IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 17, 2019, accepted July 28, 2019, date of publication August 9, 2019, date of current version September 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2934325

Efficient Triple Output Network for Vertebral Segmentation and Identification

CHENG-HUNG CHUANG[1], CHIH-YANG LIN[2], (Member, IEEE), YUAN-YU TSAI[3], (Member, IEEE), ZHEN-YOU LIAN[1], HONG-XIA XIE[4], CHIH-CHAO HSU[2], AND CHUNG-LIN HUANG[3]

[1]Department of Computer Science Information Engineering, Asia University, Taichung 41354, Taiwan
[2]Department of Electrical Engineering, Yuan Ze University, Taoyuan 32003, Taiwan
[3]Department of M-Commerce and Multimedia Applications, Asia University, Taichung 41354, Taiwan
[4]Institute of Electronics, National Chiao Tung University, Hsinchu 30025, Taiwan

Corresponding author: Chih-Yang Lin (andrewlin@saturn.yzu.edu.tw)

This work was supported by the Ministry of Science and Technology, Taiwan, under Grant MOST 106-2218-E-468-001, Grant MOST 107-2221-E-155 -048 -MY3, and Grant MOST 108-2634-F-008 -001.

ABSTRACT Precise vertebral segmentation provides the basis for spinal image analyses and interventions, such as vertebral compression fracture detection and other abnormalities. Deep learning is a popular and useful paradigm for medical image process. In this paper, we proposed an iterative vertebrae instance segmentation model, which has good generalization ability for segmenting all types of vertebrae, including cervical, thoracic, and lumbar vertebrae. In experimental results, our model not only used 17% less memory but also achieves better performance on vertebrae segmentation compared to existing methods. The existing method provides only two output for segmentation and classification respectively. However, with more memory available, our model is capable of providing third output for accurate anatomical prediction under the same amount of memory.

INDEX TERMS Deep learning, vertebra segmentation, triple output.

I. INTRODUCTION

The main trunk in the human body is composed of vertebrae, which are adjacent to the critical nerve system. Many diseases, such as spinal spondylolisthesis, compression fractures, disc herniation, spurs, etc. are all clinically related to the vertebrae. Hence, accurate vertebral segmentation can improve preoperative diagnosis and simulation performance. Traditional vertebral segmentation primarily uses mathematical methods to analyze cone features [8], [9], [13] or different model-based for addressing model fitting problems [1], [2], [6]. Darwish *et al.* [4] integrated and detailed automatic segmen- tation techniques, including conventional- based, anatomical model-based and random forest-based, to identify and lo- cate individual vertebrae in traditional spines. Most of these traditional vertebral segmentation techniques mathematically analyze the characteristics of the threshold, edge and verte- bral morphology based on a 2D spine cut, and are applied to specific targets or shapes. Thus, such techniques cannot be widely applied to various abnormal spinal images. For traditional 3D automatic vertebra segmentation, Hoon *et al.* [10] proposed fitting the Level Set method to the spine model, which is roughly the same as the traditional 2D method. In the semi-automatic vertebral segmentation, the segmentation is mostly carried out by different watershed regions, and re- sults usually depend on the user's seed point selection. If the seed point selection is good, the segmentation results usually closely align with manual segmentation results even when the vertebral segment is abnormal or damaged. Unfortunately, seed point selection tends to be time-consuming and difficult to set.

Deep learning has rapidly developed in recent years and has been proven to be an effective means of handling the segmentation of medical images. The common way to segment vertebra based on deep learning is to use semantic segmentation to give a prediction label to all pixels in an image. The 2D U-Net deep learning network proposed by Sekuboyina *et al.* [12] for multi-label segmentation of the lumbar vertebra, and 3D U-Net both perform well in the segmentation of vertebrae.

The associate editor coordinating the review of this article and approving it for publication was Weipeng Jing.

Another school of thought approaches vertebral segmentation as an instance segmentation problem. For all vertebrae, the difference between semantic segmentation and instance segmentation is that semantic segmentation classifies each voxel in the patch directly, while the instance segmentation identifies whether a complete vertebra exists in the patch or not, performs binary classification, and labels vertebra individually. In this case, partially obscured vertebrae can be either ignored or explicitly identified as incomplete so that they can be excluded from subsequent analyses. Lessmann *et al.* [7] proposed an instance segmentation network for vertebra detection and segmentation. However, this kind of architecture requires over 11 GB of memory, which demands robust experiment conditions.

Although all the aforementioned methods have achieved decent segmentation results, they are not able to predict anatomical labels for each vertebra. The massive memory usage requirements of other techniques not only result in higher hardware costs, but also limit any model's capacity to conduct multiple tasks, such as anatomical prediction. Hence, the model proposed in this paper aims to decrease memory usage through a more efficient architecture, which allows an extra output for anatomical labeling. Our method effectively reduces memory usage by applying a different concatenation method in a U-Net shape-like structure, and carefully choosing the feature maps in a different part of the network. Existing models, like Lessmann *et al.* [7], only provide outputs for segmentation and classification. However, our method frees memory, enabling the model to provide a third output for anatomical information and higher segmentation accuracy without concerns about memory shortage that existing models face. Besides creating a robust model, we have extended our research to the real world through a collaboration with China Medicine University, Taiwan and Asia University, Taiwan. Our project aims to provide an automated design solution to position pedicle screw guide plates in vertebral correction surgery. The goal is to create a vertebral orthopedic surgical guide plate by automatically finding the best implant position. Details are included in Section V.

II. RELATED WORK

U-Net [11], which outputs a corresponding label for all pixels in an image, has been proposed for 2D semantic segmentation. The main structure consists of a contracting path and an expansive path. This network architecture has achieved rather good segmentation results within biomedical imaging. Sekuboyina *et al.* [12] applied U-Net to lumbar segmentation and trained three different dimensions of a 2D tomographic image.

As U-Net had performed well on vertebral segmentation, additional reference information was infused during training since vertebra images are 3D. For example, Cicek *et al.* [3] proposed a 3D U-Net architecture that allows the entire network architecture to directly train 3D data. However, the whole network needs 19,069,955 parameters total with data inputs sized to 132*132*116, which poses a huge challenge for computing equipment. Based on 3D U-Net, Janssens, *et al.* [5] proposed using 3D fully convolutional neural network (FCN) to perform lumbar vertebra multi-label segmentation. Although the deep learning of 3D data does not require addi- tional reference information, the number of feature parame- ters is very large, which leads to huge memory requirements in the training process.

The methods above deal with vertebral semantic segmentation, and Lessmann *et al.* [7] proposed a segmentation scheme that considers vertebral segmentation to be an instance segmentation problem. The architecture includes a segmentation network with an expansion called instance memory and a branch for completeness prediction. The size of the input patch is 128*128*128, which is enough to contain one complete vertebra in the center of the patch and the partial vertebrae next to it. The main idea behind this instance segmentation is to segment each vertebra sequentially from top to bottom or vice versa. The segmentation network is trained to extract the top-most unsegmented vertebra (when segmented from top to bottom), or bottom- most unsegmented vertebra (when segmented from bottom to top), in the input patch. Instance memory is used to record segmented vertebrae. For example, if there are five vertebrae in an image, the segmentation network will first segment the top-most vertebrae; the instance memory will then be updated with information that vertebra 1 has been segmented, so in the next iteration, the top-most unsegmented vertebra becomes the vertebra adjacent to vertebra 1. This process continues until all the vertebrae are segmented.

III. PROPOSED METHODS

Although the method proposed by Lessmann *et al.* [7] shows satisfying results in vertebral segmentation, the model has some drawbacks. First, the training process for this model requires massive amounts of memory (over 11GB on a GPU). Furthermore, this model does not provide anatomical label information (i.e., cervical C1 C8, thoracic T1 T12, lumbar L1 L5) of vertebrae. Additionally, the iterative instance seg- mentation procedure proposed by Lessmann *et al.* [7] starts from the very beginning of an image in a slide window fashion to find the first patch that contains enough bone voxels before starting sequential segmentation, which might contain redundant iterations. In this paper, we propose a new architecture inspired by 3D-UNET and Lessmann *et al.* [7] and also alter the way the starting point for the iterative seg- mentation process is chosen. Our goal is to efficiently tackle the issues mentioned previously. The network architecture is depicted in Figure 1. This model takes advantage of the contracting and expansive path in 3D-UNET with additional modifications and the instance memory and classification network adapted from Lessmann *et al.* [7]. There are three major differences between our model and that of Lessmann et al. [7]. First, the feature maps in the model from Lessmann *et al.* [7] are fixed at 64 in every layer; however, we choose the same number of

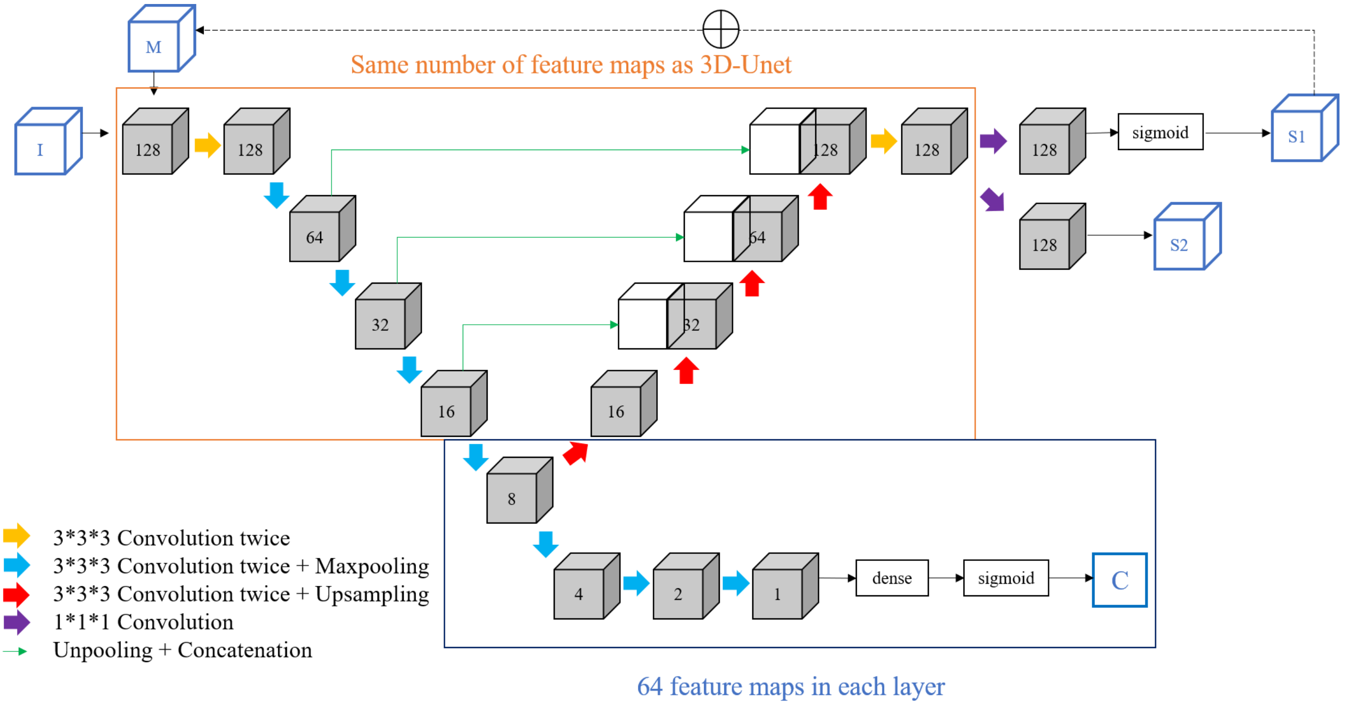
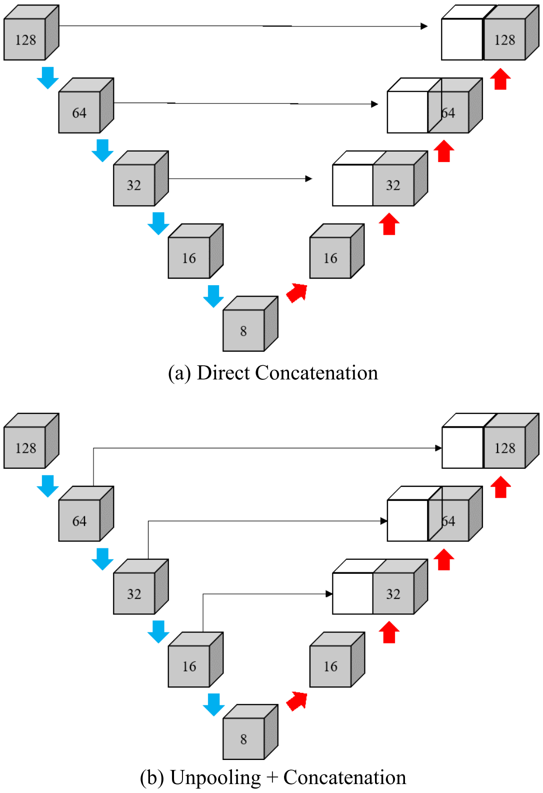

M
Same number of feature maps as 3D-Unet
I
128
128
64
32
16
16
32
64
128
128
128
sigmoid
S1
128
S2
8
4
2
1
dense
sigmoid
C
3*3*3 Convolution twice
3*3*3 Convolution twice + Maxpooling
3*3*3 Convolution twice + Upsampling
1*1*1 Convolution
Unpooling + Concatenation
64 feature maps in each layer

FIGURE 1. The architecture of the network. I and M represent the input image patch and instance memory respectively. S1 is the instance segmentation result, S2 is the semantic segmentation result, and C is the classification score. The input patch size is 128*128*128, and the number of feature maps in each layer is set to be the same as 3D-UNet's, starting at 32 and then going to 64 for every layer in the classification path, with the number on each cube indicating the output size in each layer. All the convolutional layers are followed by layers of batch normalization and ReLu activation.

feature maps as 3D-UNet for the contracting and expansive path, and 64 for the classification path. Second, we choose the data after max-pooling instead of the data before max-pooling for concatenation. Third, we add a third output, S2, at the end of the expansive path, which provides the result of direct semantic segmentation. The anatomical label prediction is derived from the results of instance segmentation (S1) and semantic segmentation (S2). Details will be covered in the following sections.

A. UPSAMPLING AND CONCATENATION

The whole segmentation network contains a U-shape structure with contracting and expansive paths, which is adapted from 3D-UNet. In our proposed segmentation network, the feature maps are set to be the same as those in 3D-UNet, and upsampling (instead of deconvolution) is used for the expansive path. This is inspired by DeconvNet [14], which reveals that upsampling can significantly reduce memory usage in 2D image segmentation, from 1892MB to 1174MB (about 37%). As for the concatenation, we use another concatenation method that passes the high-resolution feature data in the contracting path to the expansive path, but we pass the feature map after max-pooling instead of before it. The high-resolution feature map passing to the expansive path goes through unpooling before concatenation since we find this to be a good way to decrease the number of redundant parameters, and the data after max-pooling retains more important information. The difference between the concatenation methods is shown in Figure 2. We keep the same instance memory setting proposed by Lessmann et al. [7]. The input patch is a two-channel 128*128*128 3D patch comprised of a CT image and corresponding instance memory. Since not



(a) Direct Concatenation

(b) Unpooling + Concatenation

FIGURE 2. Comparison of two concatenation methods.

all CT images have the same resolution, we resample all the CT images to be $1mm^3$ per voxel so as to keep them identical across training images.

B. START POINT FOR ITERATIVE INSTANCE SEGMENTATION

Since the intensity value of the bone in CT images is greater than that of tissue, we can get a proper starting position by excluding voxels with intensity below 200HU, and then choosing the boundary point, (the lowest x, y, and largest or smallest z, depending on the traversal direction). Doing so avoids redundant iterations in the beginning, and reduces the possibility of interference by voxels that are not bones. Hav- ing found the starting point, depicted in Figure 3, we follow the iterative instance segmentation procedure from Lessmann *et al.* [7]. The traversal route of the sampled patch is first adjusted according to the constant step along the x, y-axis until a large enough volume object is detected in the patch. Once the segmented volume exceeds 1,000 voxels, the next patch center becomes the average center of this segmented volume for the next iteration. This step is repeated until a complete vertebra is at the center of the sampled patch, and the classifier identifies the appearance of a complete vertebra as well. After that, the instance memory is updated. This procedure then continues until all vertebrae are segmented.

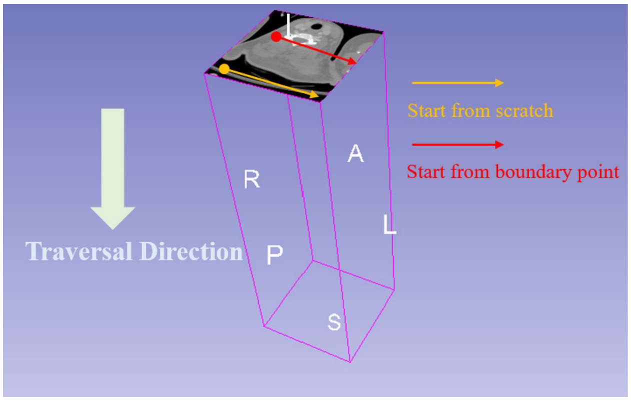


FIGURE 3. The original iterative segmentation sample patch for segmentation from the very beginning of an image. This example will need multiple iterations to get to the first patch center that contains > 1,000 voxels of a vertebral object. Those redundant iterations can be avoided by directly choos- ing the possible patch center by excluding voxels that are definitely not bones (we set a threshold of 200HU).

During the traversal, we must identify the completeness of the vertebrae at the same time, including whether the position of the vertebra is at the center of the patch, and whether the complete vertebra is already present in the instance memory but are not in the instance memory are considered positive samples; other kinds of patches are negative samples.

C. EXTRA OUTPUT FOR ANATOMICAL INFORMATION

In addition to individual vertebral segmentation and completeness identification, we hope to simultaneously obtain anatomical information when a vertebra is segmented. Modifying the architecture outlined in the previous section allows us to free up extra memory space. We can take advantage of the additional memory to deliver an extra output in our network.

Our idea is to modify the segmentation network above via the addition of an extra path in the network (output S2 in Figure 1) for multi-label classification, in order to predict the anatomical label of each voxel. However, if we use only multi-label classification results to directly predict the anatomical labels of vertebrae, the results are not satisfactory because vertebrae are morphologically similar to each other. Thus, misclassification easily occurs near the location between two vertebrae. Hence, we propose combining the output of instance segmentation and multi-label classification to achieve better results. We first leverage the iterative instance segmentation algorithm, which ensures that vertebrae are segmented one by one, so as to avoid interference within the junction of two vertebrae. Whenever we receive a binary segmentation of an individual vertebra from instance segmentation, we extract the multi-label classification result of the region where the binary prediction of this segmented vertebra is positive. Then we are able to calculate the probability of a possible anatomical label for that segmented vertebra. The anatomical label of a segmented vertebra is decided by the label with the highest portion of voxels in that region. Figure 4 provides an example of how anatomical predictions are obtained.

D. TRAINING NETWORK

The original instance segmentation loss from Lessmann et al. combines losses for segmentation and classification. Since we add an extra output in our network, the total training loss needs to have another term for semantic segmentation. We use Cross-Entropy as the loss for multi-label classification. Furthermore, we find that if the segmentation loss value is too high, the convergence speed of Cross-Entropy will be extremely slow, and vice versa, which means the tasks cannot be trained well simultaneously. Thus, we divide the segmentation loss from Lessmann et al. with $\boldsymbol{\alpha}\cdot\triangle\mathbf{p}$ in order to balance the values with two other losses. The definition $\boldsymbol{\lambda}$, $\mathbf{FP_{soft}}$, $\mathbf{FN_{soft}}$, are the same as in [7], where t is the ground truth of classification, $\triangle\mathbf{p}$ is the sum of the patch size voxel, and $\boldsymbol{\alpha}$ is a constant between 0 and 1. Through multiple experiments, we find $\boldsymbol{\alpha} = 0.25 \sim 0.5$ has the best convergence speed for both segmentation tasks. The entire form of loss function is:

$$\mathrm{L} = \frac{\lambda \cdot FP_{soft} + FN_{soft}}{\alpha \cdot \Delta_p} + \left[-tlogp - (1-t)\log(1-p)\right] + CrossEntropy \quad (1)$$

IV. EXPERIMENTAL RESULTS

For performance evaluation, we used xVertSeg [15] dataset to obtain our experimental results. It contains 15 individual spine data points with a lumbar segmentation mask and its corresponding anatomical labels i.e., L1 to L5. There are 10 additional data points without ground truth for testing. In our experiment, different settings for training were chosen

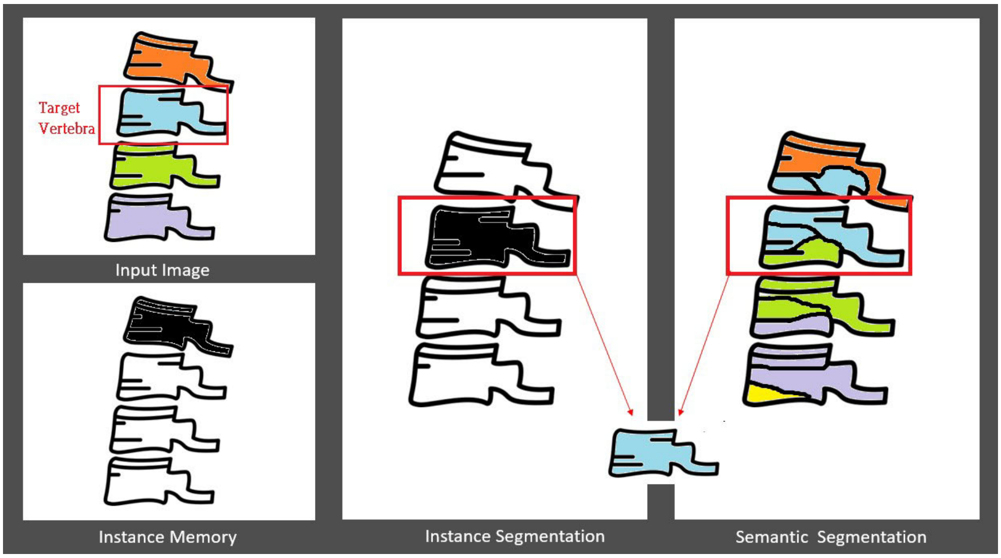


FIGURE 4. The original iterative segmentation sample patch for segmentation from the very beginning of an image. This example will need multiple iterations to get to the first patch center that contains > 1,000 voxels of a vertebral object. Those redundant iterations can be avoided by directly choosing the possible patch center by excluding voxels that are definitely not bones (we set a threshold of 200HU).

for comparison, with the Dice Coefficient serving as the metrics for egmentation performance. The definition of dice coefficient is defined as follow, where |X∩Y| is the number of the voxels that are correctly predicted, |X| and |Y| are the number of the total voxels in the prediction result and ground truth, respectively.

$$\text{Dice Coefficient} = \frac{2|X \cap Y|}{|X| + |Y|} \tag{2}$$

A. MEMORY USAGE COMPARISON AND SEGMENTATION RESULTS

In order to show whether our model addressed the memory consumption problem from Lessmann *et al.* [7], we first tried to reproduce the structure. However, we needed to reduce the feature maps from 64 to 60 in the Lessmann *et al.* [7] network due to our GPU limitations. Therefore, we ran our model with 60 feature maps in the classification network as well. Without the S2 output, our model reduced memory usage by about 17%, from 10649 MB to 8192MB. After adding the S2 output, memory usage was 10167MB, which is still less than the memory usage in Lessmann et al. [7], while providing more information. Moreover, we conducted experiments for300 training epochs; 12 groups were randomly sampled from the dataset for training and 3 were sampled for testing with a learning rate of 0.001 to compare the segmentation performance on each vertebra between the two models. Table 1 features the Dice scores of two architectures under the same experiment. The results show that our model reaches a higher accuracy than that of Lessmann et al. with more efficient memory usage. From Figures 5 and 6, we can see that our model performs even better on details (indicated by the red squares).

TABLE 1. Comparison of Dice scores for segmentation results of each single lumbar vertebra on XvertSeg.

Vertebra	Our Model	Lessman et. al
L5	**92.6%**	88.05%
L4	**89.2%**	72.4%
L3	**86.16%**	81.55%
L2	86.1%	**87.4%**
L1	**88.26%**	88.05%

B. ANATOMICAL CLASSIFICATION

With more memory available, our model also provides accu- rate anatomical classification. Table 2 shows the average se- mantic segmentation prediction for each vertebra. Although there are misclassifications due to adjacent vertebra as de- picted in Figure 4, the majority prediction within vertebrae are correctly labeled, which means if we combine the results from instance segmentation to isolate each vertebra, the accuracy of anatomical prediction actually reaches 100%.

TABLE 2. The probability of each possible label for its real label after combining the results of instance segmentation and multi-label classification.

Vertebra	L5	L4	L3	L2	L1
L5	**0.908**	0.056	0.001	0	0
L4	0.001	**0.921**	0.07	0	0
L3	0	0.013	**0.916**	0.022	0
L2	0	0	0.012	**0.978**	0.005
L1	0	0	0	0.009	**0.985**

C. TEST WITHOUT GROUND TRUTH

The XvertSeg dataset provides 10 sets of test data without ground truth for testing, and we also tried to use all data with ground truth for training and then predict those data without ground truth. As there was no ground truth for calculating the Dice score to evaluate the performance,

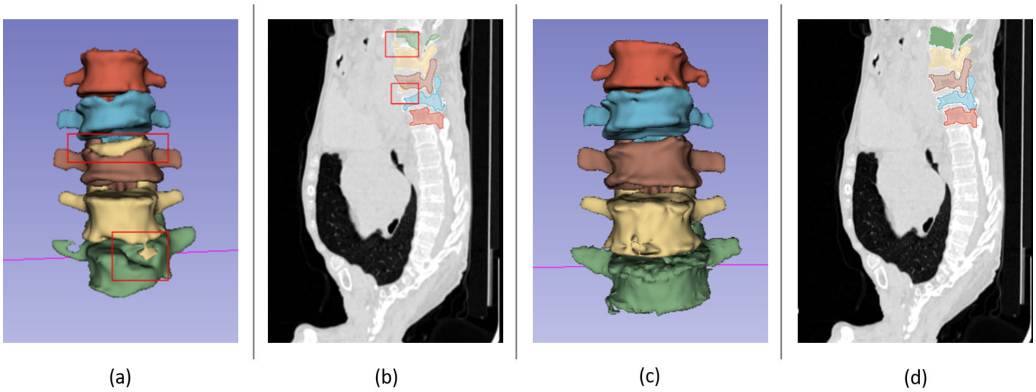


FIGURE 5. (a), (b) are the segmentation results of our model and (c), (d) are results from the Lessmann et al. [7] model of the 12th group in XVertSeg.

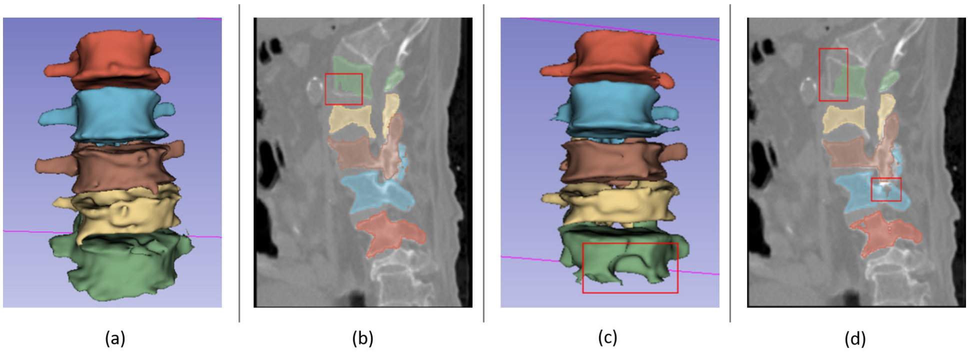


FIGURE 6. (a), (b) are the segmentation results of our model and (c), (d) are results from the Lessmann et al. [7] model of the 15th group in XVertSeg.

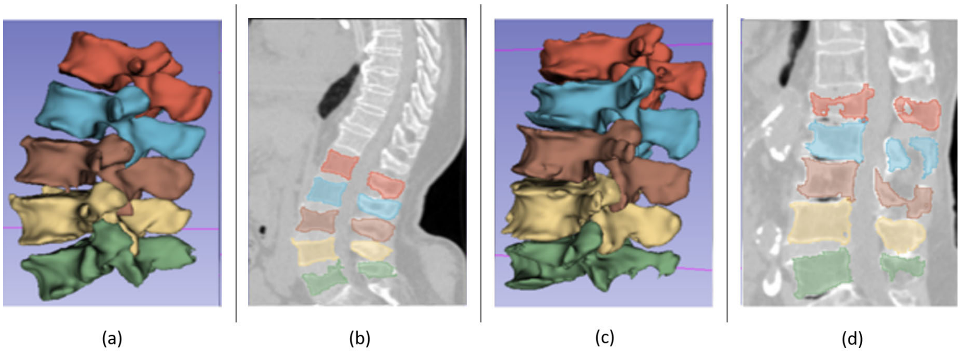


FIGURE 7. (a), (b) are the segmentation results of our model and (c), (d) are results from the Lessmann et al. [7] model of the 15th group in XVertSeg.

and the previous experimental results had already confirmed that our network architecture could effectively process this dataset, we merely visualized our segmentation result in Figure 7.

We also tried to train our model with our internal dataset. Most of the images in this dataset contain more than just lumbar vertebrae. Therefore, we sought to test whether our model is robust enough to achieve good segmentation results on cervical and thoracic vertebrae as well. This dataset contains 7 patients. The number of vertebrae contained in each image ranges from 7 to 21. The z-axis spacing is 0.625mm for all data, but the spacing of the x and y-axes are all different. As shown in Figure 8, we tested the data containing the largest number of vertebrae. The results show the robustness of our model, which is capable of segmenting all kinds of vertebrae smoothly and accurately.

V. APPLICATION

After retrieving the successful segmentation results, we be- gan a clinical collaboration with China Medicine University, Taiwan and Asia University, Taiwan. Our project aims to provide an automated design solution to position pedicle

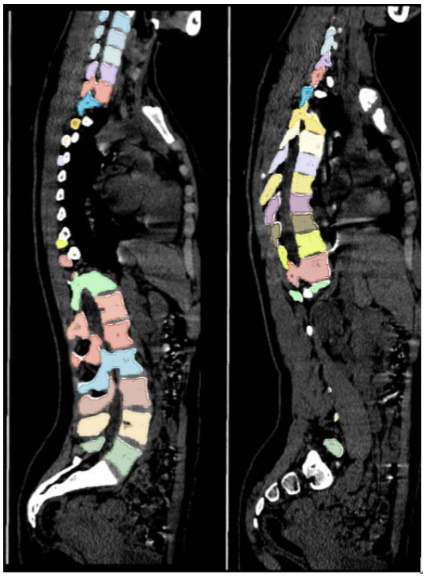


FIGURE 8. Image containing cervical, thoracic, and lumbar vertebrae. Since this patient suffers from severe scoliosis, we need two images to see the segmentation results of all vertebrae in a sagittal view. All the vertebrae are segmented accurately, regardless of the type of vertebrae.

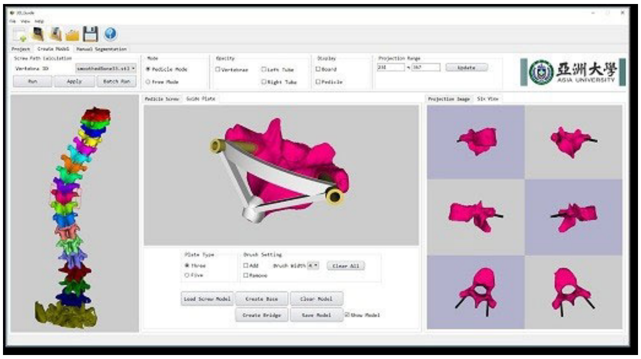


FIGURE 9. The software interface for automated design of the pedicle screw positioning guide plate.

screw guide plates in vertebral correction surgery. The goal is to create a vertebral orthopedic surgical guide plate by automatically finding the best implant position by training our model with an internal CT spine dataset from the affiliated hospital at China Medicine University. Traditionally, finding the proper implant position involves intensive manual work. Although auxiliary tools such as Reina's mechanical arm system already exist, they are still not widely applied due to the high cost. The development of guide plates is generally based on vertebrae, so the first challenge that must be tackled is accurately isolating each vertebra. Using the internal training data collected from the 3D print medical research center at the affiliated hospital in China Medicine University, and the model we proposed in this paper, we have successfully and accurately isolated vertebrae by applying our model, which facilities the design of the pedicle screw positioning guide plate for targeted vertebra. The concept is depicted in Figures 9 and 10.

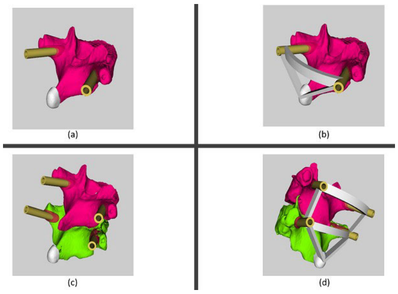


FIGURE 10. The pedicle screw positioning guide plate can only be properly designed with the help of accurate vertebral segmentation. Both (a) and (b) are guide plates designed for only one vertebra, (c) and (d) are guide plates designed for a procedure involving two vertebrae.

VI. CONCLUSION

In this paper, we have proposed an architecture with a triple output that achieves more economical memory usage while maintaining good segmentation results compared to existing techniques. We not achieve only better performance on the XvertSeg dataset for lumbar vertebrae segmentation, but also demonstrate strong performance on general segmentation of different kinds of vertebrae, including cervical and thoracic vertebrae. Moreover, our network provides an extra output to identify the anatomical positions of segmented vertebrae while using less memory for all three outputs than models that can only have two outputs. Those advantages have enabled us to successfully apply our model to a clinical software project. In future work, we expect to collect a wider range of pathological data so that our architecture can try to train different pathological features, including deformities, breakages, and fractures. In addition, we will investigate how to further decrease memory usage. If the memory usage can be reduced by about 800MB, the GPU can be lowered on low-end devices.

REFERENCES

[1] A. Rasoulian, R. Rohling, and P. Abolmaesumi, "Lumbar spine segmentation using a statistical multi-vertebrae anatomical shape+pose model," *IEEE Trans. Med. Imag.*, vol. 32, no. 10, pp. 1890–1900, Oct. 2013.

[2] S. Benameur, M. Mignotte, S. Parent, H. Labelle, W. Skalli, and J. de Guise, "3D/2D registration and segmentation of scoliotic vertebrae using statistical models," *Comput. Med. Imag. Graph.*, vol. 27, no. 5, pp. 321–337, 2003.

[3] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: learning dense volumetric segmentation from sparse annotation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2016, pp. 424–432.

[4] A. A. Darwish, A.-M. Salem, D. Hegazy, and H. M. Ebeid, "Vertebrae segmentation techniques for spinal medical images," *IEEE Trans. Med. Imag.*, vol. 32, no. 10, pp. 1890–1900, Dec. 2013.

[5] R. Janssens, G. Zeng, and G. Zheng, "Fully automatic segmentation of lumbar vertebrae from CT images using cascaded 3D fully convolutional networks," in *Proc. IEEE 15th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2018, pp. 893–897.

[6] R. Korez, B. Ibragimov, B. Likar, F. Pernuš, and T. Vrtovec, "A framework for automated spine and vertebrae interpolation-based detection and model-based segmentation," *IEEE Trans. Med. Imag.*, vol. 34, no. 8, pp. 1649–1662, Aug. 2015.

[7] N. Lessmann, B. V. Ginneken, P. A. D. Jong, and I. Išgum, "Iterative fully convolutional neural networks for automatic vertebra segmentation and identification," *Med. Image Anal.*, vol. 53, pp. 142–155, Apr. 2018.

[8] B. Naegel, "Using mathematical morphology for the anatomical labeling of vertebrae from 3D CT-scan images," *Comput. Med. Imag. Graph.*, vol. 31, no. 3, pp. 141–156, Apr. 2007.

[9] Z. Peng, J. Zhong, W. Wee, and J.-H. Lee, "Automated vertebra detection and segmentation from the whole spine MR images," in *Proc. IEEE Eng. Med. Biol. 27th Annu. Conf.*, vol. 2, no. 3, Jan. 2005, pp. 2527–2530.

[10] P. L. Hoon, B. Ulas, and B. Li, "Introducing Willmore flow into level set segmentation of spinal vertebrae," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 1, pp. 115–122, Jan. 2013.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.

[12] A. Sekuboyina, A. Valentinitsch, J. S. Kirschke, and B. H. Menze, "A localisation-segmentation approach for multi-label annotation of lumbar vertebrae using deep nets," 2017, *arXiv:1703.04347*. [Online]. Available: https://arxiv.org/abs/1703.04347

[13] C. Yen, H.-R. Su, and S.-H. Lai, "Reconstruction of 3D vertebrae and spinal cord models from CT and STIR-MRI images," in *Proc. 2nd IAPR Asian Conf. Pattern Recognit.*, Nov. 2017, pp. 150–154.

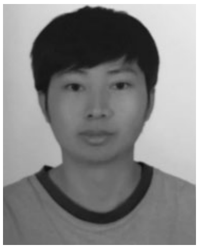
[14] Z. Tan, B. Liu, N. Yu, and S. H. Lai, "PPEDNet: Pyramid pooling encoder-decoder network for real-time semantic segmentation," in *Proc. Int. Conf. Image Graph.*, 2014, pp. 328–339.

[15] *xvertSeg DataSet*. Accessed: Jun. 1, 2017. [Online]. Available: http://lit.fe.uni-lj.si/xVertSeg/database.php

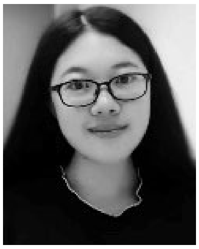
CHENG-HUNG CHUANG received the Ph.D. degree in electrical engineering from National Chung Cheng University, Chiayi, Taiwan, in 2003. From 2003 to 2007, he was a Postdoctoral Fellow with the Institute of Statistical Science, Academia Sinica, Taipei. He joined the Department of Computer Science and Information Engineering, Asia University, Taichung, Taiwan, as an Assistant Professor, in 2007, where he has been an Associate Professor, since 2013. His research interests include image/video processing, optical and biomedical signal processing, computer vision, and pattern recognition.

CHIH-YANG LIN (M'11) received the Ph.D. degree in computer science and information engineering from National Chung Cheng University, Chiayi, Taiwan, in 2006. He was with the Advanced Technology Center, Industrial Technology Research Institute of Taiwan, Hsinchu, Taiwan, from 2007 to 2009. He was a Postdoctoral Fellow with the Institute of Information Science, Academia Sinica, Taipei, Taiwan, in 2009. He joined Asia University, Taichung, Taiwan, in 2010, where he is currently an Assistant Professor and, then, became an Associate Professor, in 2013. He was the Chair of the Department of Bioinformatics and Medical Engineering, Asia University, from August 2014 to January 2017. He is currently an Associate Professor with the Department of Electrical Engineering, Yuan Ze University, Taoyuan, Taiwan. He has authored or coauthored over 100 papers and patents. His research interests include computer vision, machine learning, image processing, and the design of surveillance systems. He received Best Paper Awards from Pacific-Rim Conference on Multimedia (PCM), in 2008, Best Paper Awards, and Excellent Paper Award from Computer Vision, Graphics and Image Processing Conference, in 2009 and 2013. He has served as the Session Chair, the Publication Chair, or Workshop Organizer on many international conferences, including AHFE, ICCE, ACCV, the IEEE Multimedia Big Data, ACM IH&MMSec, APSIPA, and CVGIP.

YUAN-YU TSAI (M'07) received the Ph.D. degree from the Institute of Computer Science, National Chung Hsing University, Taiwan, in 2006. He is currently an Associate Professor with the Department of M-Commerce and Multimedia Applications, Asia University, Taichung, Taiwan. His research interests include computer graphics and information hiding algorithms for three-dimensional models and images. He is a member of the ACM and the IEEE Computer Society.



ZHEN-YOU LIAN received the B.S. and M.S. degrees in computer science and information engineering from Asia University, Taichung, in 2011 and 2013, respectively, where he has been a Research Project Assistant, since 2015, engaged in the 3D printing related research and deep learning methods applied to biomedical imaging. His research interests include image/video processing, medical image processing, computer vision, pattern recognition, and deep learning.



HONG-XIA XIE received the M.S degree in communication and information systems from Fujian Normal University, China, in 2019. She is currently pursuing the Ph.D. degree with the Institute of Electronics, National Chiao Tung University, Taiwan. She has participated in projects, including head–shoulder segmentation and pedestrian detection. Her research interests include object detection and segmentation based on deep learning.

CHIH-CHAO HSU received the B.Sc. degree in computer science from National Tsing-Hua University, Taiwan, in 2016. He is currently pursuing the master's degree with the University of Montreal, Canada. He is a Research Assistant with Yuan Ze University, Taiwan. His research interests include deep learning, medical image processing, and the combination of artificial intelligence and industrial field.

CHUNG-LIN HUANG received the B.S. degree in nuclear engineering from National Tsing Hua University, Hsinchu, Taiwan, in 1977, the M.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1979, and the Ph.D. degree in electrical engineering from the University of Florida, Gainesville, FL, USA, in 1987. From 1987 to 1988, he was with Unisys Co., Orange County, CA, USA, as a Project Engineer. Since August 1988, he has been with the Electrical Engineering Department, National Tsing Hua University, Hsinchu, Taiwan. He is currently a Professor with Asia University, Taiwan. His research interests include image processing, computer vision, and visual communication.