CT-Based Automatic Spine Segmentation Using Patch-Based Deep Learning

Syed Furqan Qadri,1,2 Hongxiang Lin,1 Linlin Shen,2 Mubashir Ahmad,3 Salman Qadri,4 Salabat Khan,2 Maqbool Khan,5,6 Syeda Shamaila Zareen,7 Muhammad Azeem Akbar,8 Md Belal Bin Heyat,9,10,11 and Saqib Qamar12

1Research Center for Healthcare Data Science, Zhejiang Lab, Hangzhou 311121, China
2AI Research Center for Medical Image Analysis and Diagnosis, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong 518060, China
3Department of Computer Science, COMSATS University Islamabad, Abbottabad Campus, Tobe Camp, Abbottabad 22060, Pakistan
4Computer Science Department MNS-University of Agriculture, Multan 60650, Pakistan
5Software Competence Center Hagenberg GmbH, Softwarepark, Hagenberg, Linz, Austria
6Pak-Austria Fachhochschule-Institute of Applied Sciences and Technology, Mang, Haripur, Pakistan
7Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
8Lappeenranta University of Technology, Department of Information Technology, Lappeenranta 53851, Finland
9IoT Research Center, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong 518060, China
10Centre for VLSI and Embedded System Technologies, International Institute of Information Technology, Hyderabad, Telangana 500032, India
11Department of Science and Engineering, Novel Global Community Educational Foundation, Hebersham, NSW 2770, Australia
12Department of Physics, Integrated Science Lab (IceLab), Umea University, Umea 90187, Sweden

Correspondence should be addressed to Syed Furqan Qadri; furqangillani79@gmail.com, Hongxiang Lin; hxlin@zhejianglab.edu.cn, and Linlin Shen; llshen@szu.edu.cn

Received 17 November 2022; Revised 13 January 2023; Accepted 31 January 2023; Published 4 March 2023

Academic Editor: Alexander Hošovský

Copyright © 2023 Syed Furqan Qadri et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

CT vertebral segmentation plays an essential role in various clinical applications, such as computer-assisted surgical interventions, assessment of spinal abnormalities, and vertebral compression fractures. Automatic CT vertebral segmentation is challenging due to the overlapping shadows of thoracoabdominal structures such as the lungs, bony structures such as the ribs, and other issues such as ambiguous object borders, complicated spine architecture, patient variability, and fluctuations in image contrast. Deep learning is an emerging technique for disease diagnosis in the medical field. This study proposes a patch-based deep learning approach to extract the discriminative features from unlabeled data using a stacked sparse autoencoder (SSAE). 2D slices from a CT volume are divided into overlapping patches fed into the model for training. A random under sampling (RUS)-module is applied to balance the training data by selecting a subset of the majority class. SSAE uses pixel intensities alone to learn high-level features to recognize distinctive features from image patches. Each image is subjected to a sliding window operation to express image patches using autoencoder high-level features, which are then fed into a sigmoid layer to classify whether each patch is a vertebra or not. We validate our approach on three diverse publicly available datasets: VerSe, CSI-Seg, and the Lumbar CT dataset. Our proposed method outperformed other models after configuration optimization by achieving 89.9% in precision, 90.2% in recall, 98.9% in accuracy, 90.4% in F-score, 82.6% in intersection over union (IoU), and 90.2% in Dice coefficient (DC). The results of this study demonstrate that our model’s performance consistency using a variety of validation strategies is flexible, fast, and generalizable, making it suited for clinical application.
1. Introduction

The spine plays a key function in mobility and weight transfer in the musculoskeletal system while supporting and sustaining the body and organ structure. It also protects the spinal cord from mechanical shocks and injuries. A number of techniques are used to understand better and quantify the human spine’s biomechanics, including vertebral finite element modeling [1], quantitative imaging [2], spinal alignment analysis [3], and complicated biomechanical models [4]. Biomechanical changes can result in disability and severe discomfort in the short term but can have far worse long-term complications, such as an eightfold higher mortality rate due to osteoporosis. Despite their critical nature, spine diseases are frequently underdiagnosed. This necessitates using a computer-aided approach to detecting such pathologies early and efficiently, allowing for their prevention and effective treatment.

Over the years, the medical imaging community has paid increasing attention to the study of spine image analysis, and vertebrae segmentation [5] is an essential step in comprehending it. Vertebral segmentation has diagnostic significance for detecting and classifying vertebral fractures, estimating the spine curve, and recognizing spine deformities such as kyphosis and scoliosis. Aside from diagnostic purposes, these tasks help with finite element modeling analysis, biomechanical modeling, and surgical planning for metal implantations. Annotating huge structures needs a lot of time, making manual segmentation impractical. A consistent and precise manual delineation is also difficult due to the complicated shape of posterior vertebral components and lower scan resolutions. Various problems exist in automating these tasks, including datasets with highly variable fields of vision, scans in large sizes, scan noise with highly correlated forms of nearby vertebrae, fluctuating scanner settings, and multiple anomalies or pathologies.

Vertebral segmentation traditionally relied on model-based techniques, which fit a shape before the spine and then distort it to conform to its shape. Statistical shape models [6–8], geometric models [9, 10], Markov random fields (MRF) [11, 12], and active contours [13] are all used as prior shape models. Other approaches use intensities such as a priori variational intensity models [14], level sets [15], and automatic vertebrae segmentation from shape models based on landmark frameworks [16]. Machine learning has recently become more popular for the segmentation of vertebrae. A. S. Suzani et al. [17] detected vertebral structures with a multilayer perceptron (MLP) and segmented them using deformable registration. Similarly, Chu et al. [18] identified and located the vertebrae using random forest regression followed by segmentation at the voxel level using random forest classification. R. Korez et al. [19] employed 3D convolutional neural networks (CNNs) to learn vertebral appearances and forecast probability maps that guided the deformable model’s boundaries for vertebral body’s segmentation.

Recent years have seen an increase in the popularity of deep learning for vertebral segmentation, and many published approaches used convolutional [20] and recurrent neural networks instead of explicit modeling of vertebral appearance and shape. The growing popularity of deep learning in vertebral segmentation and greater processing power have prompted researchers to produce promising results. A. Sekuboyina et al. [21] used U-Net to perform patch-based binary segmentation and then denoise the vertebrae masks heat maps with a low resolution. In their other work [22], two different types of neural networks were used to segment lumbar vertebrae. First, a simple multilayer perceptron is trained to regress the lumbar region localization, and then a U-Net is trained to perform multiclass segmentation. Janssens et al. [23] improved this by substituting a CNN for the multilayer perceptron and using two sequential CNNs for lumbar vertebrae multiclass segmentation. Using a two-stage iterative technique, Lessmann et al. [24] first identified and segmented lower-resolution vertebrae one after the other and then refined the masks with a low resolution using a second CNN. These findings led to the development of a single-phase fully convolutional network by Lessmann et al. [25] that iteratively regressed and segmented the vertebral anatomical label. A maximum likelihood technique is used to adjust the vertebral labels after the complete scan has been segmented. A different strategy is proposed by Payer et al. [26], using a coarse-to-fine technique including three steps: vertebra labeling, spine localization, and vertebrae segmentation, all of which rely on purposely built fully convolutional networks.

One limitation of the methodologies described previously was complicated network modeling. As an alternative to the approaches stated previously, it is argued to further improve vertebrae segmentation outcomes. Contextual high-level features that may capture more discriminative sample feature representation are exploited using a regression model to segment the vertebrae in CT images. With the fast advancement of deep learning, an increasing number of deep learning techniques specifically stacked sparse autoencoders (SSAEs), have been applied to medical images since Hinton and Salakhutdinov [27], and they developed the first deep autoencoder network. Shin et al. [28] used stacked autoencoders in MRI to identify organs in medical imaging to show the potential of the deep learning technique to be applied in medical image analysis. Many complex medical imaging problems have been addressed using it. For example, CAD system to classify gastric cancer from the breath samples using SSAE [29], stroke lesions segmentation using sparse autoencoder (SAE) layers, followed by support vector machine (SVM) classifier [30], SSAE-based modeling for the vertebrae segmentation [31], deformable prostate segmentation method [32], nuclei detection from histopathological images of breast cancer [33], an automatic vertebrae localization and identification by SSAE and structured regression forest [34], an automated nucleus detection [35], and Parkinson’s disease diagnosis.
modeling also based on stacked sparse autoencoder framework [36].

Based on the previous work [37], we presented CT-based automatic spine segmentation utilizing patch-based deep learning and new PE and RUS-modules were proposed. The PE-module is used to extract overlapping image patches and label them with a certain pixel ratio, while the RUS-module is employed to address the class imbalance problem. We tested the generalizability and flexibility of our model on three publically available datasets (VerSe, CSI-Seg, and the Lumbar dataset) to show that it is well-suited for clinical application, which was not done in preliminary work [37]. The proposed work is a fully connected framework for high-level feature extraction using SSAE instead of convolutional neural network feature-based representation, which utilizes convolutional and subsampling techniques to extract features from a cluster of locally connected neurons via their local receptive fields. SSAE is a two-stage architecture with an encoder-decoder in which “encoder” encodes pixel intensities via low-dimensional features, while the “decoder” architecture uses low-dimensional attributes to reconstruct the original pixel intensities. SSAE is a fully connected network that uses a single global weight matrix to represent features, while CNN is a model of partial connections that emphasizes the significance of locality. Notwithstanding this, SSAE extracts high-level features from the bottom up in an unsupervised manner. These efficient representations cause precise image patch classification, leading to more robust CT vertebrae segmentation. Therefore, we choose to employ SSAE rather than convolutional neural networks in this work. The major contributions of this study are as follows:

(i) PE-module is applied to extract overlapping patches from input slices of CT images. Splitting slices into patches enhances localization because the trained network is built to focus more on patches’ local details.

(ii) To classify vertebrae patches effectively (reducing false negatives), RUS-module is used to address the class imbalance problem by sampling an equivalent number of patches (vertebrae and nonvertebrae patches) in the training phase.

(iii) The pretraining step, an unsupervised feature learning module based on the SSAE framework, is used to learn high-level features from a large number of unlabeled image patches, while, in the fine-tuning step, these most discriminative sets of features are then subsequently fed to a sigmoid layer to classify each patch as vertebra or not.

(iv) We designed a five-layer SSAE architecture: one input layer, three hidden layers, and one output layer (sigmoid layer). We validated our approach on three diverse publicly available datasets (VerSe, CSI-Seg, and the Lumbar dataset) to demonstrate that our approach is flexible, fast, and generalizable, making it suited for clinical application.

The remaining paper is organized as follows: Section 2 briefly describes our proposed method, composed of six modules. Section 3 describes the experimentation, including datasets, performance evaluation, model training, and architecture optimization. Section 4 reports the results and discussion in detail. Finally, Section 5 concludes this work and offers suggestions for future work.

2. Methods

Figure 1 illustrates that the proposed method consists of the following steps: (i) preprocessing, (ii) patch extraction module (PE-module), (iii) random under sampling (RUS)-module, (iv) $L_2$ regularized SSAE, (v) sigmoid regression, and (vi) postprocessing.

2.1. Preprocessing. The main aim of the preprocessing step is to increase the discrimination between vertebrae and other tissues by identifying bone pixels and removing noise from the image. We applied a threshold approach to eliminate noise artifacts from the whole CT scan. For this reason, influences from the tissues around the vertebra, noise, and imaging artifacts are reduced by setting the intensity to zero outside the bone intensity range of 100 HU (Hounsfield unit) and 1,500 HU automatically. Input spine CT scans are volumetric and must be processed slice by slice. Because the pixel intensities of vertebrae in CT scans are higher than those of other tissues, the applied threshold differentiates them from soft tissues. However, vertebrae have similar intensities to other bones (such as the ribs), so we trained a deep learning model to discriminate between vertebrae and other bony structures in CT scans. Then, a Gaussian filter with a sigma value of 2 is applied to the CT images to smooth them out and ensure that the image gradients are well-defined and there are no intensity singularities. The data are normalized to a range of 0-1.

2.2. PE-Module. PE-module is applied to extract $n \times n$ size overlapping patches from the input CT images by taking a certain pixel stride (Figure 2). A $32 \times 32$ patch-sized image contains 1024 pixels in total. The patch is labeled 1 (vertebra) if the total pixels inside it are equal to or greater than 60%; otherwise, it is labeled 0 (background). PE-module uses a specific pixel stride to construct overlapping patches from the 2D slices for the sliding window. PE-module is an image partition module employed successfully on a patch-based deep learning model for network training to improve classification accuracy.

2.3. RUS-Module. After the PE-module, the number of image patches was unbalanced because the area occupied by the spine in the CT scans was so small compared to the background. The classifier may be biased in the background because most patches are labeled as 0. A high sensitivity rate is preferable from a medical perspective, but on a practical
level, a high false negative rate is unsuitable [38]. It is necessary to strike a balance between the size of the positive and negative training image patches to solve this dilemma.

The RUS-module is applied to balance the training data by selecting a subset of the majority class (background patches). This module deletes random image patches from the...
majority class (Figure 3). Expressing class B as the majority and class F as the minority, the ratio of the size of the minority and majority classes is defined as $r$, and we performed RUS on B to achieve a balanced ratio of $r$. The balanced $r$ ratios after RUS-module were $r$ (nonvertebrae patch) = 0.6 and $r$ (vertebrae patch) = 0.4 that were unbalanced $r$ ratios (nonvertebrae patch) = 0.94 and (vertebrae patch) = 0.06 before the RUS-module. This improves the network’s accuracy and convergence rate during the model training [39]. However, the testing stage does not include a balanced class of image patches.

2.4. $L_2$ Regularized SSAE. A fundamental component of SSAE is an autoencoder (AE) composed of three layers: an input, a hidden layer, and an output. The nodes in an AE’s different layers are all fully connected. A multilayer neural network can be formed by stacking multi-AEs. We improve the three-layer SSAE network by stacking three AEs (Figure 4). We pretrained the model using the greedy layer-wise SSAE approach. Due to the unsupervised nature of pre-training, the label (ground truth) information is not used. We consider that $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ expresses the autoencoder input vector, $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ expresses the reconstructed representation vector of $\mathbf{x}$, and $\mathbf{z} = (z_1, z_2, \ldots, z_n)$ expresses $k$ hidden node activation vector. The autoencoder uses the weights $w_1$ and bias $b_1$ for encoding the input vector $\mathbf{x}$ to $\hat{y} = f(w_1\mathbf{x} + b_1)$ because it uses the intermediate hidden layer to rebuild input features on the output layer. In the hidden layer, activation vector $\mathbf{y}$ is decoded the $\mathbf{z}$ output using decoding weights $w_2$ and bias $b_2$ and then $y$ maps the hidden layer latent representation to the output $z$ by $y = f(w_2y + b_2)$. We constructed an $L_2$ regularized sparse autoencoder using the following cost function:
2.5. Sigmoid Regression. As SSAE is an unsupervised learning approach, each network layer has been trained using unlabeled data. A feature vector was used to generate the input reconstruction. The classifier uses these feature vectors to classify the input data of the stacked sparse autoencoder. We used a sigmoid regression layer to discriminate between vertebrae and nonvertebrae patches (Figure 4). MLP and SVM are other classifiers that can be used instead of the sigmoid layer. The MLP is a feed-forward neural network with several layers and a large number of nodes in each layer that gets stuck in local minima due to the over-fitting problem. In contrast, SVM classifiers determine whether a pixel is part of the target or background class based on its posterior probability value, but it takes a lot of generalization to produce a probability image by reconstructing the score vector. However, the sigmoid layer enables the joint to optimize the entire deep framework via fine-tuning. Sigmoid regression is a binary classification technique for supervised learning. The output probabilities calculate each class label’s likelihood based on the input data. The sigmoid regression model’s coefficient vector gets optimization by reducing the cost function.

\[
\sigma(x) = \frac{1}{1 + e^{-x}},
\]

where \(\sigma\) is the output sigmoid function and \(x\) is the input. At the stage of supervised learning, the pretrained SSAE and sigmoid layer are combined into a single model for classification. Using the scaled gradient descent approach [40], each iteration simultaneously updates the weights of all SSAE layers and all sigmoid layer parameters to fine-tune the whole model.

2.6. Postprocessing. Following training, the trained model is validated using unseen test patches. Our study addresses two-class classification issues where the patch labels are 0 and 1, with 1 denoting vertebrae patches and 0 representing nonvertebrae patches, respectively. The same preprocessing is applied to the CT scans used for testing. Input image patches are given to the trained model, which returns a value between 0 and 1, which can be analyzed as the probability that an image patch belongs to a vertebra or not. The segmented binary image is created by reconstructing the predicted image patches based on these results. Due to the high contrast between the vertebra, rib, and other skeletal structural tissues, some background pixels are misclassified as vertebrae, while some vertebrae pixels were missed from the foreground. For this reason, morphological operations [41] were applied to the binary predicted image in the postprocessing step to eliminate the outliers.
3. Experiments

3.1. Datasets. Three publicly available datasets of CT spine images were used to evaluate the proposed automated method for vertebral segmentation. Reference segmentation ground truths for three of these datasets are publicly available. Figure 5 shows examples of images from the datasets.

3.1.1. Dataset 1. The University of California-Irvine School of Medicine’s Department of Radiological Sciences acquired CT scans using multidetector CT scanners from Philips and Siemens. At a trauma center, thoracolumbar spine CT scans [42] were taken as part of the daily routine without intravenous contrast from 15 adults aged 16 to 35 years old. The slice thickness is 1 mm, and the in-plane resolution is 0.312 to 0.336 mm. Each scan included manual segmentation of all 12 thoracics and 5 lumbar vertebrae, for a total of 180 thoracic and 75 lumbar vertebrae across 15 subjects, and served as references for ground truth. We used 5 scans to train the model and 10 scans (120 thoracic and 50 lumbar) to test it.

3.1.2. Dataset 2. The CT dataset for the lumbar (T1–T5) spine comprises 10 scans and associated manual reference ground truth of the 50 lumbar vertebrae [10]. In-plane voxel sizes ranged from 0.28 to 0.79 mm, and slice thicknesses ranged from 0.72 to 1.53 mm. Each of the lumbar vertebrae was manually segmented to create a binary mask. These scans were utilized as a training dataset.

3.1.3. Dataset 3. The VerSe [43, 44] dataset was acquired at multiple locations utilizing CT scanners from four main vendors (Phillips, Siemens, Toshiba, and GE). In terms of field-of-view (FoV), findings, and scan parameters, the data were carefully arranged to match a clinical distribution. Data were acquired from patients with an average age of 59 (±17) years. It comprises a range field of views (including cervicothoracolumbar, thoracolumbar, and cervical scans) and a combination of sagittal and isotropic reformations and fractures of the vertebrae, foreign materials, and metallic implants. Manual ground truths for the cervical, thoracic, and lumbar vertebrae are included in the data. Experiments used 25 thoracolumbar (T1–T12, L1–L5) scans for the training data.
3.2. Performance Evaluation. Evaluation metrics [45] are used to compare the performance of vertebrae segmentation with other existing approaches. In medical image analysis, these metrics are widely used and well-known. In this paper, precision, recall, accuracy, $F$-score, intersection over union (IoU) [46], and Dice coefficient (DC) [47] are quantitative assessment measures for segmentation performance evaluation [48, 49]. We evaluated true positive (TP), false positive (FP), true negative (TN), and false negative (FN) by comparing the ground truth images with predicted segmented images.

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$F - \text{score} = \frac{2 \times TP}{2 \times TP + FP + FN}$$

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN}$$

$$\text{DC} = \frac{2|A \cap B|}{|A| + |B|} = \frac{2TP}{2TP + FP + FN}$$

3.3. Model Training. For training the model, a total of 805,120 image patches were extracted, of which 483,072 were nonvertebrae patches and 322,048 were vertebrae patches. We randomly selected 644,096 (80%) image patches for training and 161,024 for validation. For efficient training, the mini-batch size is set to 64, and training patches are split into 10,064 mini-batches, while validation patches are split into 2,516 mini-batches. The proposed approach has five network layers: a 1,024-neuron input layer, three hidden layers, each having 200 hidden neurons, and one sigmoid layer has two neurons, one for each output class. This algorithm was implemented in MATLAB 2018a on 32 GB RAM, an NVIDIA GeForce MX250 GPU, and a 1.80 GHz i7 CPU. The algorithm was trained approximately 23.42 hours and 9 seconds for the segmentation time of a 512 $\times$ 512 image.

We first determined the number of epochs required in pretraining of SSAE to ensure the training process convergence in the proposed model. Figures 6(a)–6(c) show the training patch-based learning curve for weights between the input layer and the hidden layers (pretraining learning curves of 3-hidden layers). The mean square error (MSE) between the original input and the reconstructed input from the autoencoder-decoder was computed and plotted. We conducted our studies with a variety of empirical numbers of hidden nodes. These observations reveal that learning processes converge after 300 epochs in different hidden node settings. We chose 500 epochs in the experiment pretraining to ensure SSAE convergence. Figure 6(d) shows the fine-tuning model learning curve for a number of epochs after pretraining. We found the best fit curve for our model training with an MSE of 0.025 for training and 0.029 for validation. Prior to 2,000 epochs, the learning curve rapidly diverges and then stabilizes after 4,000 epochs, and we chose 5,000 epochs in the model fine-tuning.

The five-layered SSAE is based on a visualization model [50] to show the feature presentations of the first, second, and third hidden layers in Figure 7. There are 200 nodes in the first hidden layer, representing the learned feature representation of the vertebrae and other structures, while the second and third hidden layers (200 nodes) represent more high-level feature learning from image patches. Weights between hidden nodes and pixels in the original image are represented by squares. In the weight matrix, white pixels represent positive values, while gray pixels represent negative values in the weight matrix.

3.4. Architecture Optimization. Next, we started optimizing the architecture of the proposed model. A grid search was used to optimize the number of hidden layers and nodes on each layer of SSAE.

Until now, there has been no theory to determine the optimal SSAE architecture for a particular application. Therefore, we conducted the experiments using a variety of empirical values for the number of hidden layers and nodes. SSAE’s high-level feature representation is determined by the number of hidden nodes. Hence, we chose empirical values (100, 200, 300, 400, and 500) for the number of hidden nodes and empirical values [1–4] for the number of hidden layers. The sparsity coefficients were set to sparsity $L_2$ regularization $\lambda$ 0.10, sparsity constraint $\beta$ 0.20, and target activation $\rho$ 0.05. For every possible combination of hidden layers and nodes, the performance metrics were calculated, and the results are shown in Figure 8.

The 3-hidden-layer architecture with 200 nodes produced the best precision results (89.9%). The same design yielded the best recall (90.2%), accuracy (98.8%), IoU (82.6%), and DC (90.2%), respectively. A 2-hidden-layer architecture with 200 hidden nodes produced the best $F$-score results (90.5%). Precision, recall, accuracy, IoU, and DC values from the best-performing architecture resulted in an acceptable $F$-score (90.4%). Different architectures might be chosen depending on the needs of the application. We chose the three-layer architecture for SSAE based on DC requirements, and each layer contains 200 hidden nodes. Figure 9 shows the results in confusion matrices of training and randomly selected test case separately.

4. Results and Discussion

4.1. Results. Our approach is developed based on SSAE for vertebrae segmentation. In experiments, our model achieved 89.9% in precision, 90.2% in recall, 98.8% in accuracy, 90.4% in $F$-score, 82.6% in IoU, and 90.2% in DC. In order to show the effectiveness of $L_2$ regularized SSAE, our method is compared against the state-of-the-art three-layered stacked autoencoder (TSAE) model [33]. If $L_2$ regularization term's
Figure 6: (Pretraining) Learning curves of three hidden layers’ SSAE framework (a–c), all hidden layers have 200 nodes in each layer and 500 epochs were used, and no ground-truths are provided. (Fine-tuning) Model-supervised best-fit learning curve (d) has an MSE of 0.025 for training and MSE of 0.029 for validation, and 5,000 epochs were used.

Figure 7: Three layers of SSAE learned feature visualization. (a) The learned feature representation of the first hidden layer, which includes information of boundary and corner. (b) The second hidden layer’s learned high-level feature representation that indicates the combinations of these first-order representations. (c) A third hidden layer with high-level features. Note that all three hidden layers have the same 200 nodes in each layer.
penalty coefficient $\lambda$ and sparsity regularization coefficient $\beta$, the second and third portions of the cost function in equation (1) are limited to zero, then it turns into a three-layered stacked AE (TSAE) model. Table 1 indicates the means of precision, recall, accuracy, F-score, IoU, and DC of our approach and comparative model of TSAE. The results show the significance of the $L_2$ regularized SSAE of our method that is in superior performance compared to TSAE in all metrics.

The proposed approach is also compared with multiple deep learning models. Table 2 shows our approach is compared with well-known vertebrae segmentation methods including classical U-Net [51], SpineParseNet [48], patch-based deep belief networks model (PaDBNs) [49], TSAE [33], Butterfly FCN model [21], OP-convNet [20], and Mask R-CNN [52], proving that the proposed model outperformed all of the other models in terms of F-score, IoU, and DC. Note that for TSAE results, we used the

<table>
<thead>
<tr>
<th>Precision (%)</th>
<th>Recall (%)</th>
<th>F-Score (%)</th>
<th>DICE Coefficient (DC) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 HL</td>
<td>2 HL</td>
<td>3 HL</td>
<td>4 HL</td>
</tr>
<tr>
<td>100 N</td>
<td>82.6</td>
<td>85.3</td>
<td>79.7</td>
</tr>
<tr>
<td>200 N</td>
<td>86.5</td>
<td>87.1</td>
<td>89.9</td>
</tr>
<tr>
<td>300 N</td>
<td>82.8</td>
<td>88.4</td>
<td>89.6</td>
</tr>
<tr>
<td>400 N</td>
<td>85.1</td>
<td>84.9</td>
<td>85.2</td>
</tr>
<tr>
<td>500 N</td>
<td>82.4</td>
<td>83.8</td>
<td>85.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Accuracy (%)</th>
<th>Intersection over Union (IoU) (%)</th>
<th>DICE Coefficient (DC) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 HL</td>
<td>2 HL</td>
<td>3 HL</td>
</tr>
<tr>
<td>100 N</td>
<td>90.9</td>
<td>89.8</td>
</tr>
<tr>
<td>200 N</td>
<td>95.3</td>
<td>97.1</td>
</tr>
<tr>
<td>300 N</td>
<td>95.9</td>
<td>94.9</td>
</tr>
<tr>
<td>400 N</td>
<td>92.4</td>
<td>96.7</td>
</tr>
<tr>
<td>500 N</td>
<td>90.1</td>
<td>95.2</td>
</tr>
</tbody>
</table>

Figure 8: Proposed method’s performance (precision, recall, accuracy, F-score, IoU, and DC) with diverse architectures. The columns represent the number of hidden layers (HL), and the rows present the number of hidden nodes (N) in each layer, and the top results are shown in bold text.

Figure 9: Confusion matrices of training (a) and randomly selected test case (b).
same experimental sitting except for sparsity regularization $\lambda$ on hidden layers. When compared with the classical U-Net [51], SpineParseNet [48], PaDBNs [49], TSAE [33], Butterfly FCN model [21], OP-convNet [20], and Mask R-CNN [52], our model outperforms them by (9, 10.7, 6.9), (2.8, 5.1, 2.9), (5.5, 6, 4.1), (7.6, 8.7, 5), (4, 5.7, 3.2), (0.2, 0.3, 0.3), and (20.3, 29.5, 21) average in (F-score%, IoU%, and DC%), respectively.

Qualitative examples of segmentation results for 2D axial images are shown in the first, second, third, and fourth rows with the original images, their respective labeled images, predicted segmented images, and overlaid predicted segmented images on original images, respectively (Figure 10). It can be seen that our proposed approach performed well, and the generated results were well-segmented.

4.2. Discussion. To our knowledge, this is the first time an SSAE has been used in patch-based classification for automatic vertebral segmentation using three distinct CT spine datasets. Using deep learning’s excellent data mining advantage on big data, our approach proved the SSAE network’s strong ability to segment vertebrae automatically. Our approach has the potential to function as fully automated CAD software with minimal human intervention and training or analysis; there is no need to choose any handcrafted features. This is an important feature of CAD in today’s fast-paced clinical settings. The SSAE neural network was used to capture the high-level features from overlapping patches in unsupervised learning. In our method, vertebrae and nonvertebrae patches were effectively classified by these high-level features.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Precision (%)</th>
<th>Recall (%)</th>
<th>Accuracy (%)</th>
<th>F-score (%)</th>
<th>IoU (%)</th>
<th>DC (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSAE</td>
<td>87.3</td>
<td>86.9</td>
<td>95.7</td>
<td>82.8</td>
<td>77.8</td>
<td>85.2</td>
</tr>
<tr>
<td>Our method</td>
<td><strong>89.9</strong></td>
<td><strong>90.2</strong></td>
<td><strong>89.9</strong></td>
<td><strong>90.4</strong></td>
<td><strong>82.6</strong></td>
<td><strong>90.2</strong></td>
</tr>
</tbody>
</table>

Bold values represent that our method has better results than others.

<table>
<thead>
<tr>
<th>Methods</th>
<th>F-score (%)</th>
<th>IoU (%)</th>
<th>DC (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classical U-Net [51]</td>
<td>81.4</td>
<td>71.9</td>
<td>83.7</td>
</tr>
<tr>
<td>SpineParseNet [48]</td>
<td>87.6</td>
<td>77.5</td>
<td>87.3</td>
</tr>
<tr>
<td>PaDBN model [49]</td>
<td>84.9</td>
<td>75.6</td>
<td>86.1</td>
</tr>
<tr>
<td>TSAE [33]</td>
<td>82.8</td>
<td>73.9</td>
<td>85.2</td>
</tr>
<tr>
<td>Butterfly FCN model [21]</td>
<td>86.4</td>
<td>76.9</td>
<td>87.0</td>
</tr>
<tr>
<td>OP-convNet [20]</td>
<td>90.2</td>
<td>82.3</td>
<td>89.9</td>
</tr>
<tr>
<td>Mask R-CNN [52]</td>
<td>70.1</td>
<td>53.1</td>
<td>69.2</td>
</tr>
<tr>
<td>Our method</td>
<td><strong>90.4</strong></td>
<td><strong>82.6</strong></td>
<td><strong>90.2</strong></td>
</tr>
</tbody>
</table>

Bold values represent that our method has better results than others.

Figure 10: Examples of visual demonstration of segmentation results: (a) axial plane images; (b) ground truth; (c) our predicted segmented images; (d) prediction images overlaid on original images.
sigmoid regression layer was then used to incorporate these high-level features to improve classification accuracy. Our proposed approach is reliable, robust, and precise. In terms of clinical applications, the developed approach has a high level of overall performance.

SSAE is a neural network, so the convergence of the training procedure was critical to the model’s classification of image patches. In SSAE, a premature network might be the result of insufficient training epochs that cause a lack of optimal performance. Therefore, it is required to conduct a convergence test to determine the correct number of epochs in deep learning. In our experiment, we used 500 epochs for pretraining. This setting ensured the training’s convergence and avoided time wastage. The architecture of neural networks is another critical consideration. It has already been stated that there are no general criteria for designing a neural network’s architecture. Indeed, the optimum neural network architecture is decided by the intricacy of the data that is being used. An early indication of SSAE architecture design could be provided by optimization experiments in our situation. We also found that sparse regularization was necessary during training to build deep feature representations that positively impacted the fine-tuning phase. During training, sparsity pushes the filters to capture more detailed features from image patches. The performance comparison indicated our approach’s effectiveness and its superior capability when compared to other well-known models.

Our proposed approach performs well in classifying the test patches into vertebrae or nonvertebrae patches and then segmenting the vertebrae from the reconstruction of image patches. Each spinal level has its own set of vertebral patterns. Significant morphologic variations can be seen between two vertebrae separated by a wide spatial distance within the spinal column, such as the upper thoracic vertebrae and the lower lumbar vertebrae. It is therefore challenging to achieve accurate segmentation of all the vertebrae. Our proposed model has some limitations with segmentation in the upper thoracic vertebrae (T1–T3) due to the existence of rib structure, and the L5 vertebrae also obtained lower DC than other vertebrae. Figure 11 illustrates the results of poor segmentation where high false negatives represent the vertebrae regions which are not detected by the model, whereas false positives are the background regions that have been segmented wrongly as vertebrae.

5. Conclusion

This study presented a stacked sparse autoencoder framework for automated vertebrae segmentation using publicly available three distinct CT spine datasets (VerSe, CSI-seg, and the Lumbar dataset). We used 2D image slices to extract overlapped patches for model training. A high-level feature representation of pixel intensity is captured in an unsupervised fashion using the proposed model from image patches. A sigmoid layer efficiently classifies vertebrae and nonvertebrae patches using these high-level features. Our approach performed optimally after setting main parameters such as the number of hidden layers, dimension of hidden nodes, and epochs. Sparsity constraints on hidden layers are also demonstrated to be efficient. It was noticed that the training using sparsity regularization is necessary to build feature representations that positively influence the final supervised tuning phase. The scheme of distinguishing vertebrae regions using image patches rather than individual pixels also decreases the rate of false positives. The method demonstrates significant potential for resolving issues caused by morphological variations of vertebrae. When
compared to other state-of-the-art vertebrae segmentation methods, our approach outperformed them in terms of segmentation accuracy. We carried out further experiments that enabled us to identify our method’s limitations, specifically in fractured vertebrae. Future work will improve our approach by developing a more discriminative deep neural network design to make our method more robust in these cases.

Data Availability
The data supporting the findings of this study are available at (Dataset 1 and Dataset 2) https://spineweb.digitalimaginggroup.ca/Index.php?n=Main.Datasets and (Dataset 3) https://osf.io/t98fz/.

Conflicts of Interest
The authors declare that there are no conflicts of interest.

Acknowledgments
This work was supported in part by the National Natural Science Foundation of China under grant numbers 82261138629 and 91959108 and Shenzhen Municipal Science and Technology Innovation Council under grant number JCYJ20220531101412030.

References
23. R. Janssens, G. Zeng, and G. Zheng, "Fully automatic segmenta-
tion of lumbar vertebrae from CT images using cas-
caded 3D fully convolutional networks," in Proceedings of the 2018
IEEE 15th International Symposium on Biomedical Imag-
ing, pp. 893–897, ISBI 2018, Washington, DC, USA,
April 2018.

24. N. Lessmann, B. van Ginneken, and I. Isgum, "Iterative con-
volutional neural networks for automatic vertebra iden-
tification and segmentation in CT images," Medical Imaging

25. N. Lessmann, B. van Ginneken, P. A. de Jong, and I. Isgum,
"Iteratively fully convolutional neural networks for automatic
vertebra segmentation and identification," Medical Image

26. C. Payer, D. Štern, H. Bischof, and M. Urschler, "Coarse to
fine vertebrae localization and segmentation with SpatialCon-
figuration-net and U-net," in Proceedings of the 15th Interna-
tional Joint Conference on Computer Vision Imag-

27. G. E. Hinton and R. R. Salakhutdinov, "Reducing the di-
mensionality of data with neural networks," Science, vol. 313,

28. H.-C. Shin, M. R. Orton, D. J. Collins, S. J. Doran, and M. O. Leach,
"Stacked autoencoders for unsupervised feature learning and mul-
tiple organ detection in a pilot study using 4D patient data," IEEE
Transactions on Pattern Analysis and Machine Intellig-

29. M. A. Aslam, C. Xue, Y. Chen et al., "Breath analysis based
early gastric cancer classification from deep stacked sparse
autoencoder neural network," Scientific Reports, vol. 11, no. 1,
pp. 1–12, 2021.

30. G. B. Praveen, A. Agrawal, P. Sundaram, and S. Sardesai,
"Ischemic stroke lesion segmentation using stacked sparse
autoencoder," Computers in Biology and Medicine, vol. 99,

31. S. F. Qadri, Z. Zhao, D. Ai, M. Ahmad, and Y. Wang,
"Vertebrae segmentation via stacked sparse autoencoder from
computed tomography images," in Proceedings of the Eleventh
International Conference on Digital Image Processing,
Guangzhou, China, August 2019.

32. Y. Guo, Y. Gao, and D. Shen, "Deformable MR prostate
segmentation via deep feature learning and sparse patch
matching," IEEE Transactions on Medical Imaging, vol. 35,
no. 4, pp. 1077–1089, 2016.

33. J. Xu, L. Xiang, Q. Liu et al., "Stacked sparse autoencoder
(SSAE) for nuclei detection on breast cancer histopathology
images," IEEE Transactions on Medical Imaging, vol. 35, no. 1,

34. X. Wang, S. Zhai, and Y. Niu, "Automatic vertebrae locali-
zation and identification by combining deep SSAE contextual
features and structured regression forest," Journal of Digital

35. S. Li, H. Jiang, J. Bai, Y. Liu, and Y. d. Yao, "Stacked sparse
autoencoder and case-based postprocessing method for nu-

36. S. Li, H. Lei, F. Zhou, J. Gardezi, and B. Lei, "Longitudinal and
multi-modal data learning for Parkinson's disease diagnosis
via stacked sparse auto-encoder," in Proceedings of the 2019
IEEE 16th International Symposium on Biomedical Imag-

37. S. F. Qadri, L. Shen, M. Ahmad, S. Qadri, S. S. Zareen, and
M. A. Akbar, "SVseg: stacked sparse autoencoder-based patch
classification modeling for vertebrae segmentation," Mathema-

38. J. Jantzen, J. Norup, G. Dounias, and B. Bjerregaard, "Pap-
smear benchmark data for pattern classification," in Pro-
cedings of the Nature inspired Smart Information Systems
(NISIS), pp. 1–9, Albufeira, Portugal, 2005.

neural networks for computer-aided detection: CNN archi-
tectures, dataset characteristics and transfer learning," IEEE
Transactions on Medical Imaging, vol. 35, no. 5, pp. 1285–
1298, 2016.

40. M. F. Moller, "A scaled conjugate gradient algorithm for fast
supervised learning," Neural Networks, vol. 6, no. 4,

41. Y. Kang, K. Engelke, and W. A. Kalender, "A new accurate
and precise 3-D segmentation method for skeletal structures
in volumetric CT data," IEEE Transactions on Medical Im-

42. J. Yao, J. E. Burns, D. Forsberg et al., "A multi-center mile-
stone study of clinical vertebral CT segmentation," Com-
puterized Medical Imaging and Graphics, vol. 49, pp. 16–28,
2016.

43. M. T. Löfler, A. Sekuboyina, A. Jacob et al., "A vertebral
segmentation dataset with fracture grading," Radiology: Ar-
tificial Intelligence, vol. 2, no. 4, Article ID 190138, 2020.

44. H. Liebl, D. Schinz, A. Sekuboyina et al., "A computed to-
mography vertebral segmentation dataset with anatomical
variations and multi-vendor scanner data," Scientific Data,

45. A. A. Taha and A. Hanbury, "Metrics for evaluating 3D
medical image segmentation: analysis, selection, and tool," 

46. P. Jaccard, "The distribution of the flora in the alpine zone. 1.,

47. L. R. Dice, "Measures of the amount of ecologic association

48. S. Pang, C. Pang, L. Zhao et al., "SpineParseNet: spine parsing
for volumetric MR image by a two-stage segmentation
framework with semantic image representation," IEEE
Transactions on Medical Imaging, vol. 40, no. 1, pp. 262–273,
2021.

49. S. F. Qadri, "Automatic deep feature learning via patch-based
deep belief network for vertebrae segmentation in CT images," 

50. H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional
deep belief networks for scalable unsupervised learning of
hierarchical representations," in Proceedings of the 26th An-
nual International Conference on Machine Learning,

51. O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolu-
tional networks for biomedical image segmentation," in Pro-
cedings of the International Conference on Medical Image
Computing and Computer-Assisted Intervention, pp. 234–241,
Munich, Germany, October 2015.

52. R. Wang, J. H. Yi Voon, D. Ma, S. Dabiri, K. Popuri, and
M. F. Beg, "Vertebra segmentation for clinical CT images
1165, 2021.