



# Vertebra Fracture Classification from 3D CT Lumbar Spine Segmentation Masks Using a Convolutional Neural Network

Charmae B. Antonio, Louise Gillian C. Bautista, Alfonso B. Labao, and Prospero C. Naval Jr. (✉)

Computer Vision and Machine Intelligence Group,  
Department of Computer Science, College of Engineering,  
University of the Philippines Diliman, Quezon City, Philippines  
pcnaval@dcs.upd.edu.ph

**Abstract.** Accurate and efficient identification of vertebra fractures in spinal images is of utmost importance in improving clinical tasks such as diagnosis, surgical planning, and post-operative assessment. Previous methods that tackle the problem of vertebra fracture identification rely on quantitative morphometry methods. Standard six-point morphometry involves manual identification of the vertebral bodies' corners and placement of points on identified corners. This task is time-consuming and requires effort from experts and technicians and prone to subjective errors in visual estimation in spinal images. In this paper, we propose an automated method to detect and classify vertebra fractures from 3D CT lumbar spine images. Fifteen 3D CT images with accompanying fracture labels for each of the five lumbar vertebra from the xVertSeg Challenge were utilized as data set. Each vertebra from the 3D image is processed into  $100 \times 50$  2D 3-channel images composed of three grayscale images. The three grayscale images represent the vertebral slices in the sagittal, coronal, and transverse anatomical planes. These  $100 \times 50$  2D images are fed into the 152 layer Residual Network. A total of 13,400 images were generated from the data pre-processing stage. 12,700 of which having varying classifications were used as training data, and 100 images for each of the seven vertebra fracture classifications were used as testing data. The network achieved 93.29% testing accuracy.

**Keywords:** Vertebra fracture detection  
Convolutional neural network · ResNet

## 1 Introduction

Vertebra fractures are categorized into three major patterns, namely: flexion, extension, and rotation. This work focuses on flexion fracture patterns, which are further categorized into two: the compression fracture and the axial burst fracture. A compression fracture is characterized by a reduction in height of the

anterior portion of the vertebra, while an axial burst fracture is characterized by reduction of both anterior and posterior portions. These reductions in height are caused either by high-energy trauma such as various physical accidents, or by low-impact activities such as reaching or twisting. Furthermore, diseases or physiological abnormalities that may be inferred from having vertebra fractures are osteoporosis, tumors, and various conditions that weaken the bone.

A precise vertebra fracture identification method in spinal imaging is in high demand due to its importance in orthopedics, neurology, oncology, and many other medical fields. A common method in identifying vertebra fractures is the semi-quantitative (SQ) method proposed by Genant et al. [1]. This method introduced specific morphological cases and grades of vertebral body fractures and is accepted as the ground truth for the evaluation of vertebra fractures. By estimating the differences in the anterior, central and posterior heights of the vertebral body in sagittal radiographic images, the SQ method presents the following morphological cases of 3 vertebra fractures.

#### Morphological Cases

- Wedge: characterized by distinctive difference between the anterior and posterior vertebral height.
- Biconcavity: characterized by similar anterior and posterior vertebral height but smaller central vertebral height.
- Crush: characterized by the mean vertebral height lower than the statistical value for that vertebra or for adjacent vertebral bodies.

For morphological grades, mild grade is characterized by a height difference of 20–25%. Moderate grade is characterized by a height difference of 25–40%. Severe grade is characterized by a height difference of more than 40%.

For identification of these cases in CT Scans or MRI scans of bones, the usual recourse is through visual inspection of vertebral bodies, which requires the expertise of radiologists and physicians. The visual inspection of each vertebra is time consuming and prone to subjective errors.

For our proposed procedure, we automate the classification of 3D bone images according to the seven classes of the semi-quantitative method: (1) Normal, (2) Mild Wedge, (3) Moderate Wedge, (4) Severe Wedge, (5) Mild Biconcavity, (6) Moderate Biconcavity and (7) Moderate Crush. Automation is done using a deep residual network from [2], which achieved outstanding benchmark results in the ImageNet classification challenge. The deep residual network makes use of ‘skip-connections’ to allow information to freely pass between network layers. This facilitates training and allows the network to extend to very deep layers. Employing deeper layers allow the network to abstract better and generate features that contain high-value information.

We train the automated procedure using a total of 12,700 images, and then test using 700 images spread equally to each of the seven vertebra fracture classifications. The automated network procedure achieved 93.29% accurate classification.

## 2 Methodology

### 2.1 Data Acquisition

The data set used in this paper were obtained from the xVertSeg challenge, an open online computational challenge in the field of spine medical image analysis. The data set consists of 25 3D computed tomography (CT) lumbar spine images in MetaImage (MHD) format, which requires a pair of .mhd and .raw file per image. The fracture labels for the lumbar vertebra were given in a CSV file. Fracture labels used in the data set follow the semi-quantitative vertebra fracture classification system proposed by Genant et al.

This semi-quantitative system presents a total of 10 classifications, but we exclude the following classifications due to insufficient samples: (1) Severe Biconcavity, (2) Mild Crush, and (3) Severe Crush.

The data set split the 25 images into two: Data1 and Data2. Data1 consists of 15 CT images, and Data2 consists of 10 CT images. The images from Data1 have accompanying segmentation masks and fracture labels, while the images from Data2 do not. For the purpose of vertebra fracture classification, only the Data1 images were used.

The segmentation mask given assigns a corresponding pixel value for each of the five lumbar vertebra: 200 for the L1 vertebra, 210 for the L2 vertebra, 220 for the L3 vertebra, 230 for the L4 vertebra, and 240 for the L5 vertebra.

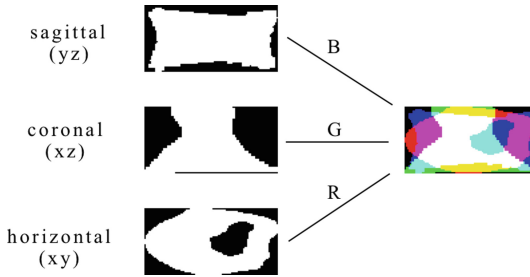
### 2.2 Data Preprocessing

The data to be fed to the deep residual network are  $100 \times 50$  CT images that are created from 3 grayscale images corresponding to the sagittal, coronal, and transverse slices of the vertebra. Preprocessing for each of the 5 lumbar vertebra in a single CT image undergoes the following steps:

1. **Find the areas in the 3D image that are indicative of the vertebral shape.** The 3D image is repeatedly sliced into a 2D image in the sagittal plane for every 2 pixels. Each sliced plane is checked for the 5 lumbar vertebral bodies that are cleanly separable from their spinal parts.
2. **Find the z and y coordinates for the coronal and horizontal planes.** From the first sagittal slice of each vertebra deemed indicative by step 1, the contour of each of the five vertebral bodies are obtained using OpenCV's findContour function. From each of the contour coordinates, the leftmost coordinate marks the area where the coronal planes of the respective vertebra will be sliced from, and the topmost coordinate marks the area where the horizontal planes of the respective vertebra will be sliced from.
3. **Crop the vertebra in three planes.** Step 1 and step 2 results to a voxel coordinate for each of the lumbar vertebra. This voxel coordinate marks the probable corner of the vertebral body. From this coordinate, the coronal, and horizontal planes are each inspected. An algorithm checks if the slices contain enough data or pixels to be indicative of the vertebral shape. Once

the algorithm deems both planes indicative, it proceeds to crop the vertebral shapes using OpenCV’s findContour function. The next iteration moves the voxel coordinate by 2 pixels in each plane.

4. **Create a single BGR image from the three grayscale images.** Step 3 generates sets of three 2D grayscale images, each associated to each other according the same voxel coordinate. These 3 grayscale images corresponds to the sagittal (yz), coronal (xz), and horizontal (xy) planar views. Using OpenCV’s merge function, the 3 grayscale images are combined to form a 3-channel BGR image. A single instance of this 3D image is equivalent to one sample to be used either for training or testing (Fig. 1).

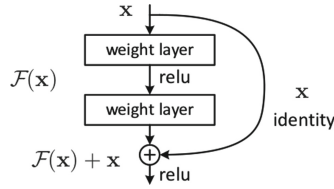


**Fig. 1.** Patches from the sagittal, axial, and horizontal planes are merged into a BGR image

For every vertebral body, the algorithm iterates through steps 2 and 3. The voxel coordinate is moved by 2 pixels to each of the planes, until it reaches the probable end of the vertebral body. The data is augmented through rotation of the 3D image through  $-30^\circ$  to  $30^\circ$ , with an interval of  $5^\circ$ . Each BGR image produced is vertically flipped to double the sample data. In total, this produces 13,400 vertebra samples, with the following classifications: Normal (4602), Mild Wedge (1384), Moderate Wedge (3222), Severe Wedge (1454), Mild Biconcavity (570), Moderate Biconcavity (1172), Moderate Crush (996).

### 3 Methods

After the data pre-processing step, we build a 3D convolutional network based on ResNet-152 which takes 3-channel images as input. A residual learning framework brings ease of training compared to other convolutional networks by adding a  $+x$  to the traditional networks’ activation which is  $y = f(x)$ , allowing the gradient to pass backwards directly. This formulation of  $y = f(x) + x$  can be realized in the shortcut connections of feed forward networks. The difference is that, ResNet’s shortcuts perform identity mapping and adding its output to the outputs of the stacked layers [2]. Figure 2 shows a visualization of the shortcut operation in a feed forward network.



**Fig. 2.** Residual learning: a building block [2]

### 3.1 Layers and Architecture

A generic ResNet is composed of Batch Normalization, Rectified Linear Unit, Max Pooling, Convolutional Layers and Fully Connected. ResNet-152, or Deep Residual Network with 152 Layers is implemented for this research. The ResNet used in this research is largely based on the Residual Network implementation for ImageNet. The network accepts 3-channel images as input and are resized to  $244 \times 244$ , with its shorter side randomly sampled in [256, 480] for scale augmentation. Batch Normalization is done after each convolution and before activation. Training is done from scratch since the data on hand is 3-dimensional and therefore needed to be augmented to fit the requirements of 3-channel Residual Net. The learning rate is 0.0001. Stochastic Gradient Descent is used with mini-batch size of 256. The last layer is a single fully connected layer with an output vector of size 1000.

1. **Convolution.** Convolution operations are applied to input volumes to form a 3-dimensional arrangement of neurons arranged by height, width and depth. Neurons are connected to a region of the input volume called a receptive field of size  $N \times N$ . Each neuron therefore has  $N \times N \times 3$  weighted inputs. A convolution operation takes a kernel on an input image and transforms information encoded in pixels. A specified kernel size is convolved through the entire input image forming “activated regions” [3].
2. **Pooling.** Pooling reduces the amount of parameters and the computations needed to be executed in the network. This is also employed to reduce overfitting [3]. Max Pooling is used after the first batch of convolution operation. A  $3 \times 3$  sliding window is used and for every channel the values will be down-sampled according to the results of the max pooling operation. Average Pooling is used at the end of all the residual blocks with sliding window of size 7.
3. **Batch Normalization.** Batch Normalization is an operation used in deep neural networks that allows the usage of higher learning rates. It is a step between each layer where the output of the previous layer is normalized first before the next layer [4]. Batch Normalization is used every after convolution operation in this research. The learning rate used is 0.0001.
4. **Rectified Linear Unit (ReLU).** ReLU is the most common activation function for the outputs of CNNs used in deep learning. It has an equation  $\max(0, x)$  [3]. All residual blocks in this research uses ReLU.

5. **Fully Connected Layer.** Neurons of the fully connected layer have full connections to all the activations from the previous layer [3]. It is used to construct the desired number of outputs and therefore usually placed on the last layers of the network. This research used only one Fully Connected layer after average pooling operation. It is a single vector with 1000 outputs.

ResNet-152 Architecture

Layer	Output Size	Num. of Outputs	Kernel Size	Stride	Padding	
conv1	112x112	64	7	2	3	
bn_conv1						
scale_conv1						
conv1_relu						
max pooling						
res_2a	56x56	64	3 (sliding win)	2		x 3
bn_2a						
scale_2a						
res_2a_relu						
res_2b						
bn_2b	56x56	64	1	1	1	
scale_2b						
res_2b_relu						
res_2c	56x56	256	1	1	0	
bn_2c						
scale_2c						
res_2c_relu						
res_3a	28x28	128	1	1	1	x 8
bn_3a						
scale_3a						
res_3a_relu						
res_3b	28x28	128	3	1	1	
bn_3b						
scale_3b						
res_3b_relu	28x28	512	1	1	0	
res_3c						
bn_3c						
scale_3c						
res_3c_relu						
res_4a	14x14	256	1	1	0	x 36
bn_4a						
scale_4a						
res_4a_relu						
res_4b	14x14	256	3	1	1	
bn_4b						
scale_4b						
res_4b_relu	14x14	1024	1	1	0	
res_4c						
bn_4c						
scale_4c						
res_4c_relu						
res_5a	7x7	512	1	1	0	x 3
bn_5a						
scale_5a						
res_5a_relu						
res_5b	7x7	512	3	1	1	
bn_5b						
scale_5b						
res_5b_relu	7x7	2048	1	1	0	
res_5c						
bn_5c						
scale_5c						
res_5c_relu						
average pool	1x1		7	1		
fully connected	1x1	1000				
softmax	1x1000					

**Fig. 3.** ResNet with 152 layers is used to create models and classifiers for vertebra fracture classification problem. It is composed of 4 building blocks that are stacked together

6. **Residual Block.** This research uses the ResNet architecture that is composed of 4 building blocks. Each building block is composed of 4 groups of 4-layers of different operations stacked together. In Fig. 3 we refer to these 4 groups as, ResNet Building Block 1, ResNet Building Block 2, ResNet Building Block 3, and ResNet Building Block 4. After the first group, an identity mapping is applied where the output of the previous group is added to the new output. This identity mapping allows the “shortcut” connections. Each of the four ResNet Building Blocks is stacked on top of the same ResNet Building Block design 3, 8, 36, and 3 times, respectively. ResNet Building Block 1 is stacked together 3 times. This will attach to ResNet Building Block 2 which is stacked together 8 times. This is attached to ResNet Building Block 3 which is stacked together 36 times. This is attached to ResNet Building Block 4 which is stacked together 3 times. The number of outputs and the parameters used for each layers stacked and a visual representation of the ResNet design are shown in Fig. 3.

## 4 Results and Discussion

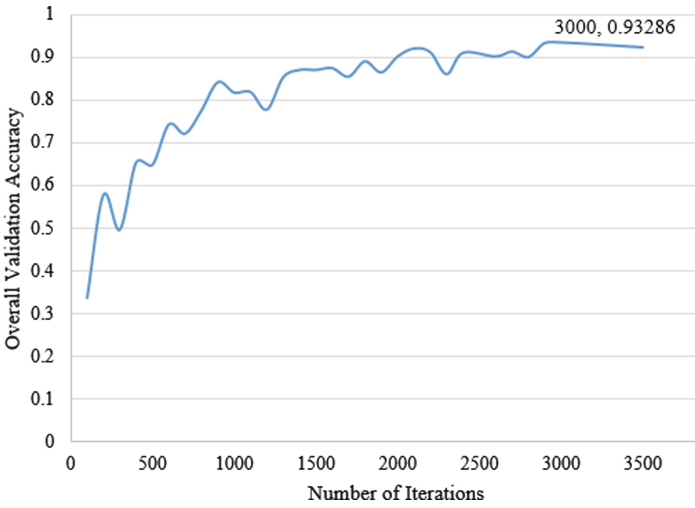
The network was trained up to 3500 iterations. Weights are saved every 100th interval from the 1st iteration to the 3500th iteration. A randomly selected 700 validation data was fed to every model saved. From the 3000th iteration, only the 3500th iteration was saved coming up with 29 weights saved, thus yielding 29 validation results. Each of the 7 classifications had 100 test samples each. The testing was done with batch size 100, therefore arriving with 7 test results.

Figure 4 suggests that ResNet-152 presents 93.29% overall classification accuracy for the validation data using the model trained at 3000 iterations. As the number of iterations increases, validation accuracy also increases.

Training (testing) accuracy is the accuracy when a model is applied on the training (testing) data. Figure 5 shows that training accuracy is generally higher than the testing accuracy with an average of 0.092 difference.

Figure 6 shows the predicted labels of vertebra fracture classification after 700 randomly selected and balanced test data was applied to the classifier. The classifier used was trained on 3000 iterations. The classifier was able to correctly predict 99 true normal 3D vertebra images. Among all the classes, normal class got the highest correct predictions followed by moderate crush and moderate wedge.

Figure 7 shows comparison between the predictions given by ResNet-152 trained on 3000 iterations. Normal class got the most correct predictions and the least number of predicted as false moderate wedge. It can also be inferred that 5 out of 7 classes tend to be falsely classified as normal. This includes mild wedge, moderate wedge, severe wedge, mild biconcavity, and moderate biconcavity. Such trend can be a result of the number of training data set available for normal 3D vertebra images amounting to 4 502, the largest among all the classes. In majority, all 3D vertebra images in the test dataset ended up being predicted correctly.



**Fig. 4.** Trend of the classification accuracy for 700 validation data fed to 29 models trained on different iterations from 100 to 3500

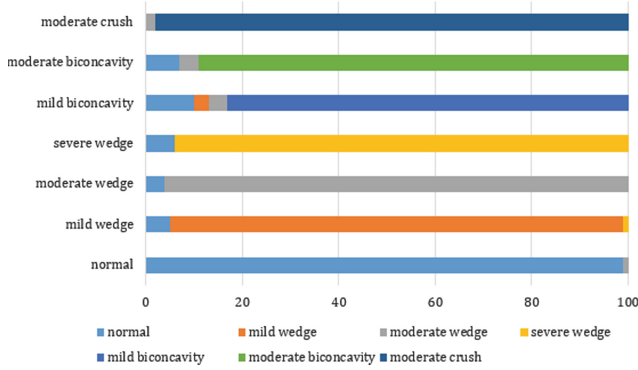


**Fig. 5.** Training accuracies and testing accuracies when 12700 training data and 700 validation data were fed to 29 models trained on different iterations from 100 to 3500

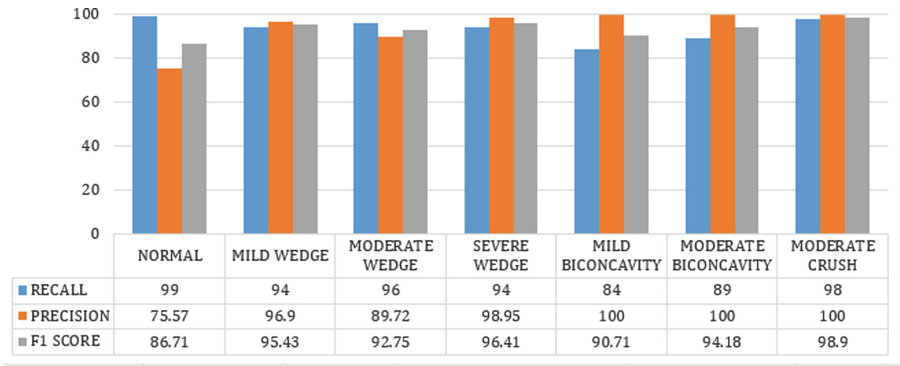
Vertebra Fracture Class	normal	mild wedge	moderate wedge	severe wedge	mild biconcavity	moderate biconcavity	moderate crush
normal	99	0	1	0	0	0	0
mild wedge	5	94	0	1	0	0	0
moderate wedge	4	0	96	0	0	0	0
severe wedge	6	0	0	94	0	0	0
mild biconcavity	10	3	4	0	83	0	0
moderate biconcavity	7	0	4	0	0	89	0
moderate crush	0	0	2	0	0	0	98

**Fig. 6.** 7-way classification confusion matrix of lumbar vertebra fracture classes





**Fig. 7.** Distribution of class predictions given by ResNet-152 trained on 3000 iterations



**Fig. 8.** Evaluation measures using recall, precision and f1-score per lumbar vertebra classification

Figure 8 shows the Recall, Precision and F1 Scores for all seven lumbar vertebra fracture classes. The ResNet-152 trained on 3000 iterations returned 100% predictions of Mild Biconcavity, Moderate Crush and Moderate Biconcavity. This means that out of all the 3D Vertebra Images predicted as moderate biconcavity, mild biconcavity, or moderate crush, 100% of it are truly moderate biconcavity, mild biconcavity, or moderate crush. It can be seen from Fig. 6 that there were a total of 98, 89, and 83 3D vertebra images classified as moderate crush, moderate biconcavity, and mild biconcavity consecutively and 98, 89 and 83 of them are truly moderate crush, moderate biconcavity, and mild biconcavity. Normal and Moderate Wedge scored lowest in precision. There were 131 and 101 3D vertebra images classified as normal and moderate wedge. 32 and 11 of these two were wrongly classified as such. Normal scored highest in recall measure with 99. It can be deduced that out of the all the 100 normal 3D vertebra Images, 99 of them were classified as normal. Mild biconcavity scored lowest in recall with 84.

Among the 700 testing data, the ResNet-152 model trained on 3000 iterations has identified a total number of 653 correct classifications. This shows that the automated model has an overall accuracy of 93.29%.

## 5 Conclusion

The research proposes an automated method to detect and classify vertebra fractures from 3D CT Lumbar Spine Images. This system is limited to 7 classifications only, namely the normal, mild-wedge, moderate-wedge, severe-wedge, mild-biconcavity, moderate-biconcavity, and moderate-crush. The techniques on handling and preparing data for the network is also included in this paper. The architecture followed the ResNet-152, or the Deep Residual Network with 152 layers. ResNet-152 resulted to 93.29% overall testing accuracy for the model trained at 3000 iterations. We have achieved these results by using extracting multiple 3D RGB images from 3D Greyscale CT Images of the lumbar spines.

## References

1. Genant, H.K., van Wu, C.Y., Kuijk, C., Nevitt, M.C.: Vertebral fracture assessment using a semiquantitative technique. *J. Bone Miner. Res.* **8**(9), 1137–1148 (1993)
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. ArXiv e-prints, December 2015
3. Karpaty, A.: Cs231n convolutional neural networks for visual recognition. ArXiv e-prints (2015)
4. Lofe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. ArXiv e-prints 18, 1484–1496 (2015)