Original paper

Segmentation of bones in medical dual-energy computed tomography volumes using the 3D U-Net

José Carlos González Sánchez\textsuperscript{a}, Maria Magnusson\textsuperscript{a,b,c}, Michael Sandborg\textsuperscript{a,b}, Åsa Carlsson Tedgren\textsuperscript{a,b,d}, Alexandr Malusek\textsuperscript{a,b,s}

\textsuperscript{a} Radiation Physics, Department of Medical and Health Sciences, Linköping University, Linköping SE-581 83, Sweden
\textsuperscript{b} Center for Medical Image Science and Visualization (CMIV), Linköping University, Linköping SE-581 83, Sweden
\textsuperscript{c} Computer Vision Laboratory, Department of Electrical Engineering, Linköping University, Linköping SE-581 85, Sweden
\textsuperscript{d} Department of Medical Radiation Physics and Nuclear Medicine, Karolinska University Hospital, Stockholm SE-171 76, Sweden

\textbf{ARTICLE INFO}

**Keywords:**
Deep learning
Convolutional neural network
Segmentation
Dual-energy computed tomography

**MSC:**
92B20

**ABSTRACT**

Deep learning algorithms have improved the speed and quality of segmentation for certain tasks in medical imaging. The aim of this work is to design and evaluate an algorithm capable of segmenting bones in dual-energy CT data sets. A convolutional neural network based on the 3D U-Net architecture was implemented and evaluated using high tube voltage images, mixed images and dual-energy images from 30 patients. The network performed well on all the data sets; the mean Dice coefficient for the test data was larger than 0.963. Of special interest is that it performed better on dual-energy CT volumes compared to mixed images that mimicked images taken at 120 kV. The corresponding increase in the Dice coefficient from 0.965 to 0.966 was small since the enhancements were mainly at the edges of the bones. The method can easily be extended to the segmentation of multi-energy CT data.

1. Introduction

Tissue segmentation provides a basis for further analysis of computed tomography (CT) images. Manual segmentation methods can be very tedious, time-consuming and subject to inter and intranidividual variability [1]. Automatic segmentation methods can overcome these problems, nevertheless their performance is in many cases adversely affected by low tissue contrast and image artifacts. Traditional segmentation methods are often based on conventional computer vision and machine learning approaches, see e.g. Ref. [2]. Recently, methods based on deep learning have demonstrated a potential to notably outperform the traditional methods in medical image analysis [3]. In case of segmentation, of special interest are convolutional neural networks (CNNs) [4]—a class of deep neural networks which utilizes the fact that input data have the form of images. These networks typically consist of convolutional layers, rectified linear unit (ReLU) layers and pooling layers. A convolutional layer consists of a set of learnable filters (kernels) that can detect specific features in the input image. The resulting feature maps represent abstractions of the input image; the deeper the CNN network is the more abstract information the feature maps contain. One or more subsequent fully connected layers combined with a softmax layer can be used for classification tasks. For image segmentation, however, the encoded abstract information has to be decoded to form the segmented image. This approach was first applied in the fully convolutional network (FCN) by Shelhamer et al. [5]. Better segmentation results were achieved when the encoder and decoder parts were organized in levels and connected with skip connections. This design known as the U-Net was first implemented by Ronneberger et al. [6] in 2D and later extended to 3D by Çiçek et al. [7]. A general problem of deep neural networks trained via gradient-based learning methods and backpropagation is that weights and biases in the “front” layers may receive only small updates. It was shown that this so called vanishing gradient problem can be mitigated by using residual blocks [8]. Another improvement of the U-Net designed to reduce the coarseness of the resulting segmented image is the use of segmentation maps with different resolutions at several levels of the U-Net; these segmentation maps are upsamled and added to form the final segmented image. This approach known as deep supervision was proposed by Kayalibay et al. [9]. Our work uses the modifications to U-Net proposed by Isensee et al. [10] who used both the residual blocks and the deep supervision. Moreover, they replaced the cross entropy loss function with the Dice loss function to cope with class imbalances and used extensive data augmentation to prevent overfitting.

Applications in radiation therapy can benefit from the use of dual-
energy CT (DECT) as it can provide more information about the material than the single-energy CT [11–14]. For instance the model-based image reconstruction (MBIR) algorithm DIRA [15] developed by the authors combines automated segmentation with material decomposition for the estimation of elemental composition and mass densities of tissues. These data can be used to derive electron densities and I-values. Advanced approaches based on multi-material decomposition [16] or the use of segmented-tissue-specific material bases [15] have been proposed. Nevertheless, good results can also be achieved with bone tissue and soft tissue as material bases. For such purposes, traditional methods have been developed [17,18]. Of interest is whether CNNs can improve the situation.

In case of the segmentation of bone, promising results were achieved via CNNs using two-dimensional kernels in a slice-by-slice segmentation [19] or via a pseudo-3D segmentation where axial, coronal and sagittal slices were used simultaneously to train the network [20]. The latter approach used the successful 2D U-Net developed by Ronneberger et al. for a whole body segmentation of the skeleton. All the cases mentioned above used single-energy CT data. The use of DECT data for the segmentation of lungs, spleen, liver and kidneys using an FCN based on the 3D U-Net design by Ćiçek et al. [7] was studied by Chen et al. [21]. They tested 3 different methods of handling the DECT data sets: the low- and high-energy data sets were (i) mixed (see the methods section); the mixing parameter was learned by the network, (ii) treated as separate input channels, and (iii) treated separately in the contracting part (see the methods section). The last method performed best, nevertheless it also had the highest memory requirements. In all three methods, patches with size (32, 32, 32) and overlap 16 were used. To our knowledge, no study published in scientific literature has focused on the segmentation of bones using both CNNs with three-dimensional kernels and the data from DECT or multi-energy CT scans.

The aim of this work is to develop and test a CNN based on the 3D U-Net architecture for the segmentation of bones in DECT. The network should be easily adapted to the segmentation of multi-energy CT data. Such solution can be used for instance in the MBIR algorithm DIRA [15], which focuses on applications in radiotherapy.

2. Material and methods

2.1. Data acquisition and annotation

Pseudonymized DECT data sets free of metallic parts and covering the pelvic area of 30 patients were obtained from a hospital Picture Archiving and Communication System (PACS) as follows: A list of patients whose abdomen or pelvis were examined using the Siemens SOMATOM Force 192-slice dual-source scanner in the dual-source mode between May 2017 and August 2019 was generated. The list was in consecutive order, no patients were removed. From this list, the first 30 data sets that did not contain metallic implants were selected. Acquisition and processing of these patient data was approved by the regional ethics committee (Dnr 2015/373-31).

Each data set consisted of the low (mostly 80 kV) and high (mostly 150 kV) tube voltage images and the mixed images (corresponding to approximately 120 kV [22] and produced by Siemens SOMATOM Force DECT scanners), see Table 1. A linearly mixed image with an intensity $I$ is produced as $I = wI_L + (1 - w)I_H$ where $I_L$ and $I_H$ are intensities of the low- and high-energy images, respectively, and $w$ is a mixing parameter [23]. In our case, $0.6 \leq w \leq 0.7$. The mixing parameter is typically chosen so that a resemblance with 120 kV images is achieved. The mixed images often result in better image quality than either the low- or high-energy images alone [24]. The images were converted from the Digital Imaging and Communications in Medicine (DICOM) format to the Neuroimaging Informatics Technology Initiative (NIfTI) format. Parts of the volumes other than the pelvic region (see the coronal view in Fig. 4) were discarded; each resulting volume consisted of 550 slices. Intensity values were represented by CT numbers increased by 1024 HU to avoid negative values.

Ground truth data needed for the training of the neural network and measuring the accuracy of the results were annotated manually in the ITK-SNAP software application [25] by correcting results of a traditional segmentation algorithm JJ2016 [18] applied on the mixed images.

2.2. Network architecture

The proposed neural network was based on the 3D U-Net design by Isensee et al. [10], see Fig. 1. The only difference was an additional instance normalization after the convolution in the segmentation layer. As in the original U-Net design by Ronneberger et al. [6], the net comprised a context aggregation pathway that encoded increasingly abstract representations of the input (the left side of the U-shape) and a localization pathway that recombined these representations with shallower features (the right side of the U-shape) and a localization pathway that recombined these representations with shallower features (the right side of the U-shape). Contrary to the original design by Ronneberger et al., the localization pathway was amended with a part employing deep supervision [9], which integrated segmentation layers at different levels of the network and combined them via element-wise summation.

The U-shape consisted of five levels. The first 3 × 3 × 3 stride 2 convolution block on the left side decreased the dimensionality and increased the number of output feature maps by the factor of 2. The exception was the top level that used one or two data sets as the input and did not perform the downsampling. This block was then followed by a pre-activation residual block [8] represented by a residual connection and a context module with two 3 × 3 × 3 convolution blocks and a dropout layer ($p_{dropout} = 0.3$) in between. The right side of each level took upsampled information from a lower layer and concatenated it with the abstract features from the aggregation pathway. The

Table 1

Mean Dice coefficients $D_H$ (high tube voltage images), $D_M$ (mixed images) and $D_{LH}$ (pairs of low and high tube voltage images) obtained during training for the validation patient data set $P$. Corresponding fold (iteration) numbers $F$, tube voltages $U$ and pixel sizes are also listed; slice thickness was 1 mm. Overall mean Dice coefficients for validation and training are listed in the last two rows.

<table>
<thead>
<tr>
<th>$F$</th>
<th>$P$</th>
<th>$U$ (kV)</th>
<th>pixel size (mm$^2$)</th>
<th>$D_H$</th>
<th>$D_M$</th>
<th>$D_{LH}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>80, 150</td>
<td>0.977 × 0.977</td>
<td>0.966</td>
<td>0.970</td>
<td>0.964</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>70, 150</td>
<td>0.727 × 0.727</td>
<td>0.968</td>
<td>0.972</td>
<td>0.976</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>11, 80</td>
<td>0.750 × 0.750</td>
<td>0.972</td>
<td>0.974</td>
<td>0.977</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>80, 150</td>
<td>0.894 × 0.894</td>
<td>0.975</td>
<td>0.972</td>
<td>0.975</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>80, 150</td>
<td>0.711 × 0.711</td>
<td>0.970</td>
<td>0.973</td>
<td>0.976</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>80, 150</td>
<td>0.742 × 0.742</td>
<td>0.973</td>
<td>0.974</td>
<td>0.974</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>80, 150</td>
<td>0.730 × 0.730</td>
<td>0.966</td>
<td>0.970</td>
<td>0.972</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10, 80</td>
<td>0.977 × 0.977</td>
<td>0.973</td>
<td>0.967</td>
<td>0.969</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>80, 150</td>
<td>0.775 × 0.775</td>
<td>0.969</td>
<td>0.969</td>
<td>0.969</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>70, 150</td>
<td>0.680 × 0.680</td>
<td>0.964</td>
<td>0.971</td>
<td>0.973</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>80, 150</td>
<td>0.746 × 0.746</td>
<td>0.974</td>
<td>0.976</td>
<td>0.977</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>80, 150</td>
<td>0.777 × 0.777</td>
<td>0.969</td>
<td>0.970</td>
<td>0.972</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>70, 150</td>
<td>0.670 × 0.670</td>
<td>0.953</td>
<td>0.959</td>
<td>0.962</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>80, 150</td>
<td>0.734 × 0.734</td>
<td>0.970</td>
<td>0.974</td>
<td>0.977</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>80, 150</td>
<td>0.639 × 0.639</td>
<td>0.972</td>
<td>0.975</td>
<td>0.977</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>80, 150</td>
<td>0.805 × 0.805</td>
<td>0.975</td>
<td>0.966</td>
<td>0.961</td>
</tr>
<tr>
<td>3</td>
<td>80, 150</td>
<td>0.773 × 0.773</td>
<td>0.979</td>
<td>0.979</td>
<td>0.978</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>80, 150</td>
<td>0.674 × 0.674</td>
<td>0.969</td>
<td>0.974</td>
<td>0.976</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>80, 150</td>
<td>0.676 × 0.676</td>
<td>0.975</td>
<td>0.978</td>
<td>0.980</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>80, 150</td>
<td>0.723 × 0.723</td>
<td>0.973</td>
<td>0.974</td>
<td>0.976</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>24</td>
<td>80, 150</td>
<td>0.777 × 0.777</td>
<td>0.968</td>
<td>0.970</td>
<td>0.972</td>
</tr>
<tr>
<td>5</td>
<td>80, 150</td>
<td>0.826 × 0.826</td>
<td>0.974</td>
<td>0.976</td>
<td>0.976</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>80, 150</td>
<td>0.676 × 0.676</td>
<td>0.973</td>
<td>0.978</td>
<td>0.980</td>
<td></td>
</tr>
</tbody>
</table>

Validation mean | 0.971 | 0.972 | 0.973 |
Training mean | 0.974 | 0.974 | 0.976 |
localization module (a $3 \times 3 \times 3$ convolution block followed by a $1 \times 1 \times 1$ convolution block) recombined these features together and the upsampling module increased the resolution and decreased the number of feature maps. Each convolution block comprised a convolution layer, an instance normalization and a leaky ReLU with the slope of 0.01.

### 2.3. Implementation details

The network was implemented using Keras [26] on top of TensorFlow [27] according to the code available on GitHub (https://github.com/ellisdg/3DUnetCNN). Data augmentation was performed on-the-fly and consisted of four affine transformations: scaling (zooming), shift (translation), rotation and flip (reflection) of the volumes. The transformation parameters were set in the NIFTI format headers. Random rotations in the range $[-30^\circ, 30^\circ]$, random scaling $[90\%, 110\%]$ in the lateral and anteroposterior directions, random translations up to 50 pixels in the same directions and a random horizontal flip were used.

Rotations and scaling leading to an out-of-range object were corrected by a translation. The data were stored in the Hierarchical Data Format (HDF5) [28] to avoid opening and processing different files in each step. Lazy evaluation [29] was used. The data were downsampled to $(128 \times 128 \times 128)$ to fit to the GPU’s memory. The binary prediction of the neural network was an array of the same dimensions.

In the training and evaluation of a neural network, the data should be split to training, validation and test data. The model should be trained using the training and validation data. Predictions of the model should be evaluated using the test data. In our case, 5 data sets were used as the test data and the remaining 25 data sets were used as training and validation data in a 5-fold cross validation scheme. In k-fold cross-validation the data sets are randomly split into k equal sized...
subsamples, see Fig. 2. One subsample is used for validation, the remaining \(k-1\) subsamples are used for training. The process is then repeated \(k\) times, with each of the \(k\) subsamples used exactly once as the validation data. In our case with \(k = 5\), the first iteration subsample (patients 18, 2, 11, 14, 12) was used for validation, the remaining 4 subsamples were used for training. Resulting Dice similarity coefficients, \(D\), for the validation data are reported in corresponding rows of Table 1 \((F = 1)\); they were calculated as

\[
D = \frac{2|A \cap B|}{|A| + |B|},
\]

where \(A\) and \(B\) were the segmented and ground truth voxel sets, respectively. The process was repeated for all 5 iterations; each iteration used a Keras model that was trained independently from the others. The whole procedure was applied on high tube voltage images (H), mixed images (M) and pairs of low and high tube voltage images (LH). Consequently, \(5 \times 3 = 15\) Keras models were trained. In the LH setups, the low and high tube voltage data were represented as channels 1 and 2 in Keras; the H and M setups used 1 channel only. Batch size was set to 1 owing to large memory demands of the models.

Multiclass Dice similarity coefficient defined as [10]

\[
\mathcal{D}_{ik} = \frac{2}{|K|} \sum_{k \in K} \sum_{i} u_{i,k} v_{i,k},
\]

Fig. 4. Example of 3D representation and views of the section (axial, sagittal and coronal) of one of the data sets prediction by 3D U-Net. The pixels classified as bone are shown in red on the CT image. Images obtained with ITK-SNAP.

Fig. 5. Segmented bone for (a) ground truth, (b) 3D U-Net and (c) the traditional JJ2016 algorithm. The segmented voxels (red) are fused with the mixed image.
Table 2
Dice coefficients for patients $P$ in the test data sets obtained for the H, M, and LH setups. $S$ is the number of subjects in each fold. The 5-fold cross-validation scheme ($D_{P,1}, \ldots, D_{P,5}$) was used to calculate Dice coefficients and averages over folds and patients ($\bar{D}_{P}$) are listed.

<table>
<thead>
<tr>
<th>$S$</th>
<th>$P$</th>
<th>$D_{P,1}$</th>
<th>$D_{P,2}$</th>
<th>$D_{P,3}$</th>
<th>$D_{P,4}$</th>
<th>$D_{P,5}$</th>
<th>$\bar{D}_{P}$</th>
<th>$\bar{D}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0.961</td>
<td>0.965</td>
<td>0.967</td>
<td>0.966</td>
<td>0.969</td>
<td>0.969</td>
<td>0.966</td>
</tr>
<tr>
<td>26</td>
<td></td>
<td>0.965</td>
<td>0.971</td>
<td>0.972</td>
<td>0.968</td>
<td>0.972</td>
<td>0.970</td>
<td>0.9632</td>
</tr>
<tr>
<td>27</td>
<td></td>
<td>0.975</td>
<td>0.975</td>
<td>0.977</td>
<td>0.972</td>
<td>0.975</td>
<td>0.975</td>
<td>0.975</td>
</tr>
<tr>
<td>28</td>
<td></td>
<td>0.967</td>
<td>0.969</td>
<td>0.965</td>
<td>0.970</td>
<td>0.968</td>
<td>0.968</td>
<td>0.966</td>
</tr>
<tr>
<td>29</td>
<td></td>
<td>0.975</td>
<td>0.974</td>
<td>0.974</td>
<td>0.973</td>
<td>0.974</td>
<td>0.974</td>
<td>0.974</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>0.975</td>
<td>0.975</td>
<td>0.978</td>
<td>0.974</td>
<td>0.976</td>
<td>0.976</td>
<td>0.976</td>
</tr>
<tr>
<td>H</td>
<td>28</td>
<td>0.960</td>
<td>0.958</td>
<td>0.956</td>
<td>0.960</td>
<td>0.959</td>
<td>0.959</td>
<td>0.959</td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>0.975</td>
<td>0.974</td>
<td>0.974</td>
<td>0.973</td>
<td>0.974</td>
<td>0.974</td>
<td>0.974</td>
</tr>
<tr>
<td>M</td>
<td>30</td>
<td>0.967</td>
<td>0.969</td>
<td>0.966</td>
<td>0.967</td>
<td>0.968</td>
<td>0.967</td>
<td>0.967</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>0.975</td>
<td>0.974</td>
<td>0.974</td>
<td>0.973</td>
<td>0.974</td>
<td>0.974</td>
<td>0.974</td>
</tr>
<tr>
<td>LH</td>
<td>29</td>
<td>0.962</td>
<td>0.962</td>
<td>0.956</td>
<td>0.959</td>
<td>0.960</td>
<td>0.960</td>
<td>0.960</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0.973</td>
<td>0.974</td>
<td>0.970</td>
<td>0.971</td>
<td>0.971</td>
<td>0.972</td>
<td>0.972</td>
</tr>
</tbody>
</table>

where $u$ is the softon output of the network, $v$ is a one-hot encoding of the ground truth and $K$ is the set of considered classes, was used as the loss function since it intrinsically handles class imbalances (different numbers of voxels in each class $k$). The network weights were initialized with a Xavier [30] method in order to keep the signal in a reasonable range of values across the many layers. Adam [31] with an initial learning rate of $10^{-4}$ was used as the optimization algorithm. Higher values ($10^{-4}$) made the network converge faster but increased the loss, lower values ($<10^{-5}$) increased training time without producing noticeable improvements. The learning rate was decreased by the drop factor of 0.5 when 10 epochs had passed without any improvement in validation loss. The training was stopped after 1000 epochs or when the validation loss had not improved after 50 epochs.

The five H, five M and five LH Keras models were then applied on the 5 test data sets. Dice coefficients for the predicted data were calculated for the whole segmented 3D volumes and individual segmented 2D slices using Eq. (1). The resulting Dice coefficients were analyzed with the statistical software R [32].

3. Results

The network converged around epoch 700 for all 15 models. In our case, the epoch was defined as the processing of all 20 randomly transformed (augmented) training data sets; the 5 validation data sets were not augmented. Thus each model was trained on approximately $20 \times 700 = 14000$ different data sets. We recall that training data consisted of $4 \times 5 = 20$ data sets in the 5-fold cross-validation scheme. An example for the fold 1 is shown in Fig. 3, where the loss defined by Eq. (2) as a function of epoch number is presented for both the training and validation data sets. Only the training data loss is used to update the weights and biases while the validation data loss determines whether the learning rate parameter is lowered or the training is stopped. For this reason, it is common that models better fit the training data, i.e. their loss is lower than the validation data loss. In our case, however, the utilization of data augmentation and dropout reversed this order.

Each epoch took approximately 81 and 82 s for inputs with one and two channels, respectively, on a computer with two Tesla K40 GPUs. For a single model, the training took approximately 16 h, all 15 models took 240 h. One prediction of the network (segmentation of a DECT volume re-scaled to $128 \times 128 \times 128$ voxels) took about 5.2 s.

An example of the resulting segmented bone is shown in Fig. 4. All results (H, M and LH) for validation data are available as GIP animations [33]. A comparison between the ground truth and segmentation results obtained with the proposed algorithm and the traditional JJ2016 algorithm is shown in Fig. 5. Note that the proposed algorithm correctly filled holes (bone cavities) and did not include vessels filled with the iodine contrast agent.

Dice coefficients, $D$, for all 5 validation data sets in each fold of the 5-fold cross-validation scheme are listed in Table 1. Table 2 lists Dice coefficients for patients in the test data. On the average, the LH setup increased the per-volume Dice coefficient by approximately 0.001 compared to the M setup for both validation and test data; the means were 0.9652 and 0.9662 for the M and LH setups. In case of validation data, this difference was statistically significant; a paired-samples t-test resulted in $p = 0.017$ (t(24) = -2.57). In case of test data, the number of patients was too small to make the difference statistically significant, see Fig. 6. As a workaround, the Dice coefficients were compared for each axial slice. As the corresponding distributions were non-normal, the Wilcoxon rank sum test (wilcoxon.test in R) was used; the p-value was 0.004. The null hypothesis of the test was that the two distributions differed by a location shift of 0, and the alternative was that they differed by some other location shift. Means and standard deviations of the per-slice Dice coefficients were (0.9620, 0.027) and (0.9626, 0.033) for the M and LH setups, respectively. Note that these means were slightly smaller than the corresponding per-volume ones, nevertheless, the increase of approximately 0.001 was observed too. The standard deviations of the distributions reflect the inter-patient variability of the distributions; they are notably larger than the increase of 0.001.

4. Discussion

The proposed segmentation method may be used in MBIR algorithms in DECT specializing on radiation treatment planning, for instance the brachytherapy of prostate with low energy photons. In this case, the same scanning protocols may be used for the acquisition of the training and prediction data. To fully automate the segmentation process, the manual truncation of the data sets in the z-direction may be replaced with a truncation using a localization network [34]. The number of segmented organs in the pelvic region may be increased. Chen et al. [21] have already demonstrated the feasibility of

Fig. 6. Box plots of Dice coefficients for the H, M and LH setups obtained for (a) validation data and (b) test data. The box extends from the lower to the upper quartile values of the data. A black dot and red star represent the median and mean, respectively.
segmentation of lungs, liver, spleen and kidneys via DECT and the U-Net.

The proposed algorithm outperformed the traditional JJ2016 algorithm [18] both in speed and accuracy; the computation time was shortened from 10–15 min to about 5 s. Other traditional segmentation algorithms may have notably shorter execution times, for instance Aslan et al. [35] report 116 s for the segmentation of vertebrae in 100 slices. However, it is not clear how the algorithm would handle the segmentation of all bones in the pelvic region, where the main problem in our application is the hole filling.

The high memory demand of the 3D U-Net architecture on the GPU has been overcome by downsampling the input data to 128 × 128 × 128 voxels. The resulting coarse resolution did not present a problem for large bones in the pelvic region. Some effects can, however, be expected for bones with complicated shapes like the vertebrae; so far, the most successful algorithms specialized on the segmentation of vertebrae achieved the Dice coefficient of 0.9577 [34] only. Our attempt to work around the memory problem by preserving the original resolution and using non-overlapping patches was not successful; the network produced artifacts at planes where the patches met. Tests with overlapping patches were not performed owing to limited GPU time at the supercomputer center.

An extension of the proposed algorithm to segmentation of multi-energy CT data sets is straightforward; an extra energy data represent an additional channel in the 3D U-Net network. The alternative independent contraction path approach by Chen et al. [21] promises better performance, nevertheless it does not fit to the 12 GB of GPU memory in the 128³ voxels resolution and requires the use of patches. We speculate that our high values of Dice coefficients resulted from the fact that we did not use patches.

The annotation of 3D ground truth data was time consuming and thus was performed by a non-expert. It is believed that this fact did not strongly bias the ground truth owing to the large contrast of bones in CT images.

The main purpose of the additions to the original U-Net architecture (residual blocks, dropout layers, deep supervision) was to improve the learning process. Results of segmentation contests show that these additions proved beneficial in particular competition tasks. An alternative approach is to focus on the optimization of the training process itself. Isensee et al. [36] tested a number of modifications proposed for encoder-decoder networks in the U-Net design. They found that these variants did not provide additional benefits compared to a well trained U-Net. This was demonstrated in their contribution to the BraTS 2018 challenge where they achieved state of the art segmentation (rank 2 out of 60 participants) with the U-Net without using significant architectural alterations. This fact makes the comparison of performance of different network architectures even more difficult. For this reason, we avoided a simplistic comparison of the presented solution with the state of the art U-Net; an in-depth comparison may be the subject of a future work.

5. Conclusions

It was demonstrated that the 3D U-Net performed well on all the data sets (high tube voltage, mixed images and dual-energy); the mean Dice coefficients for the validation and test data sets were larger than 0.971 and 0.963, respectively. Of special interest is that it performed better on dual-energy CT volumes compared to mixed images mimicking images taken at 120 kV. The corresponding increase of 0.001 in the Dice coefficient from 0.965 to 0.966 for the test data was small since the enhancements were mainly at the edges of the bones. Despite the good results, the implementation had some limitations; the data sets had to be down-sampled to fit the memory of the GPU.

Acknowledgments

The authors thank Bharti Kataria for help with selecting the data sets and the National Supercomputer Centre (NSC) at Linköping University for providing their infrastructure. The following grants are acknowledged: SNIC 2018/7–22, LiO-724181, CAN 2017/1029 and VR-NT 2016–05033.

References


