Segmentation of bone structure in X-ray images using convolutional neural network

Cosmin Cernazanu-Glavan[1], Stefan Holban[1]
[1]*Politehnica University of Timisoara, 300006, Timisoara, Romania*
cosmin.cernazanu@cs.upt.ro

***Abstract*—The segmentation process represents a first step necessary for any automatic method of extracting information from an image. In the case of X-ray images, through segmentation we can differentiate the bone tissue from the rest of the image. There are nowadays several segmentation techniques, but in general, they all require the human intervention in the segmentation process. Consequently, this article proposes a new segmentation method for the X-ray images using a Convolutional Neural Network (CNN). In present, the convolutional networks are the best techniques for image segmentation. This fact is demonstrated by their wide usage in all the fields, including the medical one. As the X-ray images have large dimensions, for reducing the training time, the method proposed by the present article selects only certain areas (maximum interest areas) from the entire image. The neural network is used as pixel classifier thus causing the label of each pixel (*bone or none-bone*) from a raw pixel values in a square area. We will also present the method through which the network final configuration was chosen and we will make a comparative analysis with other 3 CNN configurations. The network chosen by us obtained the best results for all the evaluation metrics used, i.e. warping error, rand error and pixel error.**

***Index Terms*—image segmentation, neural network, convolution, biomedical image processing**

I. Introduction

It is relatively easy for a human to recognize some objects out of an image. Thus, starting from small age, we manage to accurately identify certain common objects from a picture. The identification of specific objects from the field we are working in can be performed only after a process of acquiring good knowledge and experience over time. In the same way, the present article aims at identifying (segmentation) some bone tissues from an X-ray.

If the entire process is desired to be automatically performed by a computer, then we need a model trained to make the difference between a bone tissue and the rest of the image. Deep neural networks are among the best models that are recommended for this type of tasks due to their method of managing to imitate the human recognition process in the visual cortex.

Their mode of operation consists in extracting some features characteristic for each image and creating some feature maps at the level of each physical layer of the network. The method of creating each feature map consists in convolving the input image with a linear filter, adding a bias term and applying a non-linear function. Thus, we create a sequence of features that converge to a final layer (the output layer). On the final layer a 2-type classification process is performed: bone tissue and others. As the neural network must segment an X-ray image, the 2 classes are sufficient because we are interested only in the one identifying the bone tissue.

This architecture was presented for the first time by Fukushima[1] (1980) by using a network architecture (Neocognitron) that further led to the development of several similar architectures. This network architecture was later developed by LeCun[2] and Simard[3] under the name of Convolutional Neural Networks (CNN).

As we mentioned above, the mode of operation of these networks consists in realizing some feature maps that are extracted from an image with the help of specialized filters. This operating method is used by other similar techniques from computer vision field: local receptive fields for predictor units based on semi-linear predictability minimization [4] or Gabor filters [5]. The main problem of these methods is that the parameters of the filters are calculated in the initial phase and remain constant until the end of the analysis process.

Contrary to this method, the use of CNN makes possible the random initialization of filters whose final values are determined only after the neural network training process. This procedure presents a double advantage. First, the filters are calibrated on the dataset that must be recognized and will be used for training the network. Second, the minimum intervention of the user (the network can calculate itself the necessary parameters).

A major improvement for the CNN architectures consisted in introducing max pooling [6][7] layers. The main role of this layer is to extract only the best features existing at a certain time period and to eliminate the others. This method is highly connected to the capacity of generalization/specialization of the network. By selecting a feature and eliminating the others we increase the capacity of generalization; and by selecting the best feature from a great number of features, we manage to keep the level of specialization constant.

The use of this type of network for image recognition is very common and so successful that in the field of handwriting recognition, the human recognition capacity was surpassed by the neural network. [8] Moreover, this type of network was used for segmenting the neural membranes and the results obtained were promising. [9] During time, this technique was used for face recognition [10], field in which it recorded the best results.

A step forward in using this type of neural networks is

This work was partially supported by the strategic grant POSDRU/89/1.5/S/57649, Project ID 57649 (PERFORM-ERA), co-financed by the European Social Fund – Investing in People, within the Sectoral Operational Programme Human Resources Development 2007-2013.

building some Convolutional Neural Network Committees. This technique consists in training several neural networks with different configurations on the same dataset. There are several techniques for combining their results, but Meier et al. [11] showed that the arithmetic mean for determining a single set of outputs remains the best technique. We must add that this technique was not approached in this paper because of the longer period of training time (several days) required for a single network.

In present, there are several techniques for X-ray image segmentation, but in general they all require the human intervention during one or several stages of the procedure. In what follows, we will present some of the most important techniques used, their number being much higher. We must add that we have chosen the techniques with the most promising results and with a high degree of automation.

Consequently, we can point out the segmentation technique proposed by Hu et all[12] consisting in two key procedures: a fitting weight matrix built to reduce the effect of subjective factors entered by the user in order to obtain the initial rough contour of cervical vertebra and a curve fitting method based on rotating and overlapping parabolic curves to derive the final segments of cervical vertebra.

An interesting approach is presented in [13]. It aims at determining some salient regions. Thus, the proposed algorithm uses a hexagonal representation of the image pixels and a hyper graph structure to process this hierarchical structure.

Another method developed by Qu et all [14] uses the PCA based particle filtering for defining a segmentation algorithm used in noisy medical images such as X-ray angiographic and ultrasound images where the image quality is very poor. It exploits the prior clinical knowledge of object's shape information through a PCA model.

For automated medical image segmentation, a new fuzzy level set algorithm [15] uses fuzzy clustering as the initial level set function. The fuzzy level set method is derived from the classical Hamilton-Jacobi functional and it is enhanced with locally regularized evolution. Even if the author mentions it is an automated method, the human intervention is necessary in the initial stage of the algorithm.

In [16] too, it is required a priori global knowledge of the geometric structure of each hip. Then, a search is performed using a shape context model. The shape model consists of the integration of two statistical models: the first contains knowledge about the edges extracted from radiograph and the second describes the local image structure around each model point. The method proposed in [17] creates a new ontology for knowledge-based approaches of automated systems used in segmentation of radiographic images. The author aims at creating an automated system which could help the doctors in making some decisions.

Using two specialized artificial neural networks trained with statistical information for the segmentation task, in [18] is proposed a new method of X-ray segmentation. Thus, the first network sections the image in two regions (white pixels and non-white pixels), and the second network sections the interest areas only from the white-pixels region. Although it is an automated method, its utility is restrained as it can be used only for simple tasks.

In contrast to the methods presented above, our approach aims at eliminating the human intervention from the segmentation process. It is desirable to obtain a complete automation of the segmentation process irrespective of the type of information contained in an X-ray image. For a successful solution, we considered as a best option the use of a convolutional neural network that could perform an automatic segmentation.

Our goal consists in identifying the bone tissue and delimitating it from the rest of the image. Thus, input data for other automatic methods of diagnose and analysis are supplied.

We performed a pixel segmentation method that determined if the pixel is/is not part of the bone tissue. This can be achieved by cutting up the area surrounding the pixel (this is situated in the centre of the area) and applying a convolution operator for that area.

The network presented in this article uses an area of 128x128 pixels for classifying a single pixel. The classification of all the pixels in an image is done by scanning the entire image with the determined area (128 x 128 pixels). We start from the top left corner and we move the area to the right. When we reach to the right edge, we pass to another row and the area is moved to the left, and the process is being repeated.

The process ends when the area reaches the right bottom corner of the image, thus being covered all the pixels from the image. Each pixel from the image was classified as belonging/not belonging to the bone tissue.

A. Data collecting

The X-rays for which we performed the segmentation process make part from a database of images of DICOM type. They have a resolution of 2492 x 1984 pixels. An example of X-ray can be observed in Fig. 1.

Figure 1. Chest X-ray (DICOM image).

For training the network, we need an initial preprocessing of the image which consists in a manual selection of the areas of bone tissue. Then, it is formed the training set made up of pairs of the type: selected area (128 x 128 pixels) and the type of pixel situated in the centre of the area (bone tissue/others). For achieving a training set, even for a single image, we must determine a number of 2492*1984 pairs defined above.

The number is very high, and if we want to use several images, the number would be much higher. For this reason, interest areas have been selected from these images. For each of these areas, we determine, through the method explained above, pairs of data for each pixel. In Fig. 2 we can notice that the number of pairs resulted is much lower than the original (approx. 10.000 pairs). The new set is more consistent than the older one because there have been selected only the areas of maximum interest, being eliminated the areas of low importance.

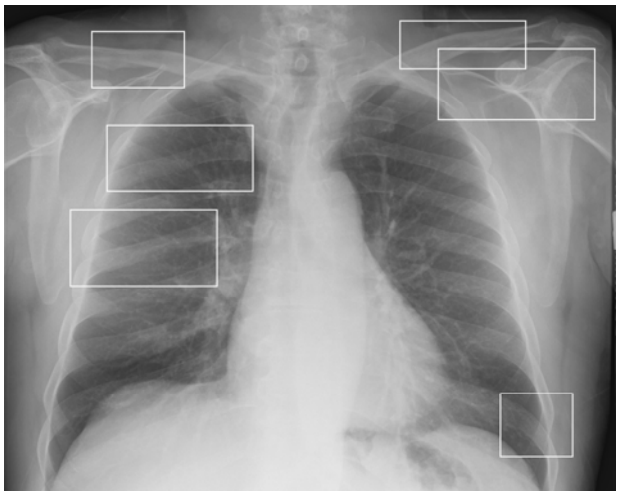

Figure 2. Selecting only a small number of areas of interest.

For each area, we select the bone tissue, the final result being shown in Fig. 3. Now, we will have a correspondence in the field *{bone tissue, others}* for all the pixels in the interest area. This type of representation allows the passage to a new stage: training a convolutional neural network for an automatic classification of all the pixels in the image.

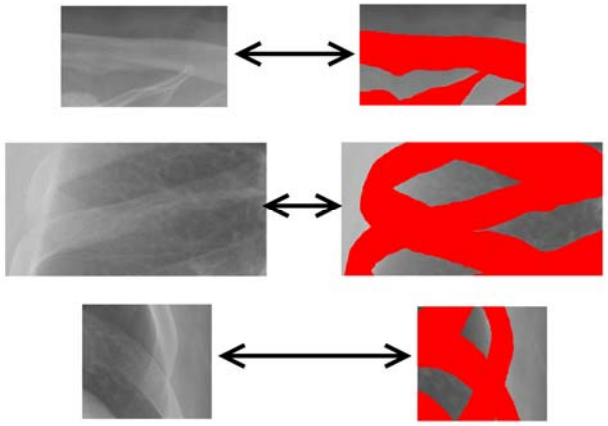

Figure 3. Pre-processing step. Selecting the *bone* area.

II. The Architecture of the Network

A convolutional neural network is composed of several convolutional layers and max-pooling layers that alternate. In our case, the final layers are fully connected layers and two neurons are present at the output (one neuron for each classification class). The model of this network is based on the work on the visual cortex performed by two biologists. [19] [20]. They showed that eyesight has as basis two types of cells. The first type is a simple one that superposes several layers on the local eye receptive cells, while the second type (a complex one) performs a filtering operation of all the outputs supplied by the first type.

The final architecture of the convolutional network used for bone tissue recognition can be seen in Fig. 4. The network is composed of 9 layers: 3 of convolution, 3 max-pooling, one input and two fully connected layers. The training of the network was done using a simple backpropagation algorithm. [24]

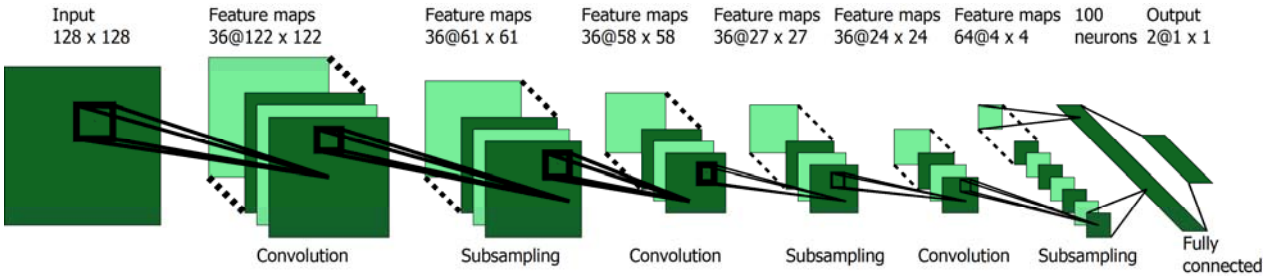

Fig. 4. Convolutional Neural Network architecture

Each of the layers of the network (except for the final layers) is made up of one or several feature maps.

TABLE I. LAYER ARCHITECTURE FOR OUR NETWORK

Layer	Type	Maps @ neurons	Filter size
0	Input	1map@128x128	
1	1 (convolutional)	36 maps@122x122	7 x 7
2	2 (max pooling)	36 maps@ 61x61	2 x 2
3	3 (convolutional)	36 maps@58x58	4 x 4
4	4 (max pooling)	36 maps@27x27	2 x 2
5	5 (convolutional)	36 maps@24x24	4 x 4
6	6 (max pooling)	64 maps@4x4	6 x 6
7	7 (fully connected)	100 neurons	1 x 1
8	8 (fully connected)	2 neurons	1 x 1

The number of these feature maps is determined by the type of the problem and the degree of complexity of the data that must be processed. The configuration of the network is presented in Table I.

The input data, the output data and the functioning method for each type of layer will be further detailed.

A. Convolutional layer

Each convolutional layer has several feature maps associated. For each pair of (*feature map in the previous layer, feature map in the current layer*) a filter is defined. The filter can have the shape of a rectangle, square or circle. For simplifying the operations, the filter of this network will be square.

We will have the same dimension of the filter for all the

filters in a layer. In Fig. 4 can be seen that for layer 1 the dimension of the filter is 7 x 7 pixels, and for layer 3, the dimension is 4 x 4. The role of the filters consists in extracting some features from the input images and grouping them in feature maps.

The filters are characterized by the weights associated to each input and function according to the formula (1)

$$y_{out}(n) = f\left(\sum_{k=1}^{M_x}\sum_{j=1}^{M_y} w_{kj} * y_{in}(k, j) + b_M\right) \quad (1)$$

where, y_{out} represents the output of the M filter (dimension of the filter is M_x x M_y), w_{ij} is the associate weight for a specific input, y_{in} is the input value, b_M is the biases for M filter and f is the corresponded activation function (in our case it is the *tanh* - hyperbolic tangent). Schematically, the functioning of the filter can be seen in Fig. 5.

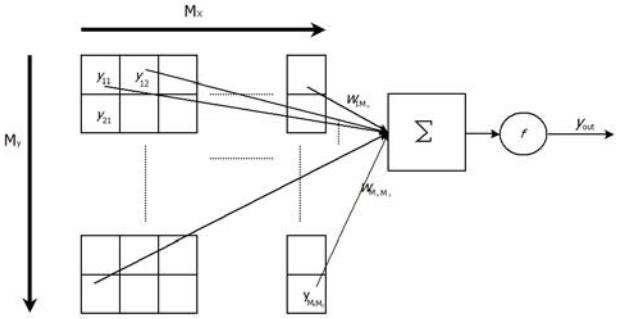

Figure 5. Schematic functioning of a filter for convolutional layer

The filter is applied on the entire previous convolutional map (see Fig. 6). Thus we will provide a new map whose values are given by the filter output.

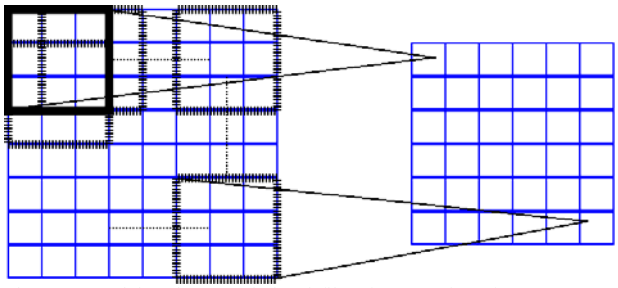

Figure 6. Applying the convolutional filter for a previous feature map. In the picture, the map has the 8 x 8 dimension, the convolutional filter is 3 x 3, and the resulting map is 6 x 6.

As it can be seen in Table I, the dimensions of the convolutional filters and their number can vary for each layer. For providing feature maps with equal dimensions on consecutive layers, we may add a border of pixels that will surround the input image of each filter. Due to the high number of pixels from the initial image, this procedure was not necessary anymore for this case. We will choose the number of filters for each layer according to the complexity of the problem and the time necessary for training the network. As a general rule, we mention that a higher number of filters will lead to better results, but the rule is applied only up to a certain threshold. Beyond this threshold, an increase in the number of filters does not affect anymore the performance.

B. Max-pooling layer

The max-pooling layer operating method selects the best features extracted by the network and at keeping the level of generalization of the network. The method of functioning for the filter is the same as the one presented in Fig. 6, but the output value is equal to the maximum value of its inputs. In our case, we will consider the input values to be 2 x 2 or 6 x 6 values, and the output value will be equal to the maximum value recorded.

For this case, the area of application of the filter is not overlapped on other areas, as in the case of convolutional layers. Consequently, the size of the filter should not be too large for not loosing information.

C. Classification layer

The classification layer provides the output values. The number of neurons in this layer is equal with the number of classification classes. In our case, there are two classes, and the output is presented as two real numbers whose sum is 1. Each of the two values represents the input probability (it must be multiplied by 100) of belonging to a class or another.

The previous layer is a fully connected one, because we desire to perform an analysis of all the features extracted by the network up to this point. Our network presents a number of 100 neurons on this layer which is fully connected to classification layer (2 neurons).

The activation function of a neuron from the classification layer is *softmax*, because it ensures an output value that represents the probability of the output to belong to a certain class.

III. EXPERIMENTS

A. Training the neural network

For training the neural network only the sections presenting special interest have been cut out from an X-ray image. The training was not performed for the entire image because of its resolution (2492 x 1984). It was performed on a NVIDIA GTX 560 graphic card due to its high speed for matrix-matrix multiplication operations (50 times faster than a standard processor). The program used for training the network was developed on the basis of a similar program written for training a convolutional network using the graphic cards. [21]

The program is written using two programming languages. Its central part is written in CUDA. It contains all the necessary steps to train the neural network and all the arithmetic routines needed for matrix-matrix multiplication. The outer shell, which deals with processing the files, building data sets and interpretation of results is written in Python.

The first stage consisted in forming the training set. For avoiding a high memory usage and for performing a rapid training, there have been organized 212 batches, each of them containing 128 training examples. Due to the large number of inputs - 16384, each of them being a full value between 0 and 255 (shades of grey), the size of a single batch was of 8.2MB.

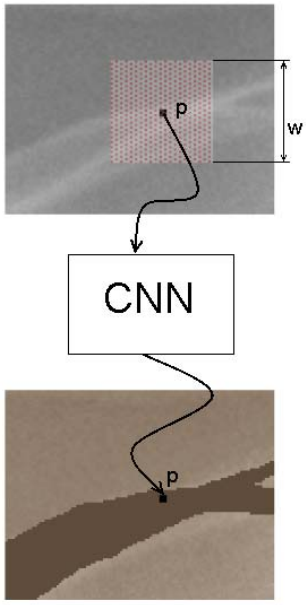

Figure 7. Determining the class of belonging for each pixel. For the present case, we will consider the *p* pixel and a window of *w*=128 pixels necessary for training the network.

The method of training can be seen in Fig. 7. Each pixel from the original image was extracted together with a rectangle window having the side *w*. The *p* pixel was centered inside this window for gathering all the information possible. For ensuring the symmetry of the method, the *w* parameter is recommended to be an odd number. In our case, because of the large size of the window ($w = 128$), this is not important. The important parameter was the hardware configuration of the graphic processor that facilitated an easier computing for the matrixes equal in size to the one used by us in the present procedure.

For training the network it was necessary to determine all its weights so that the resulting error to be minimum. As we mentioned in chapter 2, the convolutional network has a sequence of convolutional layers followed by the max-pooling layer. Each convolutional layer is characterized by its associated weights.

The weights of the filters for the convolutional layers were randomly set at the input. During the training they may change their value so that the resulting error to be minimum (the network should correctly recognize as many points as it can). In Fig. 8 there are shown the filters used on the first convolutional layer. We can easily distinguish some features that must be extracted by the first layer of filters.

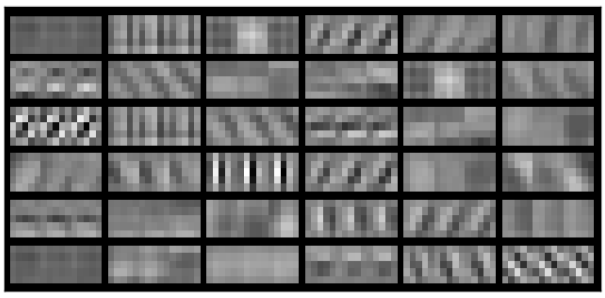

Figure 8. Filters used on first convolutional layer

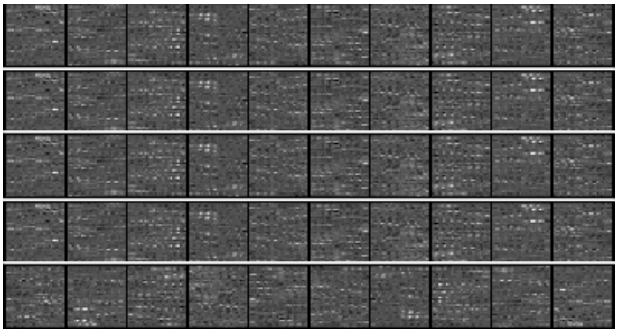

Figure 9. Filters used on the second convolutional layer

For the filter used on the second convolutional layer (Fig.9), it is more difficult to distinguish some graphic features. The same happens on the next layers as well. The explanation is that the features that are extracted by the neural network are more and more abstract and different from what we could easily view.

The neural network was trained by running 50 epochs for a set of images made up of 10 similar X-rays (chest area). We have chosen this area because it presents a certain regularity of the bone areas, but there are also places where it is difficult to differentiate the bone areas from the rest of the image.

For acquiring the best results, we recommend that the network should be trained and used for the same types of X-rays (the same physical area of the human body). This is necessary as the structure of the bones can differ, leading to different images for various areas of the human body.

For making a correct evaluation, we considered three error metrics that have been calculated for a testing set made up of 3 images:

Rand error - defined as 1- R_i , where R_i represents the Rand index[22]. Given two segmentations S_1 and S_2 of an image with *n* pixels, we define: *a* - number of pixels in the same class in both segmentations and *b* - number of pixels whose class differs from S_1 to S_2. Thus R_i is defined by formulae (2) and it measures the similarity between the 2 segmentation.

$$R_i = \frac{a+b}{\binom{n}{2}} \tag{2}$$

Warping error - is a metric for comparing boundary labeling based on concepts from the field of digital topology [23]. The warping error focuses on the objects and measures the topological differences between them (penalizes topological disagreements).

$$D\left(T \| L^*\right) = \min_{L \lhd L^*} \|T - L\|^2 \tag{3}$$

, where L^* represents a topology-preserving warped source segmentation.

Pixel error - is the simplest way of evaluating segmentation. It is computed as mean square between the original label and the segmented ones.

$$Pixel \quad error = \left\|L - L^*\right\|^2 \tag{4}$$

B. Results

During the experiments performed for obtaining a wide view upon the performances recorded by the network, we

varied the values of the *w* parameter (window size) on the input layer: 64, 96, 102 and 128. We had 4 resulting configurations (the final configuration was already presented in Table I).

TABLE II. CONFIGURATIONS OF THE CONVOLUTIONAL NETWORKS USED IN THE EXPERIMENTS

Layer no.(type)	Configuration 1	Configuration 2	Configuration 3	Configuration 4
0 (input)	1 map of 64x64 neurons	1 map of 96x96 neurons	1 map of 102x102 neurons	1 map of 128x128 neuron
1 (convolutional)	36 maps of 60x60neurons	36 maps of 92x92 neurons	36 maps of 96x96 neurons	36 maps of 122x122 neurons
2 (max pooling)	36 maps of 30x30 neurons	36 maps of 46x46 neurons	36 maps of 48x48 neurons	36 maps of 61x61 neurons
3 (convolutional)	36 maps of 28x28 neurons	36 maps of 42x42 neurons	36 maps of 44x44 neurons	36 maps of 58x58 neurons
4 (max pooling)	36 maps of 14x14 neurons	36 maps of 21x21 neurons	36 maps of 22x22 neurons	36 maps of 27x27 neurons
5 (convolutional)	36 maps of 9x9 neurons	36 maps of 16x16 neurons	36 maps of 18x18 neurons	36 maps of 24x24 neurons
6 (max pooling)	64 maps of 3x3 neurons	64 maps of 4x4 neurons	64 maps of 3x3 neurons	64 maps of 4x4 neurons
7 (fully connected)	100 neurons	100 neurons	100 neurons	100 neurons
8 (fully connected)	2 neurons	2 neurons	2 neurons	2 neurons

One of the results we wanted to follow in our experiments was the degree of covering the bone areas (pixel error). In Fig. 10 you can see the way in which the 4 networks managed to differentiate the bone area from the rest of the image, for a simple case. In this simple case, we tried to determine if CNNs are capable of differentiating some small dimension bone areas from a much larger image. We can easily notice that our network (configuration 4) managed the best covering at the level of pixel segmentation for this simple case.

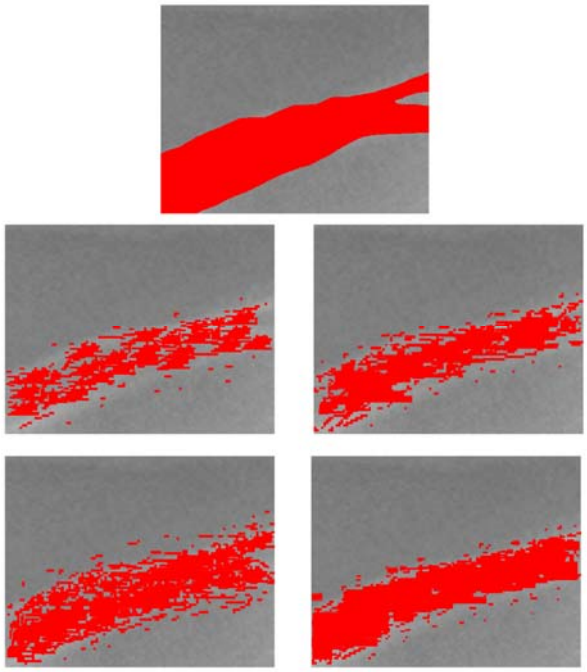

Figure 10. A simple segmentation of bone tissue made by each network. Above, the ground truth segmentation. Second row: from left to right: Network configuration 1, network configuration 2. Below: Network configuration 3, Network configuration 4

For the next set of experiments we had in view exactly the opposite of the previous experiment. Thus, we tried to determine the performances of the networks for distinguishing a *non-bone* area from a large bone area. For this case too, the results obtained by our network were the best. In Fig. 11, you can see how the segmentation proposed by our network manages to accurately determine the bone area. Moreover, the contour of the selected area has almost the same shape as the real one.

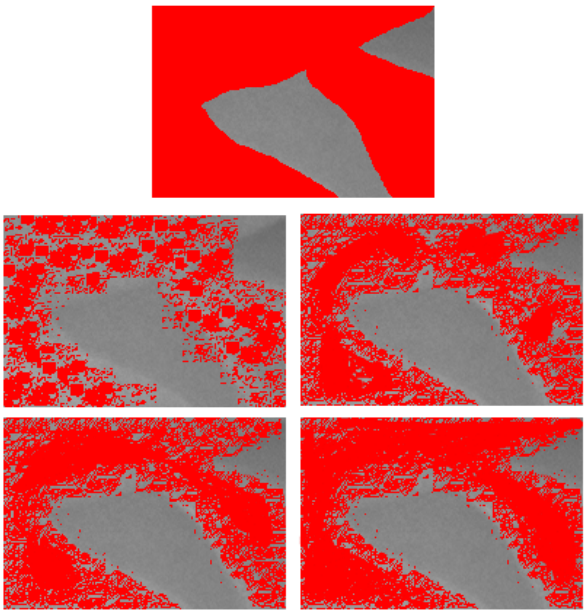

Figure 11. A more difficult segmentation of bone tissue made by each network. From top to bottom, left to right: the ground truth, configuration 1, configuration 2, configuration 3, configuration 4

The next target of our experiments was a correct extracting of the bone contour. Using the warping error as metrics, we tried to determine the CNN configuration that best recognizes the contour of the interest area from the X-ray. In what follows we will present the extraction of the bones contour performed by the 4 configurations for an area in the X-ray (Fig. 12). All the networks recognized the large bone areas, but problems appeared when the bone area presented irregularities. The first two configurations (configuration 1 and configuration 2) did not recognize the irregularities, they have partially been recognized by configuration 3 and they have totally been recognized by configuration 4. The only problems encountered by configuration 4 consisted in areas resembling in density to the bone tissue. Unfortunately, neither configuration 4 managed the segmentation of these areas.

One possible solution in order to solve this drawback is to obtain X-ray images that have superior image quality, better than our images and to use them for training the neural network.

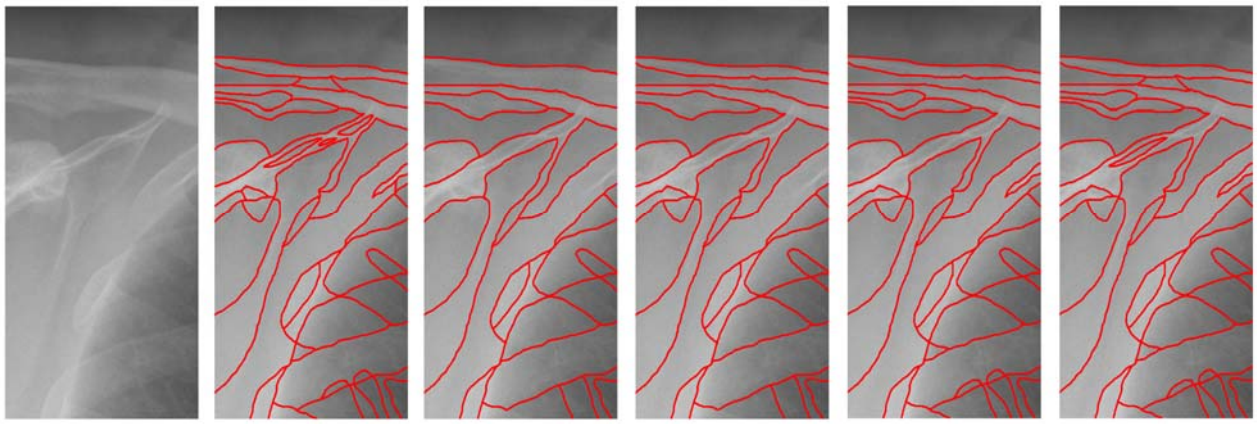

Figure 12. Bone contour extraction results for all four configurations. From left to right: original image, the ground truth segmentation, configuration 1, configuration 2, configuration 3, configuration 4

Overall, the results obtained in the experiments can be seen in Table III. Our network (configuration 4) obtained the best results for all the metrics taken into consideration in the segmentation process. The training time significantly increased for configuration 4 in comparison to configuration 1, considering that only certain sections from an X-ray were part from the training set.

TABLE III. THE PERFORMANCE AND THE TRAINING TIME FOR EACH NETWORK CONFIGURATION USED IN THE EXPERIMENTS

Performance	Configuration 1	Configuration 2	Configuration 3	Configuration 4
Pixel error [$x10^{-3}$]	904	436	312	204
Warping error [$x10^{-6}$]	878	631	490	305
Rand error [$x10^{-3}$]	123	89	78	45
Training time [for 1 epoch]	160 minutes	190 minutes	200 minutes	230 minutes

The last experiments tried to determine the accuracy of the neural network for segmenting the bone tissue in X-ray images of some areas of the human body totally different from the areas in the images from the training set. In these cases, the errors obtained by our network were twice higher than the errors from Table III.

In consequence, it is recommendable to apply other techniques than the ones described in this article or to determine a network configuration much larger than the one used by us. This target is hard to reach because of the extremely high training time and the necessity of running the experiments on better graphic cards.

An important issue for us is the execution time required to train the neural network. The time required for a single training epoch varies between 160 and 230 minutes. However, due to the neural network architecture, the training phase does not require many epochs. In our case, the total time to train the final network was 8 continuous days.

Nevertheless, running the program on next-generation GeForce 600 Series is one of the future steps the authors intent to do in order to improve the training time.

IV. Conclusions

The main strength of our approach consists in using a convolutional neural network as pixel classifier for the segmentation process of some X-ray images. The network analyzes each pixel from the image and tries to classify them in the 2 classes *{bone, non-bone}*. In this way, we try to separate the bone tissue area from the rest of the image.

The use of CNN makes our method a complete automatic one in contrast to other segmentation methods of X-ray images. We determined a CNN configuration and its performances were compared to other three networks of the same type, but with different configurations. For all the three considered metrics, i.e. pixel error, warping error and rand error, our network obtained the best results in contrast to other configurations. For ensuring a minimum training time of the network, we used only the interest areas from an image.

The last series of experiments focused on extracting an accurate contour of the bones. This method allows CNN to provide an exact delimitation of a specific bone and it is a completely different way of using the network for the X-ray images. The results were very good allowing the use of this type of neural network in order to obtain any contour of the bones from an X-ray image. [25]

The remarkable results obtained by the CNN open interesting perspectives for applying it to other medical images, too. In this regard, there have been performed experiments of segmenting some images from physical areas of the human body, others than the ones used in the images from the training set. The results obtained have not been satisfactory and they require running the experiments on a new generation of graphic cards.

As a final conclusion, for obtaining maximum performances, it is recommended to use the convolutional neural network for images that belong to the same category (in this case - X-rays for the chest area).

References

[1] K. Fukushima, "Neocognitron: A self-organizing neural network for a mechanism of pattern recognition unaffected by shift in position", Biological Cybernetics, vol. 36, pp. 193-202, 1980

[2] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-Based Learning Applied to Document Recognition", *Proceedings of the IEEE*, 86(11), pp. 2278-2324, November 1998

[3] P. Y. Simard, D. Steinkraus, J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis",

Document Analysis and Recognition, 2003, Proceedings. Seventh International Conference on , pp. 958- 963, 3-6 Aug. 2003

[4] J. Schmidhuber, M. Eldracher, B. Foltin, "Semilinear predictability minimization produces well-known feature detectors", Neural Computation, vol. 8 , pp. 773-786, 1996

[5] P. O. Hoyer, A. Hyvärinen, "Independent Component Analysis Applied to Feature Extraction from Colour and Stereo Images", *Network:Computation in Neural Systems*, vol. 11 , pp. 191-210, 2000

[6] M. Riesenhuber, T. Poggio, "Hierarchical models of object recognition in cortex", *Nature Neuroscience*, vol. 2, pp. 1019-1025, 1999

[7] D. Scherer, A. Müller, S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition", in *International Conference on Articial Neural Networks*, pp. 82-91, Springer, 2010

[8] D. Ciresan, U. Meier, J. Schmidhuber, "Multi-column Deep Neural Networks for Image Classification", in *Computer Vision and Pattern Recognition*, CVPR 2012, pp. 3642-3649

[9] D. Ciresan, A. Giusti, L. M. Gambardella, J. Schmidhuber, "Deep Neural Networks Segment Neuronal Membranes", in *Electron Microscopy Images*, NIPS 2012

[10] D. Strigl, K. Kofler, S. Podlipnig, "Performance and scalability of GPU-based convolutional neural networks", in *PDP '10 Proceedings of the 2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing*, pp. 317-324, 2010.

[11] U. Meier, C. D. Ciresan, L. M. Gambardella, J. Schmidhuber, "Better digit recognition with a committee of simple neural nets", in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pp.1250-1254, 18-21 Sept. 2011

[12] H. Hu, H. Liu, L. Chen, C. C. Hung, X. Xu, Z. Lan, "Image segmentation of cervical vertebra in X-ray radiographs using the curve fitting strategy", in *Proceedings of the 2011 ACM Symposium on Applied Computing (SAC '11). ACM,* New York, pp. 853-858, 2011

[13] E. Ganea, D. D. Burdescu, M. Brezovan, "New Method to Detect Salient Objects in Image Segmentation using Hypergraph Structure", *Advances in Electrical and Computer Engineering*, vol. 11, no. 4, pp. 111-116, 2011

[14] Q. Wei, H. Xiaolei, J. Yuanyuan, "Segmentation in noisy medical images using PCA model based particle filtering", in *Medical Imaging 2008: Image Processing, Proceedings of the SPIE*, vol. 6914, 2008

[15] B. N. Li, C. K. Chui, S. Chang, S.H. Ong, "Integrating spatial fuzzy clustering with level set methods for automated medical image segmentation", in *Computers in Biology and Medicine*, vol. 41, issue 1, pp. 1–10, 2011

[16] N. Boukala, E. Favier, B. Laget, P. Radeva, "Active shape model based segmentation of bone structures in hip radiographs", in *Industrial Technology, ICIT '04 IEEE International Conference on*, pp. 1682-1687, vol. 3, 2004

[17] O. Matei, "Ontology-Based Knowledge Organization for the Radiograph Images Segmentation", Advances in Electrical and Computer Engineering, vol. 8 (15), no. 1 (29), 2008

[18] A. Alvarenga de Moura Meneses, A. P. de Almeida, J. Soares, P. Azambuja, M. S. Gonzalez, S. Cardoso, D. Braz, C. E. de Almeida, R. C. Barroso, "Segmentation of X-ray micro-computed tomography using Neural Networks trained with Statistical Information: Application to biomedical images", in *Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2011 IEEE*, pp.3999-4001, 2011

[19] D. H. Hubel, T. N. Wiesel, "Receptive fields of single neurons in the cat's striate cortex", *Journal of Physiology*, vol. 148 , pp. 574-591, 1959

[20] D. H. Hubel, T. N. Wiesel, "Receptive fields, binocular interaction,and functional architecture in the cat's visual cortex", *Journal of Physiology*, vol. 160(1) , pp. 106-154, 1962

[21] A. Krizhevsky, "Convolutional deep belief networks on CIFAR-10", Technical report, University of Toronto, Aug. 2010.

[22] W.M. Rand, "Objective criteria for the evaluation of clustering methods", *Journal of the American Statistical association*, vol. 66(336), pp. 846–850, 1971.

[23] V. Jain, B. Bollmann, M. Richardson, D. R. Berger, M. Helmstaedter, K. L. Briggman, W. Denk, J. B. Bowden, J. M. Mendenhall, W. C. Abraham, K. M. Harris, N. Kasthuri, K. J. Hayworth, R. Schalek, J. C. Tapia, J. W. Lichtman, H. S. Seung, "Boundary Learning by Optimization with Topological Constraints", in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* , pp.2488-2495, 13-18 June 2010

[24] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies", in *A Field Guide to Dynamical Recurrent Neural* Networks, IEEE press, 2001.

[25] C. Cernazanu-Glavan, S. Holban, "Bone contour segmentation in radiograph images", Scientific Bulletin of "Politehnica" University of Timişoara, Romania, Transactions on Automatic Control and Computer Science, to be published