature representations into K clusters by k-means. After applying k-means, each representation is assigned a label $l(1 \leq l \leq K)$ and we need to project these labels onto the original image. We consider subpatches of $s \times s \times s$ voxels centered in each extracted patch. Each subpatch is assigned the same label as the closest cluster representation using Euclidean distance. This segmentation process is illustrated in Fig. 3.

3. EXPERIMENTS AND RESULTS

3.1 Datasets

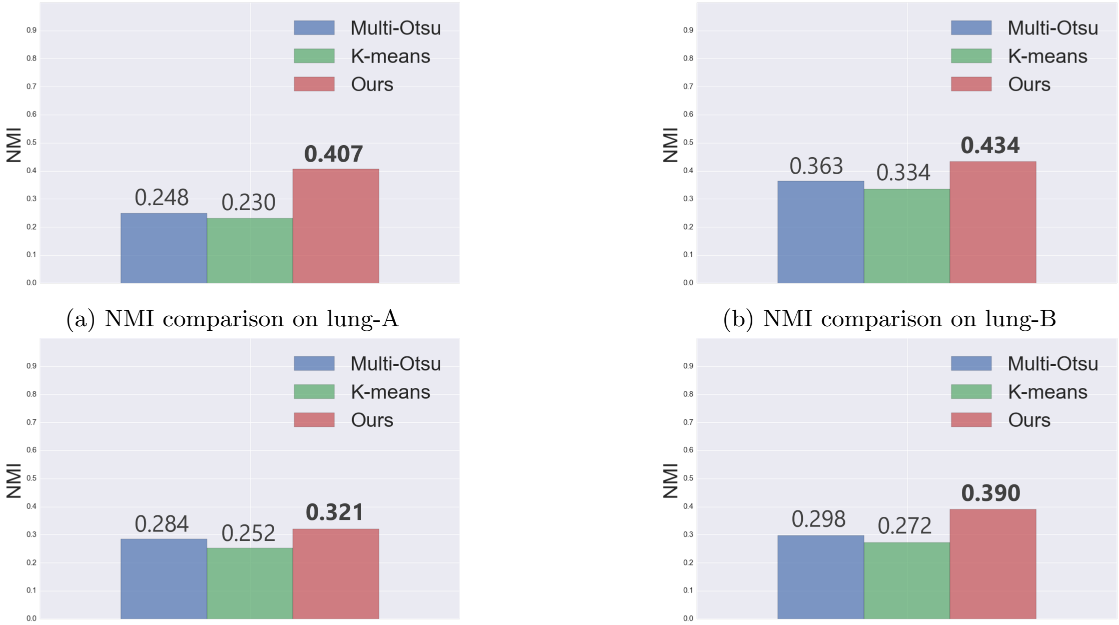
We chose three lung cancer specimen images scanned with a micro-CT scanner (inspeXio SMX-90CT Plus, Shimadzu Corporation, Kyoto, Japan) to evaluate our proposed method. The lung cancer specimens from the respective patients were scanned with similar resolutions. We aimed to divide each image into three histopathological regions: (a) invasive carcinoma; (b) noninvasive carcinoma; and (c) normal tissue. We selected these images because segmenting the regions on micro-CT images based on histopathological features could contribute to the pathological examination.[6,7] Detailed information for each image is shown in Table 1.

3.2 Parameter Settings

For JULE, we randomly extracted 10,000 patches of $27 \times 27 \times 27$ voxels from a target image. We set the number of final clusters C to 100 for lung-A and lung-C, to 10 for lung-B, which are the stopping conditions of agglomerative clustering. Other parameters are the same as in the original JULE.[3] After representation learning, we extracted

Table 1: Images used in our experiments

Image	Image Size (voxel)	Resolution (μm)	Threshold (intensity)
lung-A	756×520×545	27.1×27.1×27.1	4000
lung-B	594×602×624	29.63×29.63×29.63	2820
lung-C	477×454×971	29.51×29.51×29.51	4700



(a) NMI comparison on lung-A

(b) NMI comparison on lung-B

(c) NMI comparison on lung-C

(d) Average NMI comparison

Figure 4: NMI comparison on three datasets. Our method outperformed traditional unsupervised methods.

patches of $27 \times 27 \times 27$ voxels with a stride of five voxels and processed them by the trained CNN to obtain a 160 dimensional representation for each patch. For segmentation, we applied the conventional k-means to the feature representations, setting K to 3.

3.3 Evaluations

We used normalized mutual information (NMI)[8] to measure segmentation accuracy. A larger NMI value means more precise segmentation results. We used seven manually annotated slices for evaluation. We compared the proposed method with traditional k-means segmentation and multithreshold Otsu method.[9] We also evaluated the average NMI of each method across the datasets. The results are shown in Fig. 4. In each figure, the best performance NMI for each K is in bold. As shown in all of the figures, JULE-based segmentation outperformed traditional unsupervised methods. While the NMI scores of our methods are not high, qualitative evaluation shows promising results of our proposed method (see Fig. 5). The qualitative examples show that JULE-based segmentation divided normal tissue region from the cancer region, including invasive carcinoma and noninvasive carcinoma, well.

4. DISCUSSIONS

Qualitative evaluations demonstrate that JULE can learn features that divide higher intensity regions from lower intensity regions. This is because, generally, regions of invasive and noninvasive carcinoma have substantially high intensities, whereas normal tissue regions have low intensities. Moreover, for lung-A and lung-B, JULE divided invasive carcinoma from noninvasive carcinoma. This results shows the potential ability to learn features that reflect variation in intensity. This is because, seemingly, invasive carcinoma and normal tissue typically have a small variation of intensities, whereas noninvasive carcinoma has a large variation of intensities.

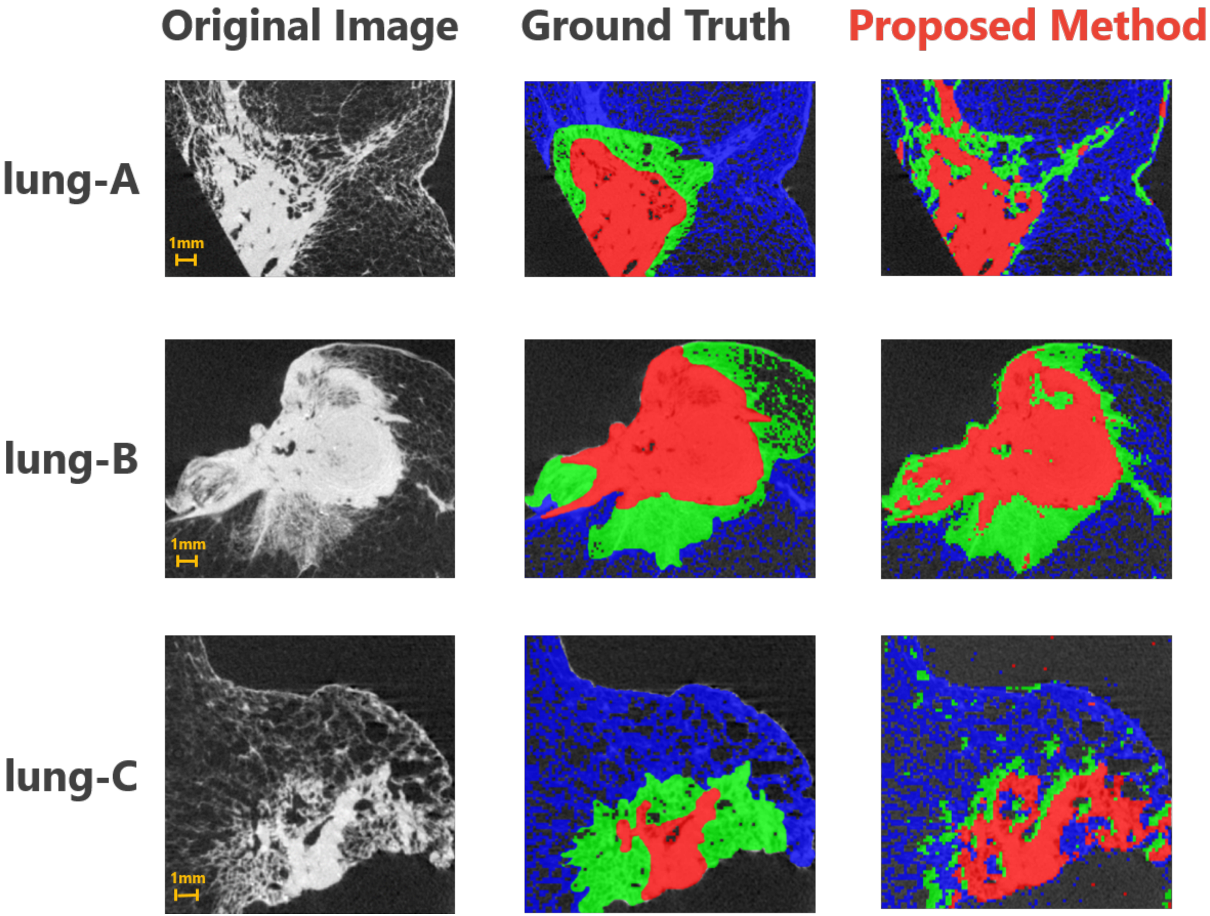


Figure 5: Segmentation results of lung-A (top line), lung-B (middle line), and lung-C (bottom line). In the ground truth, the red, green, and blue regions correspond to the region of invasive carcinoma, noninvasive carcinoma, and normal tissue, respectively. In the results of the JULE-based segmentation, colors indicate the same cluster, but are assigned at random.

5. CONCLUSION

We proposed an unsupervised segmentation using JULE that alternately learns deep representations and image clusters. We demonstrated the potential abilities of unsupervised medical image segmentation using deep representations. Our segmentation method could be applicable to many other applications in medical imaging.

ACKNOWLEDGMENTS

This research was supported by the Kakenhi by MEXT and JSPS (26108006, 17K20099) and the JSPS Bilateral International Collaboration Grants.

REFERENCES

[1] Long, J., Shelhamer, E., and Darrell, T., “Fully convolutional networks for semantic segmentation,” in [*IEEE CVPR*], 3431–3440 (2015).

[2] García-Lorenzo, D., Francis, S., Narayanan, S., Arnold, D. L., and Collins, D. L., “Review of automatic segmentation methods of multiple sclerosis white matter lesions on conventional magnetic resonance imaging,” *Medical Image Analysis* **17**, 1–18 (2013).

[3] Yang, J., Parikh, D., and Batra, D., "Joint unsupervised learning of deep representations and image clusters," in [*IEEE CVPR*], 5147–5156 (2016).

[4] MacQueen, J. et al., "Some methods for classification and analysis of multivariate observations," in [*Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*], **1**, 281–297 (1967).

[5] Zhang, W., Wang, X., Zhao, D., and Tang, X., "Graph degree linkage: Agglomerative clustering on a directed graph," in [*ECCV*], **7572**, 428–441 (2012).

[6] Mori, K., "From macro-scale to micro-scale computational anatomy: a perspective on the next 20 years," *Medical Image Analysis* **33**, 159–164 (2016).

[7] Nakamura, S., Mori, K., Okasaka, T., Kawaguchi, K., Fukui, T., Fukumoto, K., and Yokoi, K., "Micro-computed tomography of the lung: Imaging of alveolar duct and alveolus in human lung," in [*D55. LAB METHODOLOGY AND BIOENGINEERING: JUST DO IT*], A7411–A7411, American Thoracic Society (2016).

[8] Strehl, A. and Ghosh, J., "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *Journal of machine learning research* **3**(Dec), 583–617 (2002).

[9] Otsu, N., "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics* **9**(1), 62–66 (1979).