The Construction of an Individualized Spinal 3D Model Based on the X-ray Recognition

Irina Grigorieva, Natalya Vyunnik
Kemerovo State University
Kemerovo, Russia
igriva@list.ru

Gleb Kolpinsky
Kemerovo State Medical University
Kemerovo, Russia
glebss@mail.ru

***Abstract*—The actual work is devoted to the construction of an individualized mathematical model of the spine, based on the X-ray images in the frontal and sagittal projections using deep learning convolutional neural networks. The semantic segmentation of X-ray images using the convolutional neural network U-Net is performed, also justifying the choice of network parameters and architecture. All stages of the individualized 3D model building are described, the results of each stage are presented. The model is constructed to subsequently determine the type and the extent of scoliosis with the aim of designing and implementing of the Cheneau corset.**

I. Introduction

The paper presents the results of constructing an individualized spine mathematical model based on the X-ray images in projection using deep learning convolutional neural network. The model is constructed to identify the scoliosis type and extent for the further development and implementing Cheneau corset. The initial data to construct the model are X-ray images in the frontal and sagittal projections.

The spine is the supporting element of the human skeleton, consisting of 32-34 vertebrae divided into five regions. There are cartilaginous interlayers, called intervertebral discs, located between vertebral bodies. The spine can be considered as a kinematic chain, whose links are vertebrae and intervertebral discs. The spine model is constructed based on the descriptions of vertebrae and intervertebral disc. However, the information about intervertebral discs is not available while constructing the model based on X-ray images. Therefore, the model includes only the characteristics of the vertebrae: the measurements, inclination, the displacement of the vertebral bodies relative to each other, the angles between the vertebral bodies [1]. All the parameters are calculated both in the sagittal and frontal planes. In this work, we present a construction of a three-dimensional graphical interpretation of the spine model, shown as a simplified representation of the vertebrae, based on the values of the described parameters. This model will further be used to determine the type and degree of scoliosis, and also to calculate the pressure on the deformed backbone, using mathematical modeling methods, to determine corrective actions.

Further on, the graphic interpretation will be used to determine the type and extent of scoliosis. Currently, the standard method of measuring the angle and degree of curvature in international practice is the Cobb's method. Fergusson's and Lekum's methods are also known. These methods have a number of significant disadvantages. They reflect the curvature only in one dimension and do not consider the rotation. So, they cannot accurately describe the spatial deformation. The accuracy of the measurement angle is often affected by the factors such as the choice of the vertebrae to be measured and the quality of the image. The results of measurements performed on the same image could be affected by a high inter-observer variability. Therefore, it is necessary to use the mathematical methods to perform reliable quantitative scoliosis assessments.

The current research will redound to the benefit of the medical community since it satisfies the growing demand for a system that would provide a 3D model of the spine and test it for scoliosis. The spatial three-dimensional model, supplemented by data on the scoliosis measurement by various methods and by the future results of mathematical modeling, will facilitate the decision-making process and the choice of treatment for scoliosis. On the further stage, we are planning to supplement the model with the description of a body surface, which will permit to develop and implement Cheneau corset. The further implementation of the project will allow to produce Cheneau corsets in Kemerovo Region, which will lead to their cheaper and faster implementation and will ultimately increasing the availability of Cheneau corsets for the scoliosis correction. Within the initial part of the project, the individualized 3D spine model is constructed.

Radiography plays an important role in the visual diagnosis of diseases affecting the human musculoskeletal system. It is also the most accessible way of research and it is often prescribed for the children in the initial diagnosis of scoliosis.

X-ray image is a sufficiently complicated subject of the research, since it displays all the bones covered by the X-ray image area. The images of ribs, hip bones, and arm bones in the lateral projection can overlap the images of the spine, which significantly impedes its recognition. In the image, the spine has the same brightness as the bones shielding it, which causes extra difficulties. In addition, any notes made on the X-ray film by hand, as well as metallic elements of clothing, may be displayed on the images. Therefore, the primary objective of the construction of the spinal model is to locate the spinal column and to separate vertebrae.

II. Related work

Machine learning methods are increasingly being used to solve the problems of detection and recognition of objects on

medical diagnostic images. The work on analyzing images of the medical diagnostics in general and analyzing images of the spine in particular can be roughly divided in this research using deep learning neural networks and all others. Since the advent of neural networks is based on deep learning and libraries that enable the use of such networks, in particular Keras, the multi-layer neural networks have become widely used in the medical diagnostics in a wide variety of contexts. A number of works are devoted to the analysis of the spinal images. Most of them analyze the images of computed tomography (CT) and magnetic resonance imaging (MRI). These images are layered, more detailed and contain substantially less noise. In addition, in the vast majority of such works, the neural network solves the regression problem by determining the coordinates of the desired points. The research [2] is devoted to the use of neural networks to recognize the coordinates of the centers of the vertebrae on the CT images. Authors in [3] describe how to use ConvNet together with the Sobel operators to search the coordinates of the spine damages on the CT images. The work in [4] discusses the use of convolutional neural network on the basis of ConvNet with the additional layers of boosting to analyze the X-ray images and the search for the coordinates of the vertebrae. Some of the works are devoted to the use of other type of network: these networks perform semantic segmentation of images, i.e., they determine the points of the image that belong to the objects of interest. [5] describes the model of a convolutional network U-Net, used for the segmentation of ISBI cells on light microscopy images. The idea of the U-Net model was the continuation of the ConvNet model used to classify images. A distinctive feature of U-Net is that in this network, in a forward motion, the image represented by a three-dimensional feature card is folded in size but increases in depth, after which a convoluted network intended for classification has a fully connected layer that performs this function. In the U-Net, a completely connected layer is missing, instead of it, the reverse deployment of the image is performed on the reverse of the network. A similar approach is used in [6] to determine the contours of the heart, lungs and clavicles on the X-ray; this work describes the application of three different convolutional networks, other than U-Net, to solve this problem, and gives the recommendations on the choice of the parameters for convolutional networks.

The novelty of this work is that it uses the semantic segmentation approach to determine the spine, in contrast to most works solving the problem of regression. The result of the neural network activity is the image of the vertebrae in white on a black background. In this case, the classification is performed for each pixel in the image. This approach preserves more information, which can be used in further analysis and in particular to determine the contour.

There are few papers devoted to the study of X-ray images without the use of neural networks. The paper [7] discusses automatic recognition of vertebrae represented in the projections by polygons. The method proposed in the paper automatically detects the convex angles of these polygons. The key points that can be vertex angles are detected by building the SIFT descriptors, after which the model is learning, being based on the support vector method. In the research [8] the Cobb's angle is calculated from the x-ray images using a structured vector regression. The article [9] describes a combined filter to identify the features of the vertebrae. The combined filter in this article is a combination of anisotropic, sigma, and differential filters. After applying the method of differential boundary allocation to find the contours of the vertebrae on the filtered image, the Cobb's method is used in the research to determine the degree of scoliosis. Nevertheless, over the past few years, neural networks have made tremendous progress in the image recognition, the convolutional neural network approach is currently dominant, including the recognition of medical imaging.

III. Foremost stages of model construction

The construction of the spine spatial model is a complex multi-stage process; its sequence diagram is shown in Fig. 1. The original image enters the preprocessing module, which scales the image, converts the color of image to grayscale, and increases its contrast. The result is saved in a file that serves as input data for the recognition module. The recognition module uses the trained U-Net network to determine the position and the shape of the vertebrae in the projections. The result of this module's work is also a file with the recognized images of the spine, which, in its part, enters the markup module. This module marks up the contours of the vertebrae and computes the rectangles describing them in the frontal and sagittal projections. The result of the module's work is a file containing a description of the geometry of the vertebrae in the projections, which, if necessary, can be additionally marked up in the preprocessor interface of the program to work with 3D model and resaved.

Within the framework of this project, a program has been developed to work with the 3D model. This program is a preprocessor and postprocessor to build a 3D model of the spine. The preprocessor interface displays automatically created markup, which the user can change if necessary. After that, the module to construct the 3D model builds the spinal model from the generated markup and displays it in the postprocessor part of the program. The application was implemented in C#. Image preparation, recognition and markup modules were implemented in Python using Pandas, Keras, Cv2 libraries.

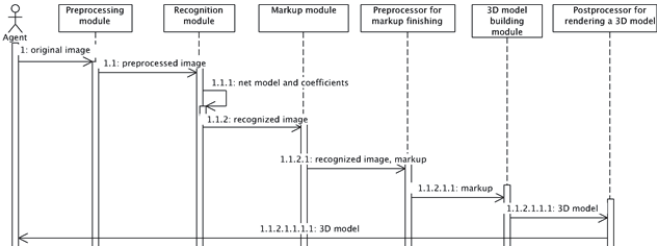


Fig. 1. Sequence diagram of the spine 3D model construction

IV. Recognition of the spine

U-Net networks have spread in recent years and are often used to analyze medical images [5]. The network architecture used in this work is shown in Fig. 2. In this network, in a forward motion, the image is minimized in size but increases in depth. The image in this case is a three-dimensional array $n \times m \times k$, where n and m are the original image sizes $n=m=512$, k is

the depth and for the original image is 1. The depth of the image is 1, since the original image is converted to the shades of gray in the range [0,1] without loss of quality. The architecture of the convolutional neural networks takes into account the fact that the input data are images; therefore, the convolution layers have neurons located in three dimensions: width, height, and depth. In the process of convolution, the image passes through a set of blocks consisting of two consecutive convolution layers and the subsequent pulling layer. In the Fig. 2 the layers are represented by rectangles indicating the type of layer and the size of the filter that it uses. The dimensions of the incoming images are indicated near the arrows showing the direction of image transmission. The convolution layers are marked as *conv*. The parameters of the convolution layer are chosen in such a way that the size of the image $n \times m$ does not change when passing through the convolution layer. The image changes only in depth, if this is the first of two successive layers, and does not change in size if it is the second one. In different versions of the network at the initial stage, when passing through the first block, 16 to 64 of such layers are formed. When passing the next layers of convolution, the number of layers doubles in depth each time until it reaches 512 or 1024 in different versions of the network. After two layers of convolution, the pulling layer is located, and its function is to reduce gradually the spatial size of the image. The pulling layer works independently on each fragment of the input depth and changes it spatially, using the operation of identifying the maximum, in the Fig. 2 such layers are labeled as *max pool*.

As a result of passing through the layers of convolution and pulling, the image essentially decreases in relation to the initial size, but increases in depth. After convolving the image in the forward direction, the image is re-deployed using both the two convolution layers blocks both the transpose convolution following them. The transpose convolution layers perform the inverse of the direct convolution, in the Fig. 2 they are labelled as *transpose*. The relationships between the size of the filters, the zero padding and the filter step for the convolution and transpose convolution layers are described in [10]. The network architecture is nonlinear, since some layers use not only the result of the previous layer, but also the computational result of one of the earlier layers. These two results are glued together in the third dimension, and thus additional layers are formed in depth. In the Fig. 2, this action is labelled as *concat*. The last convolutional layer uses sigmoid activation function and performs the convolution of the image to its original depth dimension. The result of the network is the image containing the objects you are searching for.

While conducting the research, we have tested various versions of the network architecture, as well as various values of the U-Net parameters. In this paper, when describing the network architecture, the number of layers on the first and maximal depth convolutional layer is indicated, for example, 32-512. Since the training of one version of the network requires significant computational costs, at the first stage, we selected the optimal parameters on the 32-512 network. After that the selected parameters were used to choose the optimal network architecture. The values of selected testing parameters were based on the analysis of works devoted to the image recognition using the neural networks ConvNet and U-Net for the image recognition and classification.

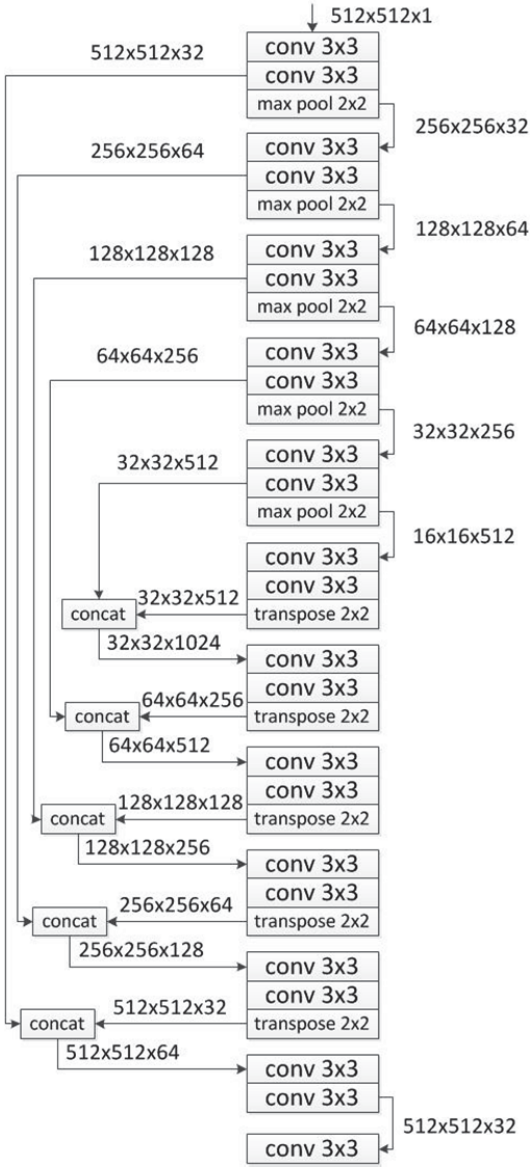


Fig. 2. U-Net network architecture

We considered the loss function and the activation function of convolutional layers, with the exception of the latter, as variable parameters. Three different activation functions were considered: rectified linear unit ReLU, scaled exponential linear unit SELU and exponential linear unit ELU. [5] describes a network using the ReLU function, but in [6] it is recommended to replace it with ELU. Testing was carried out on a test set every 10 epochs. The results were compared by means of the $F1$ score, which is the harmonic average of precision P and recall R:

$$P = \frac{TP}{TP + FP},$$

$$R = \frac{TP}{TP + FN},$$

$$F_1 = \frac{2 \cdot P \cdot R}{P + R},$$

where *TP* stands for 'true positive', i.e. the number of correctly recognized white pixels of the image; *TN* - 'true negative', i.e. the number of correctly recognized black pixels of the image; *FN* - 'false negative', i.e. the number of white pixels of the test image that were mistakenly recognized as black; *FP* – 'false positive', i.e. the number of black pixels of the test image that were mistakenly recognized as white.

The Table 1 presents average values of the *F*1-score for the test set on the best epoch. Fig. 3 shows the graphs of the average *F*1-score values in relation with the epoch number and obtained with different activation functions. The quality of recognition at the initial stage rapidly increases, then the functions noticeably oscillate. Moreover, we observe a noticeable effect of retraining in the SELU and ELU functions, while for the ReLU function this effect exists, but less extended. The function ReLU shows a significantly better recognition quality.

We considered the Dice *D* and Jaccard *J* score as a loss function:

$$D = \frac{2TP}{2TP + FP + FN},$$

$$J = \frac{TP}{TP + FP + FN}.$$

Although the use of the Jaccard score is recommended in [6], testing conducted within this research has shown that the choice of the metric has little effect on the learning of the network and its generalizing ability. It should be noted that the Dice and Jaccard metrics are similar, but the Jaccard metric penalizes the network with more false positive and false negative pixels. Nevertheless, the use of the Dice score gives a slight advantage.

In all the considered variants of the network, the effect of retraining is observed with a greater or lesser extent. The maximum average value of *F*1 score is achieved for the activation functions ReLU, ELU and SELU at the epochs 410, 100 and 50, respectively, when using the loss function Dice score. In [11], a method is described for combating retraining by adding dropout layers to the network. The dropout layers randomly exclude from the training the network nodes with their connections, thus weakening the joint adaptation of nodes to learning. As shown in [11], this approach significantly reduces retraining. To reduce this effect in this work, 2 dropout layers were added to the network shown in Fig. 2 after the first convolutional layers in blocks 1 and 4. The resulting averaged values of *F*1 score are reported in Fig. 3 and labelled as 3. It can be seen from the figure that adding dropout layers really reduces the effect of retraining. In this case, the maximum average value of *F*1 score is reached on the 910-epoch. However, the network obtained using the activation function ReLU without the use of dropout layers demonstrates the best average values of the *F*1 score, which cannot be reached by the networks with dropout layers even during the following 1000 epochs. In this paper, it was not possible to find the number and position of the dropout layers able to lead to better.

For the selected set of parameters, we have compared the architecture of U-Net 16-512, 32-512, 32-1024 and 64-1024. All these networks use the blocks of two successive convolutional layers and the next layer of pulling in a forward motion. In the reverse motion the pulling layer is replaced by a layer of transpose convolution. Thus, in the networks 32-512 and 64-1024 on the forward and reverse motions, there are 4 blocks and one additional last block ending in the last convolution layer, where 4 concatenations are performed, in addition to other 5 in the networks 16-512 and 32-1024. Therefore, their use is associated with greater computing costs. The maximum average values of *F*1 score for approved architecture variants are presented in Table 2. The resulting *F*1 score for the different architecture variants in relation with the epoch number is shown in Fig. 4. The architectures 16-512 and 32-1024 demonstrate a much better overall ability by using more convolutions.

The calculations were carried out using a parallel version of the TensorFlow library on the GPU server with Nvidia Tesla P100 video card of the Center for Collective Use of High Performance Computing Resources of Kemerovo State University.

TABLE I. AVERAGE *F*1 SCORE VALUES FOR VARIOUS COMBINATIONS OF NEURAL NETWORK PARAMETERS

	Activation function		
Loss function	ELU	SELU	ReLU
Dice score	0.6490	0.6354	0.7106
Jaccard score	0.6288	0.6184	0.7103

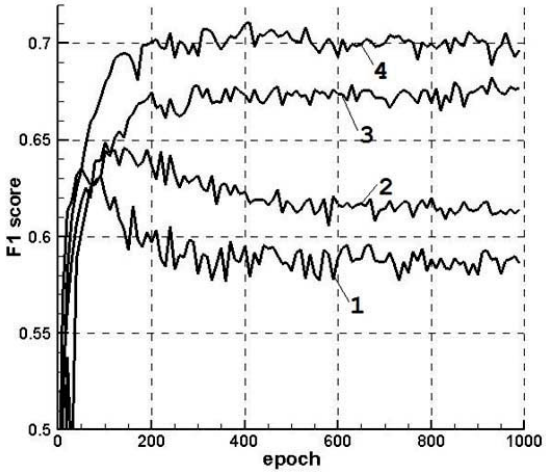


Fig. 3. Recognition accuracy for Dice score loss function and different activation functions: 1 – SELU, 2 – ELU, 3 – ReLU with dropout layers, 4 - ReLU

TABLE II. AVERAGE *F*1 SCORE VALUES FOR DIFFERENT ARCHITECTURE VARIANTS

Architecture	32-512	16-512	64-1024	32-1024
***F*1 score**	0.7106	0.8694	0.7198	0.8778

As a result, we recognized 32-1024 with the activation function ReLU and the loss function Dice score as the optimal version of the network architecture. In all cases, the networks

were trained using the Adams algorithm with an initial step of 10^{-5}. When using this network, the highest value of $F1$-score is achieved at 200-epoch. An example of the recognizing result of a test image in frontal projection, shown in Fig. 5a, is reported in Fig. 5b. Fig. 6 shows the relation of *FN* and *FP* pixels with the number of epochs referred to the total number of the white pixels. It can be seen from the figure that in the learning process the number of *FP* points is reduced, but the number of *FN* points increases. This means that in the late stages of learning the vast majority of erroneous points are the *FN* points. It is worth noting that all the considered loss functions equally performed in the same way in relation with the *FN* and *FP* points. The penalty for *FN* points needs to be increased.

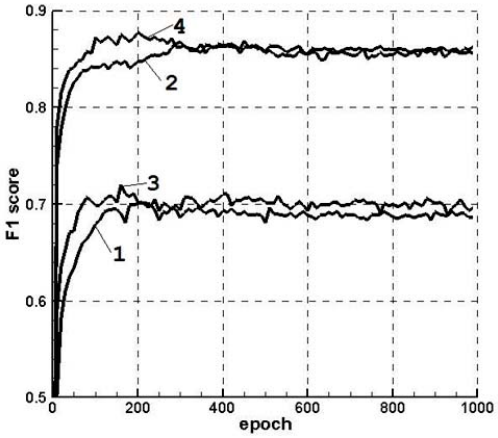


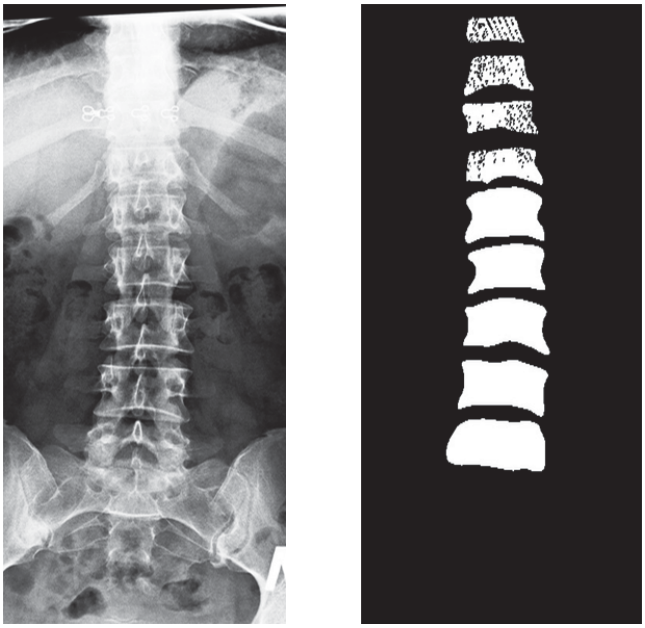
Fig. 4. Recognition accuracy for different architecture variants: 1 – 32-512, 2 – 16-512, 3 – 64-1024, 4 – 32-1024

We have used a set of 118 images in the frontal projection for training and testing, 70 of which have been used to construct the training set, and the remaining 48 as a test set. Since seventy images are not enough for the network to learn, these 70 images were the subject to augmentation, resulting in 1050 images being generated for the network training. The use of augmentation to obtain an acceptable number of images for training is a common practice and is described, for example, in [5].

Since the images with the full image of the spine are not sufficient, then for the training we used images on which fragments of the spine consisting of several vertebrae are depicted. Many of these pictures were used to recognize the vertebrae in the sagittal projection. For the training we used 80 images with fragment and only 26 pictures with a full image of the spine, as well as 18 complete images for testing. The recognition of images in the sagittal projection produced worse results than in the frontal surface. For the best version of the network, the highest average $F1$ score for the frontal surface was 0.8778, and for the sagittal surface was only 0.8085. Special difficulties arose when the images of the vertebrae appear partly overlapped with ribs ones.

V. Contours computing and a 3D model construction

The contour calculation module defines the contours of the vertebrae using the functions of the Cv2 library, along with the rectangles that describe constructed vertebrae. Fig. 7a shows the original image of the spine in the frontal projection along with the contours manually (in black) and automatically (in grey) generated. Although the contours do not completely coincide, there is a good correspondence both along the contours of the vertebrae. The result of the markup based on the recognition results is shown in Fig. 7b. Fig. 8 shows the results for the sagittal projection. As one can be se, the contour obtained with the recognition matches in a worse way with the manual contour, especially for the upper vertebrae.



(a) (b)

Fig. 5. Result of image recognition: (a) - original image, (b) - recognition result

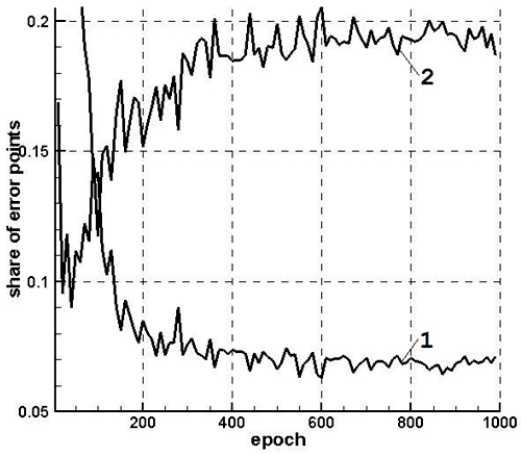


Fig. 6. Dependence of the erroneous points share on the number of the epoch: 1 – FP points, 2 – FN points.

The construction of a 3D model is based on the information about rectangles that represent vertebrae in projections. The search for a descriptive rectangle for each vertebra is a two-step procedure. The first stage involves a search for the minimal convex hull according to the Jarvis algorithm. It is then followed by a search for the minimal descriptive rectangle according to the algorithm described in [12]. After the

minimum rectangles have been detected for all the vertebrae, the adjacent rectangles are checked for an overlap. Overlapping rectangles are reduced so as to eliminate the overlap. Besides, at the preparatory stage, the images are scaled to coincide vertically with the examined part of the spinal column: then the differences in the data obtained from the sagittal and frontal projections will be minimal.

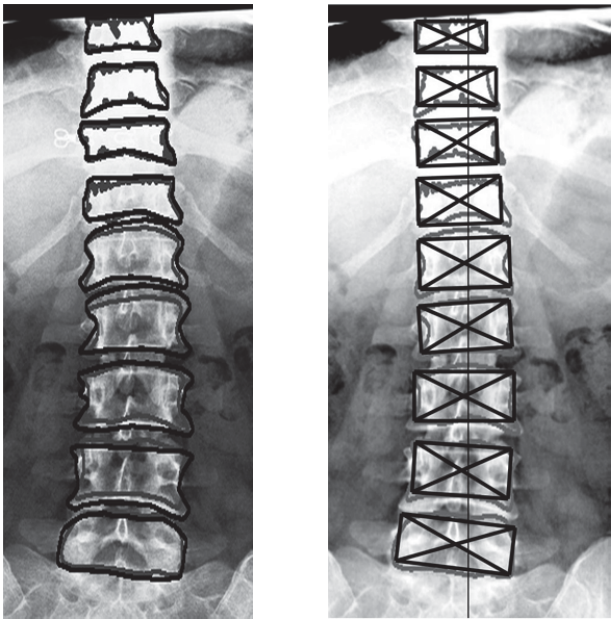
The images themselves and the description of the vertebrae in the form of rectangles are the input data for the program for constructing a 3D model. The preprocessing part of the program visualizes the images in projections with superimposed rectangles that can be corrected manually. For each vertebra, the program determines dimensions, center position, axial tilt relative to the axes of coordinates, and the rotation of the vertebra relative to its own axis. Since the vertebrae are tilted relative to the vertical axis, their height is displayed with distortion on the frontal projection. Therefore, the height of the vertebrae and the coordinates of their central points along two axes are determined from the sagittal projection. To determine the rotation of the vertebrae according to the Raimondi technique, it is necessary to determine the attachment points of the transverse processes to the vertebral body in the frontal projection, which has not yet been realized in this research. Therefore, at present all the vertebrae are considered to be unturned. It should be noted that all listed parameters are calculated precisely for vertebral bodies.

The present research employed a standard model of the spine to calculate the individualized model or its fragment. The standard model takes into account the structures of the vertebrae in various regions of the spine. The program stores previously described 3D models of all types of vertebrae that make up the spine. When constructing an individualized model, individual parameters of vertebral bodies that have been computed as a result of recognition and marking of images are transferred to this standard model. However, the shape and position of all the processes are taken from the standard model since this information is not extracted from the X-ray images.

Part of the pictures used for testing in this work contained images of not all vertebrae. Most of the images did not include the cervical section, in addition, often the upper part of the image turned out to be lighted and it was not possible to recognize the upper vertebrae. When building the model in the application, you must manually specify the number of the highest marked vertebra to form the correct vertebra configuration, taking into account the spine sections. Thanks to the use of this approach, it is possible to construct a realistic representation of the final spinal model. The Fig. 9 shows the visualization of the spinal model for a vertebral column of 9 vertebrae, presented in the figures earlier.

The generated 3D model is visualized in a separate application window, while viewing the model, standard tools are available to zoom in and rotate around the central axis. The resulting model is saved in the universal *obj* format. The model can be opened and edited in one of the popular 3D modeling programs or printed on a 3D printer.

In the future, it is planned to calculate and take into account the rotation of the vertebrae in the model, as well as the addition of a module for analyzing the type and degree of scoliosis using various methods for their determination.



(a) (b)

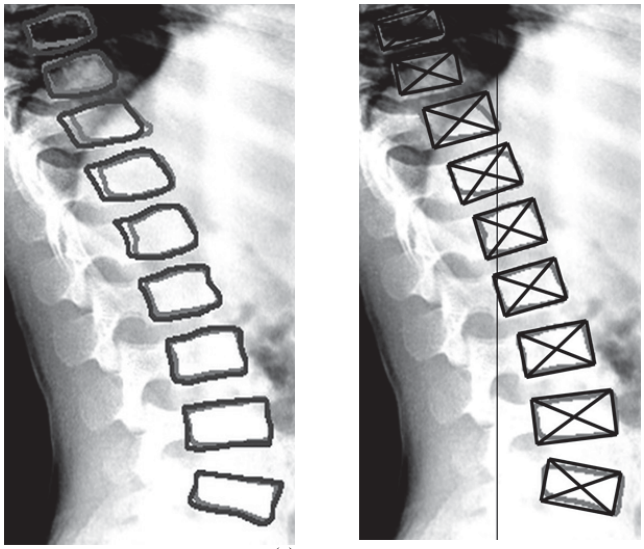
Fig. 7. Result of frontal projection markup: (a) - overlapping of manual marking contours (black color) and contours obtained during recognition (gray color) to the original image, (d) - markup based on recognition results.

VI. Conclusion

In this paper, we presented an approach to construct an individualized spinal model based on X-ray images in projections. The model takes into account the measurements, inclination, curves and position of the vertebrae relative to one another.

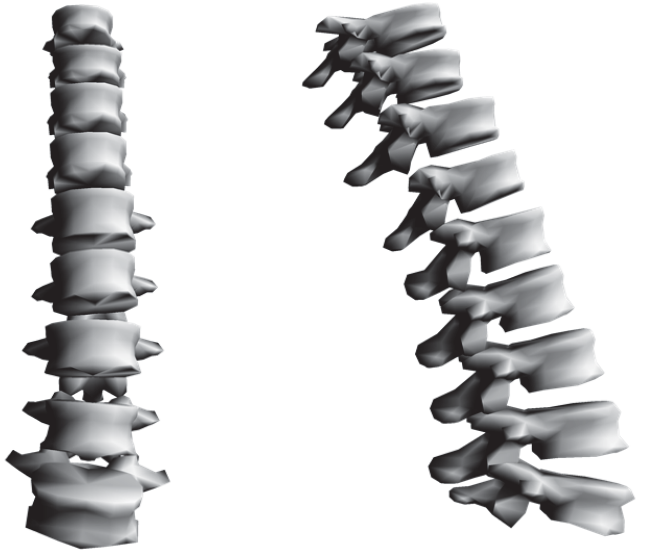
The U-Net neural network is used for the X-ray images recognition. In the paper, we described the parameters and the network architecture selection, as well as the preparation of images for recognition. The network was successfully trained, it demonstrated an acceptable generalizing ability in frontal projection images and markedly worse for sagittal projection. Currently, work on the way to improve the quality of image recognition, including the selection of methods for pre-processing images, as well as the selection of a loss function that takes into account the proportion of erroneous *FP* and *FN* points. In addition, the markup module of the images obtained as a result of recognition is being further developed, as well as the 3D model building module.

The overall program obtained as a result of this work will be used in Kemerovo medical institutions for analyzing X-ray images, determining the type and degree of children's scoliosis. On the further stage, we are planning to supplement the model with the description of a body surface, which will allow to develop and implement Cheneau corset.



(a) (b)

Fig. 8. Result of sagittal projection markup: (a) - overlapping of manual marking contours (dark gray color) and contours obtained during recognition (light gray color) to the original image, (b) - markup based on recognition results.



(a) (b)

Fig. 9. 3D model based on recognition results: a - front view, b - side view

VII. Acknowledgements

The work is supported by the “Mathematical modeling of spinal scoliosis treatment individualization” university grant.

References

[1] A.V. Gladkov, Yu.V. Sivets, K.Yu. Avdeeva, “New approaches in using the mathematical apparatus in the construction of three-dimensional models of the spine”, Surgery of the spine, 2005, (1), pp. 100-104 (in Russian).

[2] A. Suzani, “Automatic vertebrae localization, identification, and segmentation using deep learning and statistical models”, 2014, Web: https://open.library.ubc.ca/cIRcle/collections/24/items/1.0166073.

[3] H.R. Roth, Y. Wang, J. Yao, L. Lu, J.E. Burns, R.M. Summers, “Deep convolutional networks for automated detection of posterior-element fractures on spine CT”, in Proc. SPIE 9785, *Medical Imaging 2016: Computer-Aided Diagnosis*, 97850P, 24 March 2016, Web: doi: 10.1117/12.2217146.

[4] H. Wu, C. Bailey, P. Rasoulinejad, S. Li, «Automatic Landmark Estimation for Adolescent Idiopathic Scoliosis Assessment Using BoostNet», *in Medical Image Computing and Computer Assisted Intervention − MICCAI 2017: 20th International Conference*, 2017, Proceedings, Part I, Web: doi: 10.1007/978-3-319-66182-7_15.

[5] O. Ronneberger, P. Fischer, T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation”, *in Navab N., Hornegger J., Wells W., Frangi A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Lecture Notes in Computer Science*, vol 9351, 2015, Web: doi: 10.1007/978-3-319-24574-4_28.

[6] A.A. Novikov, D. Lenis, D. Major, J. Hladůvka, M. Wimmer, K. Bühler, “Fully convolutional architectures for multi-class segmentation in chest radiographs”, IEEE Transactions on Medical Imaging, Web: https://arxiv.org/pdf/1701.08816.pdf.

[7] F. Lecron M. Benjelloun, S. Mahmoudi, “Fully automatic vertebra detection in X-Ray images based on multi-class SVM”, *in SPIE Medical Imaging: Image Processing*, 2012, Web: doi: 10.1117/12.928585.

[8] H. Sun, X. Zhen, C. Bailey, P. Rasoulinejad, Y. Yand, S. Li, “Direct Estimation of Spinal Cobb Angles by Structured Multi-output Regression Information Processing in Medical Imaging”, in IPMI 2017. *Lecture Notes in Computer Science*, vol 10265, 2017.

[9] H. Anitha, G. K. Prabhu, “Automatic Quantification of Spinal Curvature in Scoliotic Radiograph using Image Processing”, *J. Med. Syst*., 36, 3, June 2012, pp. 1943-1951, Web: doi: 10.1007/s10916-011-9654-9.

[10] V. Dumoulin, F. Visin, A guide to convolution arithmetic for deep learning, 2016, Web: https://arxiv.org/pdf/1603.07285.pdf

[11] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”, *Journal of Machine Learning Research*, 15, 2014, pp. 1929-1958.

[12] D. Eberly, Minimum-Area Rectangle Containing a Set of Points, 2015, Web: https://www.geometrictools.com/Documentation/MinimumAreaRectangle.pdf.