# End-to-End Deep Learning for Reconstructing Segmented 3D CT Image from Multi-Energy X-ray Projections

Siqi Wang
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
siqi@den.t.u-tokyo.ac.jp

Tatsuya Yatagawa
Hitotsubashi University
2-1 Naka, Kunitachi-shi, Tokyo, Japan
tatsuya.yatagawa@r.hit-u.ac.jp

Yutaka Ohtake
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
ohtake@den.t.u-tokyo.ac.jp

Toru Aoki
Shizuoka University
3-5-1 Johoku, Naka-ku, Hamamatsu, Shizuoka, Japan
aoki.toru@shizuoka.ac.jp

Jun Hotta
Zodiac Co., Ltd.
145-1 Tokiwacho, Naka-ku, Hamamatsu, Shizuoka, Japan
junhotta@zodiacz.com

## Abstract

*This paper presents an end-to-end deep-learning-based (DL-based) segmentation technique for multi-energy sparse-view CT, where a single network achieves local CT reconstruction and segmentation. While recent DL-based CT segmentation outperformed traditional methods in terms of accuracy and automation, these methods input a "reconstructed" CT. Thus, its performance highly depends on the CT image quality. The reliance prohibits the application of these techniques for sparse-view CT, whereas the sparse-view CT is another important technique to reduce radiation dose and image acquisition time. Our end-to-end deep learning technique integrates the reconstruction and segmentation within a single neural network, which allows us to improve the segmentation quality for sparse-view CT data. The proposed method extracts fragments of pixels from each multi-energy projection corresponding to a bar of CT image voxels. In this way, our network, comprising "filtering", "back-projection," and "segmentation" sub-networks, directly obtains the segmented CT image directly from projections. Our CT segmentation on a bar-by-bar basis is significantly memory-efficient due to the independence of memory-expensive 3D convolution. Consequently, our method delivers high-quality segmentation, where the problems of sparse-view artifacts and memory-expensiveness of prior methods are resolved.*

## 1. Introduction

X-ray computed tomography (CT) is a widely-used imaging technique that provides cross-sectional images of objects and finds extensive applications in medical diagnosis. Concerning this, sparse-view CT is an important technique to reduce radiation dose and shorten the image acquisition time. Therefore, the reconstruction algorithms from sparse-view projections have been intensely studied for medial imaging and computer vision fields, including traditional optimization-based iterative reconstruction methods [8, 14] and recent deep-learning-based (DL-based) approaches [23, 24].

CT segmentation is another technique that plays a crucial role in extracting certain parts from a CT image. It involves dividing the CT image into multiple regions based on their underlying anatomical or pathological properties. Common segmentation approaches, such as thresholding [9], region-growing [12], and graph-cut-based methods [18], have been widely used in CT segmentation, which is typically applied to CT "reconstructed" images and agnostic to the projection images used to obtain the CT images. Moreover, there has been increasing interest in DL-based CT segmentation methods. DL-based approaches include those based on encoder-decoder networks [16, 21] and generative adversarial networks (GANs) [5, 20]. In contrast to traditional methods of processing CT images, deep-learning methods can be roughly classified into two categories, i.e., projection-domain methods [6, 22] and image-

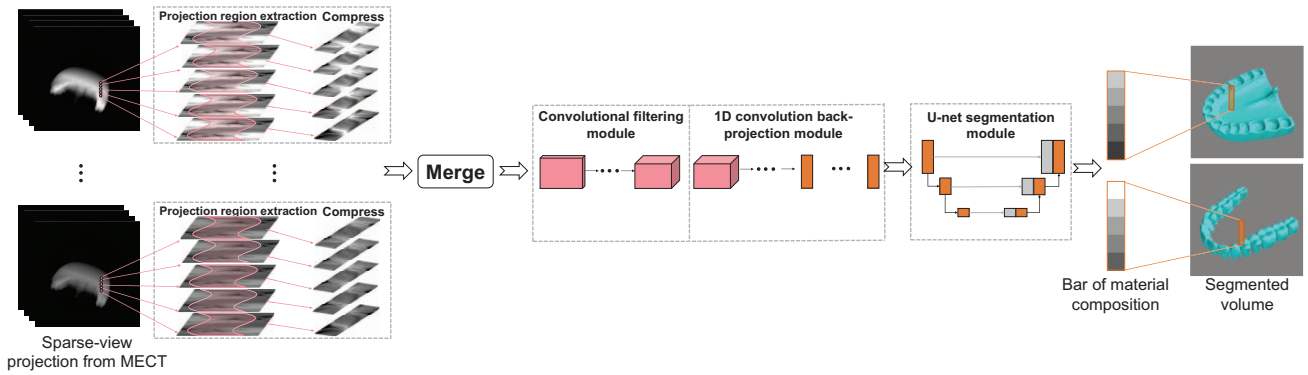


Figure 1. The overview of the proposed method, which obtains a segmented CT image by end-to-end deep learning from given projection images. The network learns the relationship between the material composition of a bar of voxels and the corresponding projection of the bar on the detector plane. To this end, the filtering module and projection module obtain the feature vector for each voxel in the bar. Then, the segmentation module delivers the material composition ratio within each voxel.

domain methods [4, 7, 13, 21]. To obtain a segmented CT image using these techniques, we have to use the projection-domain methods after CT reconstruction. In contrast, we have to use the image-domain methods before the CT reconstruction. Thus, both traditional and DL-based methods require CT reconstruction somewhere in obtaining a segmented CT image.

Unfortunately, the adherence to CT reconstruction poses a problem when using sparse-view CT data, where the number of projections is insufficient to satisfy the Shannon–Nyquist sampling theorem, resulting in the CT image being contaminated by sparse-view artifacts. The sparse-view artifact is a critical issue for medical applications because the sparsity of CT data is essential to reducing radiation dose and shortening the image acquisition process.

On the other hand, multi-energy CT (MECT) is a next-generation technique to enhance the information obtained by X-ray CT imaging [2]. MECT is physically realized by a photon-counting detector (PCD), often made of cadmium telluride (CdTe). Unlike conventional detectors that convert X-ray photons into visible light before generating an electric signal, PCDs directly convert the X-ray photons into an electric signal. The PCD is expected to further reduce radiation dose in medical X-ray CT imaging due to its capability of improving contrast-to-noise ratio and reducing electronic noise. Moreover, MECT offers more accurate material discrimination over conventional single-energy CT (SECT), as it provides additional spectral information that can be used to differentiate materials with similar X-ray attenuation coefficients in specific energy levels. However, the application of MECT for material segmentation is developing, and deep learning techniques for that still need to be explored.

Based on these motivating factors, we propose an end-to-end deep-learning method to obtain a segmented CT image from sparse-view multi-energy projections. The term "end-to-end" specifies that our method directly generates segmentation from sparse-view CT data without relying on traditional CT reconstruction algorithms. In other words, our method directly obtains a segmented CT image from projections using a convolutional neural network (CNN). Furthermore, we leverage MECT to distinguish materials more accurately. The overall process is illustrated in Fig. 1. As shown in the left of Fig. 1, our method reconstructs a segmented CT on a bar-by-bar basis following the recent study for a DL-based construction of Feldkamp–Davis–Kress (FDK) algorithm [19], which the authors refer to as "BBB-FDK." In this approach, the image domain is divided into a set of bars comprised of vertically connected voxels. Such a divide-and-conquer strategy allows us to reduce the memory resource required for the reconstruction and the amount of training data to let the training of the CNN converge. Analogous to the original BBB-FDK, our method can reconstruct a huge segmented 3D CT volume using a standard GPU with a comparatively small graphics memory, and the training data constructed with only a set of projections given by a single CT scan was proven to be sufficient. Along with these advantages inherited from the original BBB-FDK, our approach obtains high segmentation quality for ultra-sparse projections, i.e., the number of views is 1/10 of the image width while eliminating sparse-view artifacts arising from the conventional FDK algorithm.

**Contributions:** The technical contributions of this study are summarized as follows:

- This study introduces a new CT image segmentation method based on end-to-end deep learning, where the CNN obtains a segmented CT image directly from projection images;

- This study shows that the rich spectral information of MECT is beneficial even in the context of DL-based CT image segmentation;
- This study shows several experimental results for the proposed method to the cone-beam CT data in medical applications, such as distinguishing bones and teeth from soft tissues.

## 2. Data Construction

This section introduces the process of extracting the input and output for neural network training, which follows BBB-FDK, the recent CNN-based FDK algorithm [19]. The input is image fragments extracted from each projection. We achieve this extraction through the back-projection of a bar, consisting of vertically connected voxels in the image domain, to each projection domain following the geometry of an X-ray CT device. The output is the bar of voxels, each of which stores the probabilities for the materials present in the voxel. The determination of the *subvoxel-level* material composition involves calculating the volume of the polyhedron enclosed by the isosurfaces. We obtain the polyhedron for each voxel using the traditional marching cubes algorithm [11]. The subsequent subsections provide more elaborated explanations for these processes of obtaining input and output data for training.

### 2.1. Input: Projection region extraction

Given the geometry of the X-ray CT device, we can find the pixel on a detector that exists in a line of X-ray radiation traveling through a voxel. Therefore, we can extract the bar of pixels on the projection associated with a bar of voxels in the image domain. In addition, the FDK algorithm, analogous to the filtered back-projection (FBP) algorithm, applies a horizontal image filter, such as Ram–Lak and Shepp–Logan filters, to each scanline of the projection image. Inspired by this behavior, we extract the partial area of projection images rather than the simple bar of pixels. Accordingly, we extract an image area of $w_p \times h_p$ pixels from each projection, where $w_p$ and $h_p$ are the width and height of the area. Suppose that there are $N_p$ projections with $N_e$ energy levels. Then, we can obtain the input for the neural network as a volume of $(N_p, h_p, w_p, N_e)$ pixels. The examples for the partial sinogram extracted from MECT data are shown at the left of Fig. 1.

### 2.2. Output: Subvoxel-level material composition

Unlike the training data used in conventional CT segmentation methods [4, 17], our method constructs the ground-truth output data for supervised learning at a finer level, i.e., subvoxel-level estimation of the material composition within each voxel. In this way, we can not only determine whether a voxel belongs to material A or material B but also provide a more detailed analysis of the composition such that a voxel consists of 2% air, 90% material A, and 8% material B.

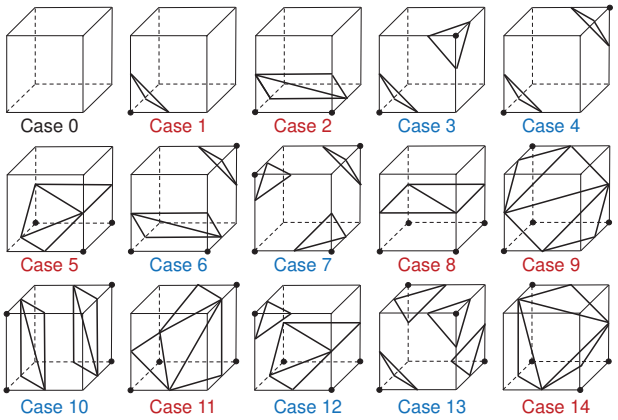


Figure 2. Triangle configurations for boxes whose corners are marked either inside or outside the object. These rules of triangle configurations used in the standard marching cubes algorithm can be classified into two: one is configured with a single set of adjoining triangles (type (a)), and the other includes multiple islands of triangles (type (b)).

We obtain such subvoxel-level material composition through MECT simulation and segment the simulated CT image based on the marching cubes algorithm [11]. The marching cubes algorithm finds the surfaces of objects in each voxel using the information of to which material each corner corresponds. As shown in Fig. 2, the surfaces are obtained as a set of triangles, and they divide the voxel into several regions. The classic marching cubes algorithm employs 14 triangle patterns shown in Fig. 2, where the black circles indicate a voxel within the isosurface, meaning its voxel value is higher than the iso-value. Figure 2 shows that we can classify these 14 cases into two categories: (a) a voxel is separated into two by an isosurface of adjoining triangles (shown with red text); (b) a voxel is separated into more than two regions by two or more isosurfaces of triangles (shown with blue text). Based on these categorizations, we determine the material composition in each voxel by the following steps.

**Step 1:** Assume that a given object consists of distinct components, each of which is made of only one of $n$, i.e., $\mu_1, \mu_2, \ldots, \mu_n$. In addition to simulating the sparse-view projection of the entire object, we simulate the full-view projection of the set of components made of each material. In other words, we perform a sparse-view CT simulation for the entire object and $n$ full-view CT simulation for the materials of the target object.

**Step 2:** We reconstruct a CT volume for every $n$ material using the full-view projection data and then apply the

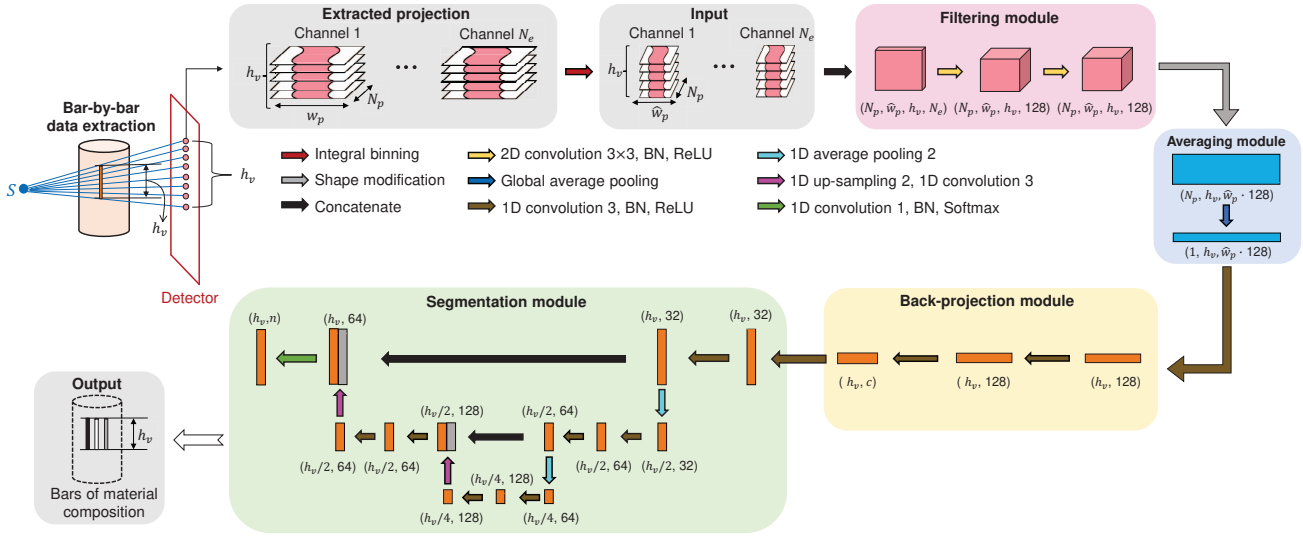


Figure 3. The network architecture of the end-to-end bar-by-bar segmentation method. Given a bar of voxels from the target object, the input of the network is a set of pixels in the corresponding projection regions of the bar in all energy channels, and the output is the material composition within each voxel in this bar. The whole network consists of two parts. The former performs CT reconstruction and is composed of filtering, averaging, and back-projection modules. The latter performs image segmentation and is comprised of a segmentation module. The network learns the mapping between the projection regions and the material composition.

marching cubes algorithm [11] to extract the isosurface for each material.

**Step 3:** For each voxel, we compute the polyhedron volume of each material and the air using the triangles given by the marching cubes algorithm. Then, we compute the ratio of materials of which each voxel consists. This process slightly differs between voxels of type (a) and type (b) depicted in Fig. 2.

(a) The voxel of type (a) includes only one closed polygon $P_k$ associated with the material $\mu_k$. Therefore, the ratio of the material composition is computed as $\rho(\mu_k) = V(P_k)/s^3$, where $V(P_k)$ is the volume of the internal region of $P_k$ and $s$ is the size of the voxel as a regular cube.

(b) The voxel of type (b) may include multiple closed polygons $P_{k,1}, P_{k,2}, \ldots$, associated with the material $\mu_k$. Therefore, in this case, we calculate the ratio of the material composition as $\rho(\mu_k) = \sum_i V(P_{k,i})/s^3$.

It should be noted that this process of computing the composition is performed $n$ times for each voxel, given $n$ CT volumes associated with the materials.

## 3. MECT Segmentation Network

To let the CNN learn the relationship between the input and output data constructed as in the previous section, we build a network architecture consisting of three computation modules, i.e., filtering, back-projection, and segmentation

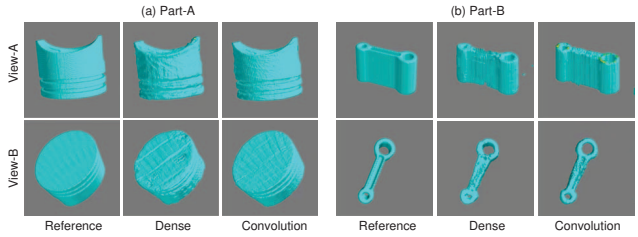


Figure 4. Comparison of a segmented part generated by dense and convolutional reconstruction modules.

modules. As illustrated in Fig. 3, our network is considered to be an extension of the network proposed by the BBB-FDK paper [19], where the segmentation module based on the U-net follows the BBB-FDK's network to achieve CT image segmentation.

The left side of the diagram illustrates the data extraction process. For a voxel bar, the corresponding projection region on the detector is a 4D tensor of shape $(N_p, \hat{w}_p, h_v, 1)$. We collect projection regions from $N_e$ energy channels and compress their horizontal lengths to $\hat{w}_p$ through integral binning proposed in [19]. Subsequently, the projection regions are concatenated along the energy channel axis and fed into the BbB-FDK reconstruction network as proposed in [19]. Here, in the back-projection module, instead of using dense layers as in the original method, we employ 1D convolutions along the last axis of the averaged tensor. Thus, we can effectively handle different bar formations by leveraging the translation invariance of convolutions.

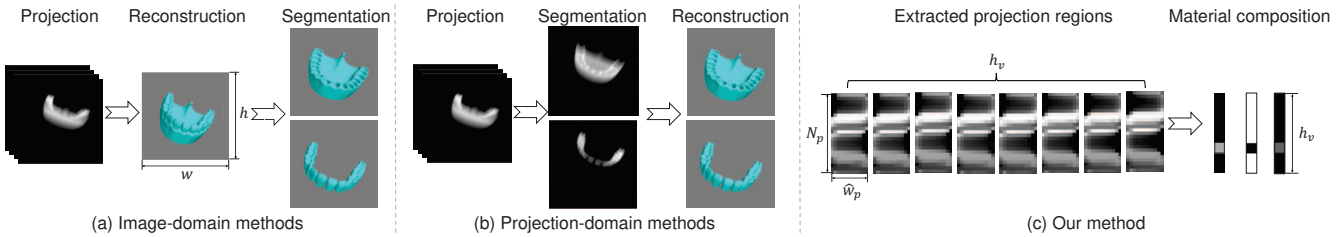


Figure 5. Comparison of datasets used in image-domain methods, projection-domain methods, and our proposed method. In the former two methods, the network is trained on the entire object as a single training data and involves an intermediate reconstruction step in addition to the segmentation step. In contrast, our method trains the network on a small portion of the projection data and directly outputs the material composition of a bar of voxels.

Figure 4 compares the effect of base layers, i.e., dense layer (also known as a fully connected layer) and convolutional layer, used in the segmentation results achieved by the filtering and back-projection modules. To highlight the difference in segmentation qualities given by varying base layers, we used a mechanical component with a smooth surface, shown in Fig. 4(a). Two different views of the segmented part are presented. Notably, the segmentation originating from dense layers exhibits an uneven surface in view-A and a fractured surface in view-B. In contrast, the network comprising convolutional layers yields a much smoother surface. These observations imply the advantage of convolutional layers that leverage weight sharing and local connectivity to enhance the reproduction of the surface geometry. Compared to the convolutional layers, dense layers are prone to overlook the spatial relationships between voxels, potentially introducing noise or inconsistencies in segmentation tasks. Thus, dense layers result in lacking surface smoothness in the segmented results.

Following the back-projection module, the processed bars, each corresponding to an energy channel, undergo segmentation through a shallow U-net architecture designed to accommodate the limited voxel heights of the bars. Employing 1D convolutions with a kernel size of 3, coupled with batch normalization and ReLU activation, the encoder employs 32, 64, and 128 feature maps successively. Dimension reduction is achieved using 1D average pooling with a size of 2. Meanwhile, the decoder uses 64 and 32 feature maps, incorporating 1D up-sampling of size 2. The final layer employs a 1D convolution followed by the softmax activation, yielding $n$ bars with distinct material compositions. The adoption of U-net is substantiated by its ability to capture local and global features effectively. The encoder-decoder architecture with skip connections often obtains feature maps that correlate with the input data, thus it works more efficiently for segmenting a small portion of data like bars of voxels that we handle in the proposed method.

### 3.1. Computational efficiency

Figure 5 compares the *volume*-level CT segmentation methods (i.e., (a) image-domain method and (b) projection-domain method) and (c) our *subvoxel*-level method. In Fig. 5 (a), $w$ and $h$ refer to the horizontal and vertical size of the reconstruction volume. In Fig. 5 (c), $N_p$, $\hat{w}_p$ and $h_v$ refer to the number of views, the horizontal length of projection regions after binning, and the number of bar voxels, respectively. In our implementation, $w = h = 256$, $N_p = 25$, $\hat{w}_p = 11$ and $h_v = 8$. In both reconstruction and projection-domain methods, the entire CT data of the object is considered a single training data, resulting in high GPU memory requirements up to 20 GB. Moreover, an additional reconstruction algorithm is applied before or after the segmentation, which can introduce artifacts into the final results. On the contrary, our method divides the object into bars and learns the direct mapping between bar projections and voxel composition, enabling efficient processing of large-scale data on standard GPUs and eliminating artifacts. Moreover, the required memory of our method is only tens of kilobytes.

### 3.2. Implementation

Our proposed method was implemented in Python, leveraging TensorFlow and Keras as machine learning software libraries. During training, we optimize the trainable parameters using the adaptive momentum estimation method (ADAM) [10] to minimize the cross-entropy loss for classification tasks, which is computed for pairs of the network output and a corresponding bar region of the correct CT segmentation. The learning rate is set to $1 \times 10^{-4}$, and the network is trained for 10 epochs with a batch size of 20.

For the training data, the input projection data consisted of 24 views evenly sampled from 0° to 360°, and each projection had a size of $256 \times 256$. Meanwhile, the output segmentation data is created by reconstructing the 250-view scatter-corrected projections by the FDK algorithm [3], then extracting the material composition by the method described in Section 2.2. During the training process, we employed the octree sampling, as described in [19], to densely extract training data in regions with significant CT value variations. Here, we used the tolerance of octree $C_{\mathrm{tol}} = 0.001$ and reduced nearly 80% of the sampling points (see [19] for more details). This approach achieves a

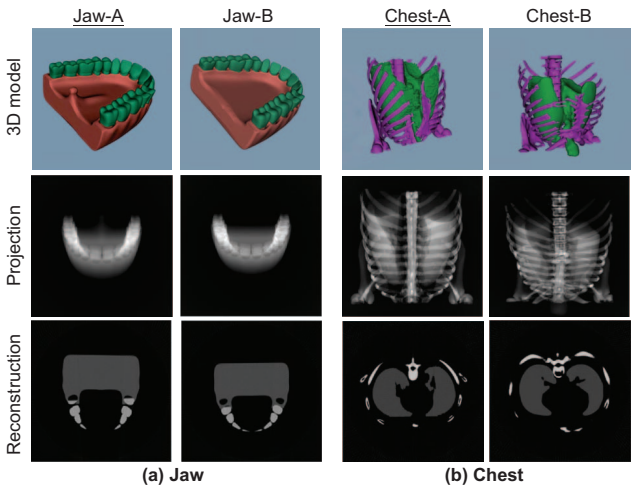


Figure 6. The experiment samples used in our experiments. The "Jaw" group consists of teeth and gum, where Jaw-A and Jaw-B represented the lower jaw and upper jaw of one patient. The "Chest" group includes rib and lung, with Chest-A and Chest-B derived from two patients. The training data for each group is indicated by underlining its name.

diverse composition of bars and reduces the redundancy in training data, resulting in improved prediction quality and an accelerated training process. On the other hand, we utilized sparse projection data with 24 views for the test data.

## 4. Experiments

We validate the proposed method using simulated samples illustrated in Fig. 6. The first row of the figure shows the 3D model of the simulated samples, the second row shows one of the projections of the samples scanned by 150 kV single-energy X-ray, and the last row exhibits the center slice of the reconstructed volume using the acquired projections. The samples were acquired from two groups: the "Jaw" group and the "Chest" group. The "Jaw" group consists of teeth and gums from a lower jaw (Jaw-A) and an upper jaw (Jaw-B) of a single patient. The "Chest" group includes ribs and lungs from two different patients, referred to as Chest-A and Chest-B, respectively. We conducted separate training and testing for each group. The training data, indicated by the underline, were used to train the network, and the trained networks were tested using the other sample of the same group. The MECT simulation is performed using aRTist [1], a commercial CT simulation software. During the simulation, we set an X-ray filter made of copper with 2.0 mm thickness.

### 4.1. Segmentation performance evaluation

In this experiment, we validated the segmentation performance of sparse-view MECT using Jaw and Chest samples. We trained four separate networks using MECT and single-energy CT (SECT) data of Jaw-A and Chest-A samples, respectively. Subsequently, we compared the performance of these networks on Jaw-B and Chest-B samples, aligning with the corresponding energy settings. For the MECT scans, we employed a tube voltage configuration consisting of 8 bins, spanning the range of 0–20 kV, 20–40 kV, ..., up to 140–150 kV. Conversely, for SECT, we determined the tube voltage as the median value of the MECT scans, specifically 80–100 kV.

Figure 7 compares the segmentation results of Jaw-B and Chest-B samples obtained from SECT and MECT scans. The reference segmentation, displayed on the left side of each group, is the material composition obtained from the reconstruction of the full-view scatter-corrected projections of each part in the target object. Here, we used the traditional FDK algorithm to reconstruct a CT image. The middle and right columns exhibit the extracted surfaces, which are obtained by applying the marching cubes algorithm [11] to the ratio of material composition estimated by the network. Two views are presented for each segmented part. Figure 7 demonstrates that both SECT and MECT appear to separate the target materials successfully from sparse-view projections. However, the segmentation result with SECT for the tooth (see Fig. 7(a)) shows a notable presence of noise. Moreover, the SECT result for the lung in Fig. 7(b) erroneously includes a part of the ribs. In contrast, the segmentation results obtained using MECT data for both samples exhibit smoother object surfaces and include significantly less noise.

In addition to the qualitative evaluation above, we show the quantitative evaluation in Table 1. The two columns at the left of Table 1 present Sørensen–Dice coefficients (DICE) between the reference data and the results obtained from SECT and MECT, respectively. The DICE value range from 0 to 1, and the performance is higher when the DICE value is closer to 1. Notably, the DICE value for MECT is substantially higher than that for SECT. This observation suggests that MECT improves the accuracy of the segmentation process significantly owing to the richness of information for material properties in MECT data. In contrast, SECT demonstrates suboptimal performance when a specific energy range is utilized. Namely, SECT fails to provide sufficient information for accurately segmenting the material combination within our target object. Consequently, employing MECT is advisable to achieve more reliable segmentation results.

### 4.2. Influence of different energy resolutions

For MECT-based segmentation, we also experimented with the effect of the resolution of energy bins. A narrower energy bin provides higher spectral resolution and improved material discrimination, but it also introduces noise due to

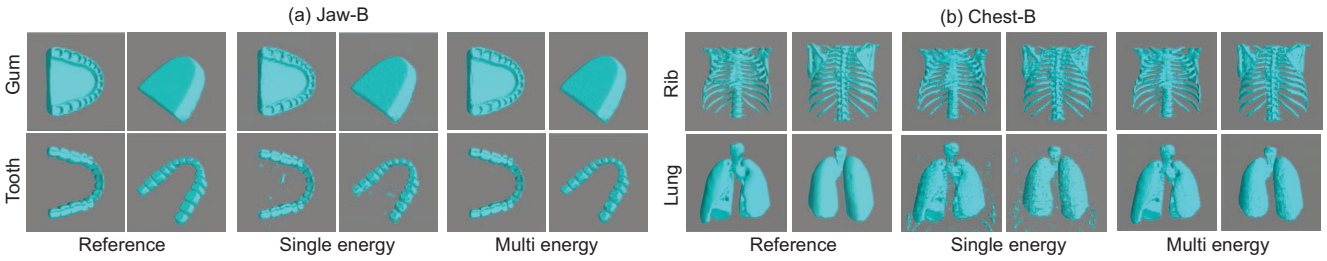


Figure 7. The comparison of segmentation results between SECT and MECT of Jaw-B and Chest-B. The SECT results are predicted by the SECT-trained network, while the MECT results are predicted by the MECT-trained network.

the presence of fewer photons in a narrow bin [15]. In this experiment, our objective was to reveal a reasonable balance between higher segmentation precision and reduced noise associated with bins with narrow energy ranges. To this end, we compared the segmentation results using four different energy resolution settings within the total range of 150 kV. The energy settings we utilized were: (i) 8 bins with a 20 kV increment (0–20 kV, 20–40 kV, ..., 140–150 kV), (ii) 5 bins with a 30 kV increment (0–30 kV, 30–60 kV, ..., 120–150 kV), (iii) 3 bins with a 50 kV increment (0–50 kV, 50–100 kV, 100–150 kV), and (iv) a single bin with a 150 kV increment (0–150 kV). Each network was trained and tested using the samples scanned by the same resolution setting.

The experimental results are shown in Fig. 8. Let us compare the result for 150 kV increment in this figure and the SECT result in Fig. 7. The comparison suggests that even the 150 kV increment, which results in a single energy bin equivalently with SECT, outperforms SECT-based segmentation, indicating that a broader energy range improves segmentation quality. On the other hand, when comparing all results of four energy resolutions to the reference data, a decrease in segmentation precision is observed as energy resolution decreases. Notably, the 20 kV increment result exhibits a smooth surface without noise. In contrast, the surfaces of other results become less smooth, particularly noticeable in the segmentation of the gum and lung. Furthermore, as the resolution decreases, noise in the tooth segmentation becomes more pronounced. The DICE shown in the second to last rows of Table 1 demonstrate performance loss as the energy resolution decreases. These observations suggest that denser energy bins contribute to high segmentation accuracy, with the noise issue effectively mitigated by the neural networks. Thus, our method achieves high segmentation precision effectively by increasing the energy resolution, suppressing more strongly noise compared to the conventional SECT-based approach.

Table 2 presents the segmentation times obtained using networks trained with different energy settings. The experiment here was carried out on a computer featuring a 3.6 GHz Intel Xeon E5-1650 v4 CPU, an NVIDIA GeForce RTX 3090 graphics card with 24GB of dedicated memory, and a total of 256 GB RAM. The input size of a dense energy setting can be several times larger than that of a sparse energy setting, resulting in longer processing times. Here, the increase of the computational time is not linear with respect to the number of energy bins, which means using the higher energy resolution can often be a better choice.

| | Jaw-B | Chest-B |
|---|---|---|
| Single energy | 0.966 | 0.960 |
| 20 kV increment, 8 bins | 0.990 | 0.986 |
| 30 kV increment, 5 bins | 0.989 | 0.985 |
| 50 kV increment, 3 bins | 0.983 | 0.980 |
| 150 kV increment, 1 bin | 0.980 | 0.978 |

Table 1. Dice coefficients (DICE) comparing the reference data with the segmentation results obtained from 80–100 kV SECT, 20 kV increment MECT, 30 kV increment MECT, 50 kV increment MECT, and 150 kV SECT. A higher DICE value indicates higher segmentation quality.

| | Jaw-B & Chest-B |
|---|---|
| Single energy | 20 min |
| 20 kV increment, 8 bins | 33 min |
| 30 kV increment, 5 bins | 30 min |
| 50 kV increment, 3 bins | 26 min |
| 150 kV increment, 1 bin | 20 min |

Table 2. Segmentation time required to process the different number of energy levels.

## 5. Conclusion and Future Work

In this paper, we proposed an end-to-end CT segmentation method for sparse-view MECT. We combined an FDK-simulated CT reconstruction neural network from [19] with a U-net segmentation network. Instead of processing the

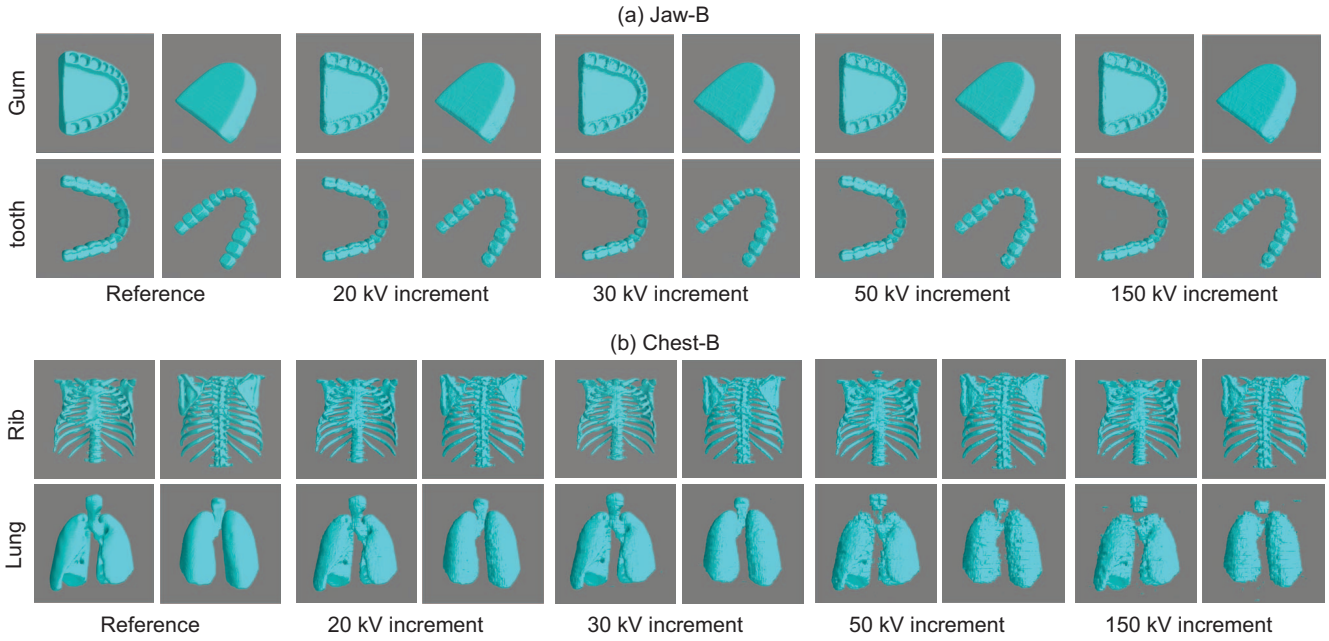


Figure 8. The segmentation results of CT data scanned by different energy resolutions. Each network is trained and tested by two similar samples scanned by the same energy setting.

entire object, we divided it into voxel bars and learned the mapping between the projection regions of a bar in different energy channels and the material composition of each voxel within the bar. Here, we defined the projection region of a single voxel as the horizontal extension of its projected trajectory, while the projection regions of a bar consist of several vertically-stacked trajectories. Within each voxel, the material composition of a specific target material was determined by the volume ratio of the polyhedron formed by the isosurfaces contained in that voxel. This data extraction approach allowed us to extract sufficient training data from a limited number of samples and process large-scale 3D CT data without memory constraints. Using this end-to-end segmentation network, we achieved artifact-free segmentation from sparse-view MECT by using only a limited number of training samples. Moreover, we demonstrated that our method achieves high segmentation accuracy by utilizing high energy resolution while effectively mitigating noise generated by narrow bins. The increased energy resolution improved the segmentation quality owing to the richer information on the absorption properties for various combinations of materials. The improved segmentation performance using MECT will facilitate the discrimination of materials with similar X-ray absorption properties.

In future work, we are interested in enhancing a filtering module to focus more on the varying contributions of energy bins. Such an update will allow the network to identify specific energy bins that affect the segmentation process more, as not all bins are considered to contribute equally. By selectively focusing on the most informative energy bins, we anticipate improvements in both the segmentation quality and the efficiency of the training process. Additionally, this approach could reduce the image acquisition time by recommending that only some specific energy bins are enough for segmentation.

## Acknowledgment

The authors would like to thank Mitsuhiko Matsukawa and Takumi Hotta for their constructive discussion on this study. This work is supported by a JSPS Grant-in-Aid for Early-Career Scientists (22K17907).

## References

[1] Carsten Bellon and Gerd-Ruediger Jaenisch. aRTist — analytical RT inspection simulation tool, 01 2007.

[2] Mats Danielsson, Mats Persson, and Martin Sjölin. Photon-counting x-ray detectors for CT. *Physics in Medicine & Biology*, 66(3):03TR01, 2021, DOI: 10.1088/1361-6560/abc5a5.

[3] L. A. Feldkamp, L. C. Davis, and J. W. Kress. Practical cone-beam algorithm. *Journal of the Optical Society of America A*, 1(6):612, 1984, DOI: 10.1364/josaa.1.000612.

[4] Ruiquan Ge, Huihuang Cai, Xin Yuan, Feiwei Qin, Yan Huang, Pu Wang, and Lei Lyu. MD-UNET: Multi-input dilated u-shape neural network for segmentation of bladder cancer. *Computational Biology and Chemistry*, 93:107510, 2021, DOI: 10.1016/j.compbiolchem.2021.107510.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron

Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, oct 2020, DOI: 10.1145/3422622.

[6] Mohamed A. A. Hegazy, Myung Hye Cho, Min Hyoung Cho, and Soo Yeol Lee. U-net based metal segmentation on projection domain for metal artifact reduction in dental CT. *Biomedical Engineering Letters*, 9(3):375–385, 2019, DOI: 10.1007/s13534-019-00110-2.

[7] Qinhua Hu, Luís Fabrício de F. Souza, Gabriel Bandeira Holanda, Shara S.A. Alves, Francisco Hércules dos S. Silva, Tao Han, and Pedro P. Rebouças Filho. An effective approach for CT lung segmentation using mask region-based convolutional neural networks. *Artificial Intelligence in Medicine*, 103:101792, 2020, DOI: 10.1016/j.artmed.2020.101792.

[8] Jing Huang, Yunwan Zhang, Jianhua Ma, Dong Zeng, Zhaoying Bian, Shanzhou Niu, Qianjin Feng, Zhengrong Liang, and Wufan Chen. Iterative image reconstruction for sparse-view CT using normal-dose image induced total variation prior. *PLoS ONE*, 8(11):e79709, 2013, DOI: 10.1371/journal.pone.0079709.

[9] Pavel Iassonov, Thomas Gebrenegus, and Markus Tuller. Segmentation of x-ray computed tomography images of porous materials: A crucial step for characterization and quantitative analysis of pore structures. *Water Resources Research*, 45(9), 2009, DOI: 10.1029/2009wr008087.

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[11] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, volume 21. ACM, 1987, DOI: 10.1145/37401.37422.

[12] Anand M. and G.G. Rajput. Automatic detection of abnormalities associated with abdomen and liver images: A survey on segmentation methods. *International Journal of Computer Applications*, 140(4):1–9, 2016, DOI: 10.5120/ijca2016909271.

[13] Sebastian Nowak, Maike Theis, Barbara D. Wichtmann, Anton Faron, Matthias F. Froelich, Fabian Tollens, Helena L. Geißler, Wolfgang Block, Julian A. Luetkens, Ulrike I. Attenberger, and Alois M. Sprinkart. End-to-end automated body composition analyses with integrated quality control for opportunistic assessment of sarcopenia in CT. *European Radiology*, 32(5):3142–3151, 2021, DOI: 10.1007/s00330-021-08313-x.

[14] Hongliang Qi, Zijia Chen, Jingyu Guo, and Linghong Zhou. Sparse-view computed tomography image reconstruction via a combination of L1 and SL0 regularization. *Bio-Medical Materials and Engineering*, 26(s1):S1389–S1398, 2015, DOI: 10.3233/bme-151437.

[15] Prabhakar Rajiah, Anushri Parakh, Fernando Kay, Dhiraj Baruah, Avinash R. Kambadakone, and Shuai Leng. Update on multienergy CT: Physics, principles, and applications. *RadioGraphics*, 40(5):1284–1308, 2020, DOI: 10.1148/rg.2020200038.

[16] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Lecture Notes in Computer Science*, pages 234–241. Springer International Publishing, 2015.

[17] Adnan Saood and Iyad Hatem. COVID-19 lung CT image segmentation using deep learning methods: U-net versus SegNet. *BMC Medical Imaging*, 21(1), 2021, DOI: 10.1186/s12880-020-00529-5.

[18] M. Haitham Shammaa, Yutaka Ohtake, and Hiromasa Suzuki. Segmentation of multi-material CT data of mechanical parts for extracting boundary surfaces. *Computer-Aided Design*, 42(2):118–128, 2010, DOI: 10.1016/j.cad.2009.08.003.

[19] Siqi Wang, Tatsuya Yatagawa, Yutaka Ohtake, and Hiromasa Suzuki. Sparse-view cone-beam CT reconstruction by bar-by-bar neural FDK algorithm. *Nondestructive Testing and Evaluation*, pages 1–23, 2023, DOI: 10.1080/10589759.2023.2195646.

[20] Xiaoqin Wei, Xiaowen Chen, Ce Lai, Yuanzhong Zhu, Hanfeng Yang, and Yong Du. Automatic liver segmentation in CT images with enhanced GAN and mask region-based CNN architectures. *BioMed Research International*, 2021:1–11, 2021, DOI: 10.1155/2021/9956983.

[21] Alexander D. Weston, Panagiotis Korfiatis, Timothy L. Kline, Kenneth A. Philbrick, Petro Kostandy, Tomas Sakinis, Motokazu Sugimoto, Naoki Takahashi, and Bradley J. Erickson. Automated abdominal segmentation of CT scans for body composition analysis using deep learning. *Radiology*, 290(3):669–679, 2019, DOI: 10.1148/radiol.2018181432.

[22] Yifu Xu, Bin Yan, Jian Chen, Lei Zeng, and Lei Li. Projection decomposition algorithm for dual-energy computed tomography via deep neural network. *Journal of X-Ray Science and Technology*, 26(3):361–377, 2018, DOI: 10.3233/xst-17349.

[23] Zhicheng Zhang, Xiaokun Liang, Xu Dong, Yaoqin Xie, and Guohua Cao. A sparse-view CT reconstruction method based on combination of DenseNet and deconvolution. *IEEE Transactions on Medical Imaging*, 37(6):1407–1417, 2018, DOI: 10.1109/tmi.2018.2823338.

[24] Zhongwei Zhao, Yuewen Sun, and Peng Cong. Sparse-view CT reconstruction via generative adversarial networks. In *2018 IEEE Nuclear Science Symposium and Medical Imaging Conference Proceedings (NSS/MIC)*, pages 1–5. IEEE, 2018, DOI: 10.1109/nssmic.2018.8824362.