Scientific Research Publishing

Open Journal of Radiology, 2022, 12, 176-188
https://www.scirp.org/journal/ojrad
ISSN Online: 2164-3032
ISSN Print: 2164-3024

An Improved Neural Network Method for Forearm Bone Imaging Segmentation

Songzheng Huang[1], Jianfeng Chen[2*]

[1]Zhejiang Kangyuan Medical Devices Incorporation, Hangzhou, China
[2]Department of Radiology and Medical Imaging, Stritch School of Medicine, Loyola University Medical Center, Chicago, USA
Email: *jfchen@live.com

How to cite this paper: Huang, S.Z. and Chen, J.F. (2022) An Improved Neural Network Method for Forearm Bone Imaging Segmentation. *Open Journal of Radiology*, **12**, 176-188.
https://doi.org/10.4236/ojrad.2022.124018

Received: October 30, 2022
Accepted: December 9, 2022
Published: December 12, 2022

Abstract

In this paper, we propose several improved neural networks and training strategy using data augmentation to segment human radius accurately and efficiently. This method can provide pixel-level segmentation accuracy through the low-level features of the neural network, and automatically distinguish the classification of radius. The versatility and applicability can be effectively improved by learning and training digital X-ray images obtained from digital X-ray imaging systems of different manufacturers.

Keywords

Human Radius, Digital X-Ray Image, U-shaped *Unet* Neural Network, Segmentation

1. Introduction

In our previous study, a snake model algorithm is used to segment the image for computing the mean gray scale value over the region of interest in forearm bone, and the bone minerial density (BMD) value of radius located at a specific region could be determined by using radiation absorption method [1]. Compared with other traditional image segmentation methods, the snake model algorithm has the advantage of high accuracy. The digital X-ray images are binarized then the initial contours and segmentation can be determined directly by applying the snake model method. This method can achieve better segmentation than just directly applying the snake model method on original images in most cases, but it still has some drawbacks: 1) it is easy to fall into a local optimal state, so the algorithm sometime cannot converge correctly, then resulting in segmentation errors, 2) the method lacks a global vision, and cannot automatically identify the

ulna and radius classification, and manual intervention is usually required to select for subsequent segmentation processing; 3) the segmentation contours obtained by binarization could be changed with the selection of different regions of interest and different mean gray values; 4) the elastic energy, bending ability, gradient energy and other parameters of the Snake model can only be manually calibrated and cannot be learned automatically. Once calibration is done, the applicability of the algorithm is almost fixed.

In recent years, neural network-based image segmentation methods have been proposed in different application fields [2] [3] [4] [5] [6]. In this paper, several improved U-shaped *Unet* neural network models [3] [7] are used to replace the traditional image segmentation methods to automatically identify and segment radius in digital X-ray images. These methods can provide pixel-level segmentation accuracy through the low-level features of the neural network, automatically distinguishing between radius and ulna classifications. With less manual intervention, these improved methods can automatically locate to 1/3 of the radius recommended by Worldwide Health Organization (WHO) [8], and the measurement results are highly consistent. The versatility and applicability can be effectively improved by learning and training on digital X-ray images acquired from digital X-ray systems from different manufacturers.

2. Methods

2.1. Imaging Processing

Figure 1 is a flowchart of the algorithm used in this paper. A general digital X-ray imaging system was used as an image acquisition platform to acquire projection images of human non-dominant forearm bones (including radius and ulna). First, each input image is resized to have 512 × 512 pixels, then the image is fed into a neural network to identify segmentation. After the trained model and output layer, a softmax with 10 categorical feature images is obtained. Each classification feature image is marked with an specific serial number. For example, the image background is marked as 0, human skin is marked as 1, ulna is marked as 2, radius is marked as 3, other arm bones are market as 4, carpal bone is market as 5, and the phalanx is market is 6, aluminum ladder is 7, imitation arm skin is 8, and the imitation arm bone is 9. Each classified feature image outputted by the neural network is binarized to find the contour of the target object. Here, through a specific binarization process, it is determined that each pixel of the feature image has a specific classification number. If it is the corresponding serial number, fill the pixel at the corresponding position of the feature map with 255, otherwise fill with 0, thus constructing an 8 bits depth image (its image size is the same as the original 16 bits image). Each feature image corresponds to a binarized image, so 10 binarized feature maps are created. OpenCV contour search algorithm is applied to these binarized maps to find their contours, and finally these contours are recorded in the corresponding data structure for later use.

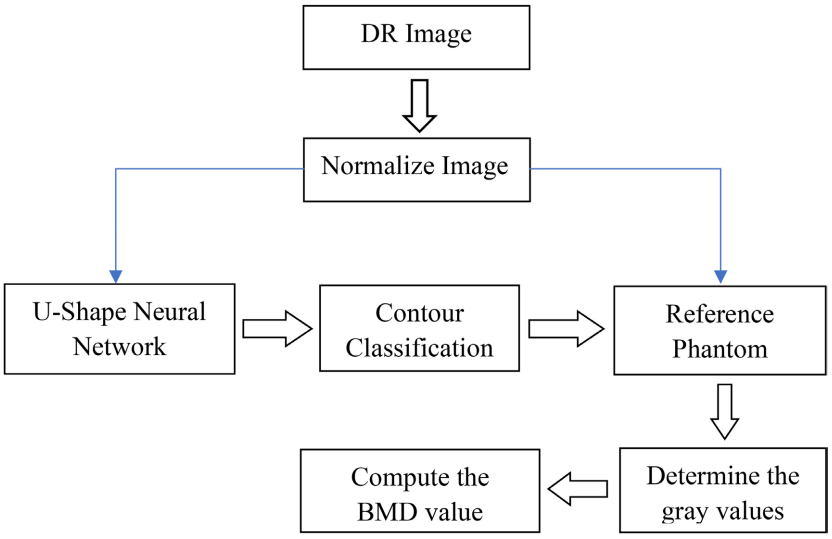


Figure 1. The main process of the algorithm.

After obtaining the contour line data, on the one hand, according to the obtained radius contour line, mark it as 3 on the map, find the minimum circumscribed rectangle of the radius contour line, and calculate the length of the minimum circumscribed rectangle, as shown in **Figure 2**. The position of the 1/3 distal of the radius, which is the WHO recommended site for BMD measurement. On the other hand, a 16-bit template image with the same size as the original image is created. The pixels inside the contour are set to 0xffff and pixels outside the contours are set to zero. The original image is used to perform a logical bitwise AND operation with these template images, respectively. The mean of the pixel values of the original image within each specific area by using the above segmentation method, combined with the automatically positioned at 1/3 of the radius, can be determined. Finally, based on the reference phantom method [1], the multi-level gray scale values of the reference phantom are fitted to calculate the BMD value in the region of interest.

2.2. Neural Network Modeling

Choice of Neural Networks

Although the anatomy of the human forearm bone is relative simple and similar in shape, the radius and ulna can often cover a large area in an X-ray extremity image. In order to get accurate radius BMD results, accurate pixel-level bone contour positions are required. At the same time, all feature information of the image categories is also important, including low-level features, which refer to binarized features and visual features within the size range of the image kernel, such as contours, edges, textures, shapes, etc., as well as high-level information, which refers to the different types of objects understood by the human brain after visual recognition.

The BP (Back Propagation) network model is a well-known multi-layer training model. When using BP network to segment an image, firstly, all pixels in the

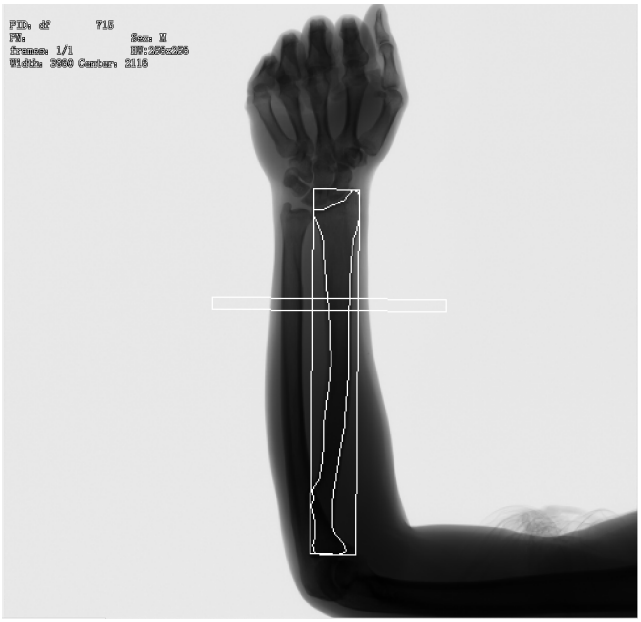


Figure 2. Mask manipulation and automatic positioning.

image are divided into target pixels and non-target pixels. After that, the non-target pixels are removed and the retaining pixels form the target image. But in our case, we not only need to obtain low-level features of the radius and ulna to obtain pixel-level gray scale value for BMD calculation, but also need to involve high-level semantics, such as classification and recognition of the radius, ulna, soft tissue, and objects other than human body. For efficiency, we need to predict all classifications at once in our application. The application of BP network in image segmentation has problems such as slow learning speed and easy to fall into local optimum [9]. So we gave up.

CNN (Convolutional Neural Network) has many advantages over general neural network, such as 1) CNN can better adapt to the structure of the image; 2) the extraction and classification operations can be performed at the same time, and feature extraction is conducive to classification; 3) weight sharing can reduce the training parameters of the network, making the network structure simple and adaptable. The skip connection structure (feature stitching) of the U-shaped network （based on CNN）not only has pixel-level feature recognition, but also retains a large field of view to provide high-level semantic features, which is especially suitable for our application [3].

Improvements of Neural Networks

In order to reduce the hardware requirements of the model, we need to reduce the number of parameters of the model. Based on the U-shaped network [3] [7], we simplify each stage, including down-sampling and up-sampling, from the original two convolution operations to one operation. At the same time, the number of input filters in the first-layer of the network is reduced from 64 to 32, which can greatly reduce the amount of parameters, as shown in **Figure 3** and **Table 1**. Each down-sampling layer in the original network is changed to a convolutional

Table 1. The results of training record with the original and modified neural networks.

Network	Description	Total Parameters	Training Set Error	Training Set Accuracy	Validation Set Error	Validation Set Accuracy	Stop Learning Epoch
(a) *UNet*	Original U shape network with preprocessing dataset	7,759,686B	0.07306	0.9789	0.1218	0.9639	80
(b) *UNet_1bn*	Add one Batch Normalization	7,759,690B	0.04543	0.9835	0.0505	0.9816	81
(c) *sUNet_1bn*	Simplified to one convolution each layer & Add one Batch Normalization	3,832,326B	0.06771	0.9764	0.0981	0.9626	80
(d) *sUNet_1bn_dilation12312*	use dilated convolution instead of convolution in *sUNet_1bn*	3,832,326B	0.06220	0.9779	0.1039	0.9664	71

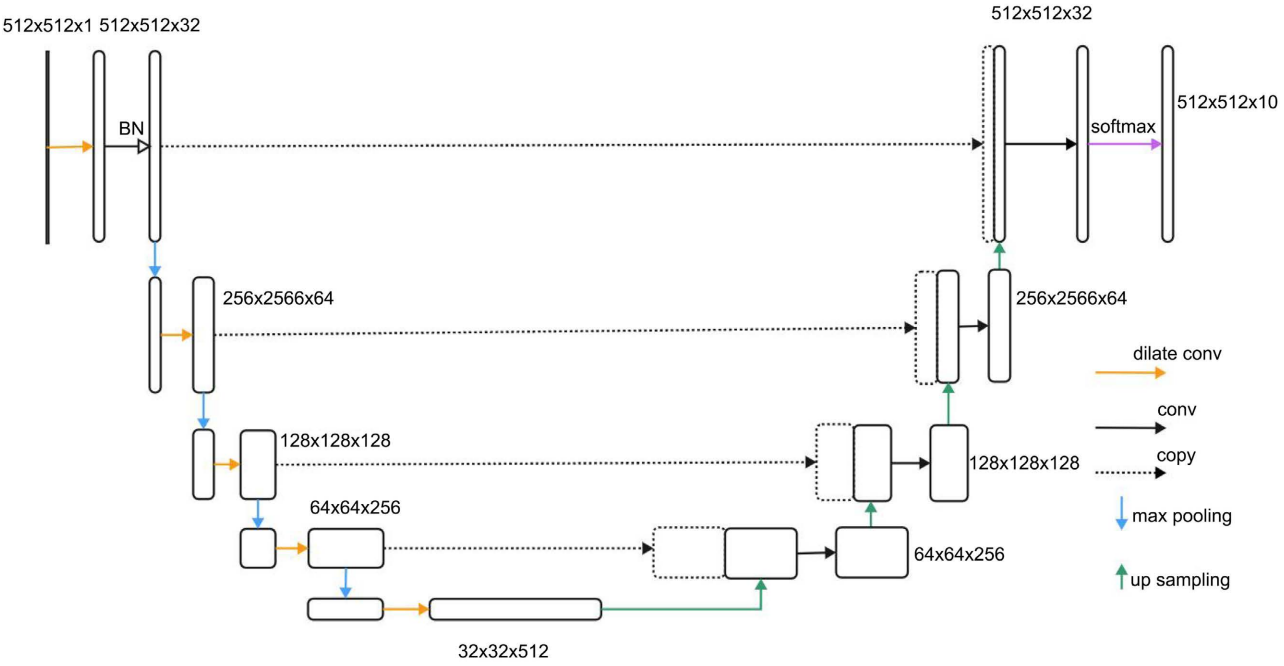


Figure 3. Simplified schematic diagram of U-shaped network (*UNet*).

sublayer (relu activation) with a kernel of 3 (in fact, we compared different kernels, and their effects were slightly different, but in the end we chose a kernel size of 3, which balances the impact of computational overhead. See Table 2). and max pooling sublayer consistency. The parameter amount of the first-stage convolution neural network is calculated as kernelWidth × kernelHeight × Channel × Number of Filters × Number of convolution operation. Therefore, the parameters of the first-stage of the UNet network are reduced from 3 × 3 × 1 × 64 × 2 to 3 × 3 × 1 × 32 × 1.

Because the number of convolution layers is reduced from two to one each stage, the receptive field is reduced. To preserve the receptive field, we set the down-sampling convolution layer to dilation (atrous) convolutions. The schematic

Table 2. The results of training record with different kernel sizes.

Network	Description	Total parameters	Training Set Error	Training Set Accuracy	Validation Set Error	Validation Set Accuracy	Stop Learning Epoch
sUNet_1bn_dilation 32123_k5	Dilation rates are set to 3, 2, 1, 2, 3: and the fifth kernel size is set to 5.	5,929,478B	0.04606	0.9832	0.0625	0.9768	61
sUNet_1bn_eca_dilation 842_k35	Dilation rates are set to 8, 4, 2: The fourth kernel is set to 3, while the fifth kernel size is set to 5.	5,929,483B	0.05303	0.9808	0.0585	0.9800	72
sUNet_1bn_dilation 12312_k5	Dilation rates are set to 1, 2, 3, 1, 2: and the fifth kernel size is set to 5.	5,929,478B	0.06480	0.9769	0.0851	0.9694	57
sUNet_1 bneca_dilation 12312_k97533	Dilation rates are set to 1, 2, 3, 1, 2: and the kernels are set to9, 7, 5, 3, 3.	3,916,262B	0.06055	0.9786	0.0859	0.9682	61
sUNet_1 bneca_dilation 12312_spatial	Dilation rates are set to 1, 2, 3, 1, 2: and the fifth stage use spatial attention.	3,832,907B	0.0737	0.9744	0.1098	0.9670	52

diagram of dilation convolution is shown in Figure 4. The formula for calculating receptive field of ordinary convolution is as follows:

$$RF_{n+1} = RF_n + \left(K_{n+1} - 1\right) * \prod_{i=1}^{n} S_i \tag{1}$$

where RF_{n+1} represents the receptive field of the current layer, and RF_n represents the receptive field of upper layer. K_{n+1} represents the kernel size of the current layer. S_i represents the stride of the upper layer. We use stride = 1 here for all layers. So the receptive field of the fifth layer of the standard *UNet* is 21.

In order to avoid the gridding effect [10], we set the dilation rate to different values (we also compared different dialition rates, and they have slightly different effects. See Table 2), the first and fourth layers are 1, the second and fifth layers are 2, the third layer is 3, so the receptive field of the fifth layer is 19.

In order to facilitate mask processing, we pad the boundary of the each payer's input image, as shown in Figure 5, so that the output feature image obtained after convolution remains the same size as the input image. Suppose the image size after convolution is $n * n$, the size of the convolution kernel is $k * k$, where k is an odd number, the padding amplitude is set to $(k-1)/2$, and the size of output image after convolution is $n - k + 2 * \left(\frac{k-1}{2}\right) + 1 = n$. That is, the size of output image after the convolution operation still is $n * n$, ensuring that the size before and after the convolution remains unchanged. After 4 times of down-sampling (the size of the feature map becomes half of the original image) and 4 times of up-sampling, the final output classification feature map size of the network has

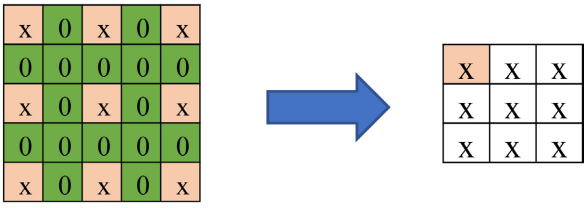


Figure 4. Dilated convolution.

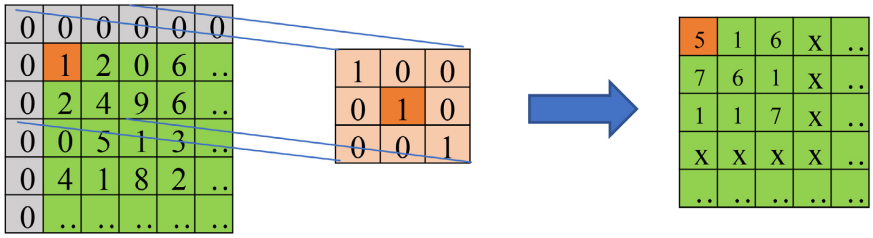


Figure 5. The schematic diagram of edge padding.

the same size as the input image. Finally, the desired pixel values are determined by masking the feature map and the original image.

In the first down-sampling layer, batch normalization is added between the convolution and the max pooling sublayers. This normalized sublayer is used to speed up network convergence. If this layer is not added, the training will be difficult to succeed (see **Table 1** for the training effect), since the sample brightness distribution is not uniform.

The input data is normalized to have a mean of zero and a variance of one. Its mathematical expressions are given in Equations (2), (3), and (4) below:

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{2}$$

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m}\left(x_i - \mu\right)^2 \tag{3}$$

$$x_i' = \frac{x_i - \mu}{\sqrt{\sigma^2 + \varepsilon}} \tag{4}$$

where x_i represents the input raw data $\{x_1, x_2, x_3, \cdots, x_m\}$, x_i' represents the data normalized by the BathNormalization layer. In Equation (3), ε is the small value to prevent the denominator from being zero. Due to the limit number of samples used in our experiments, data augmentation was used to simulate increasing the dataset to improve generalization application ability and avoid overfitting. The BatchNormalization layer will then scale and translate the normalized data. The mathematical expression for the scaling and translation is as follows:

$$y_i = \alpha x_i' + \beta \tag{5}$$

where y_i is the transformed dataset, α and β are learnable parameters, initialized to 1 and 0, respectively, which can be adjusted to appropriate values through learning and training process.

The output layer goes through a 10-channel feature convolution layer with a kernel of 1, and finally outputs 10 classification feature images through softmax; this layer will perform a cross-entropy error operation with the pre-labeled supervised data t_n, and then back-propagate the network to learn and correct the parameters of each layer; the hybrid cross-entropy E error of this neural network is expressed as

$$E = -\frac{1}{n}\sum_{0}^{n-1}\sum_{0}^{k-1} t_{nk} \log_e^{y_{nk}} \tag{6}$$

where t_{nk} represents the *k-th* feature element of the *n-th* supervised classification image, and y_{nk} represents the *k-th* feature element of the *n-th* feature output image. Through this error feedback, the neural network can accurately learn the classification label of each pixel in each supervised image.

As shown in Figure 6, the input of the tested neural network is a 16 bits $n\times512\times512\times1$ (n,w,h,c) image dataset: a training sample batch n with a sample width of 512 and a sample height of 512, the number of sample channels is 1. The output is a set of 10 categorical feature images of the same size $y_n=(n,w,h,10)$. The ten categories were described in previous section. The radius marked as 3 is the feature used in this algorithm to classify the image. The training image set is pre-collected with 400 original images and 32 validation images. The "labeled multiclass images" in Figure 6 are images that are manually labeled and classified on the basis of these original images.

The specific method of labeling and classifying images is: the gray value of the image is filled according to the value corresponding to each classification, such as the background is 0, the skin is 1, the radius is 3, etc., so that the grayscale value of each classified tissue labeled classified images varies from 0 to 9. These labeled images are processed into one_hot form after being read during training. For example, a pixel in radius marked as class 3 is represented as [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]. These raw images and labeled multi-class images are fed into the neural network, and the network is trained to converge according to Equation (6).

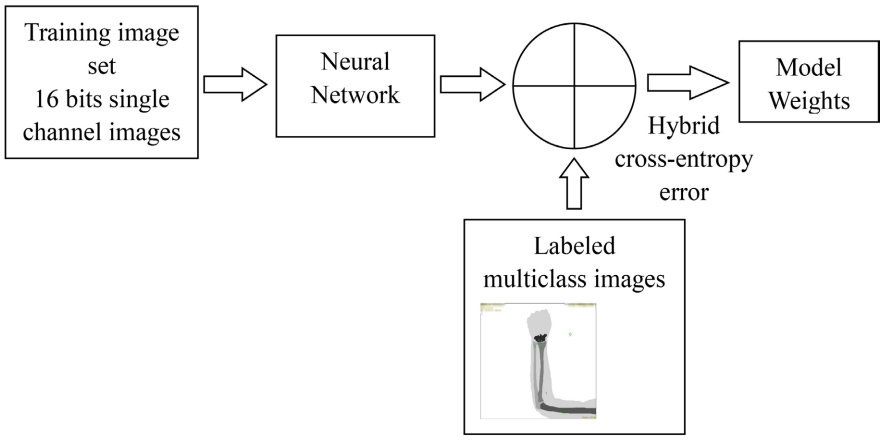
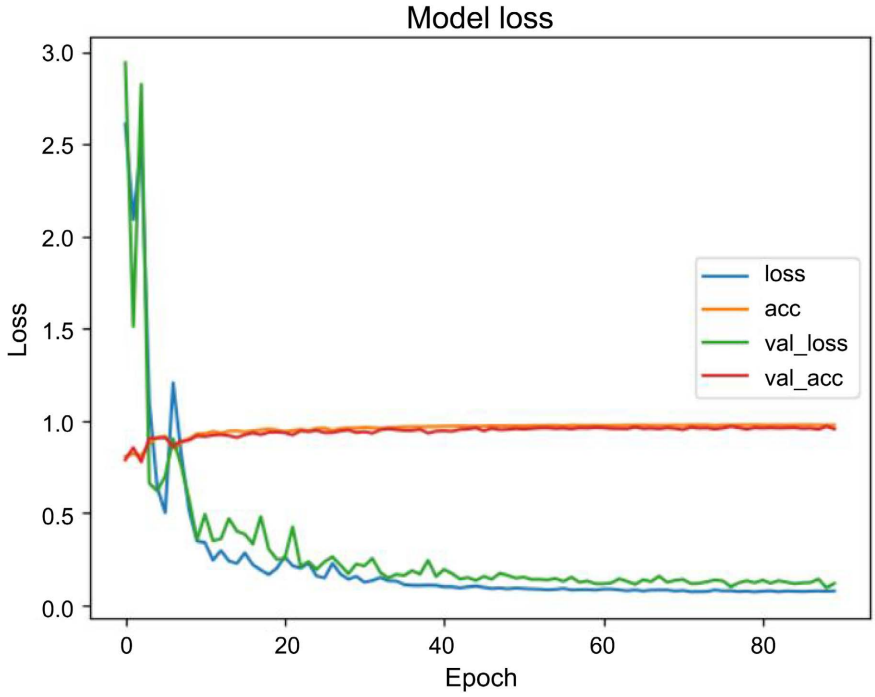
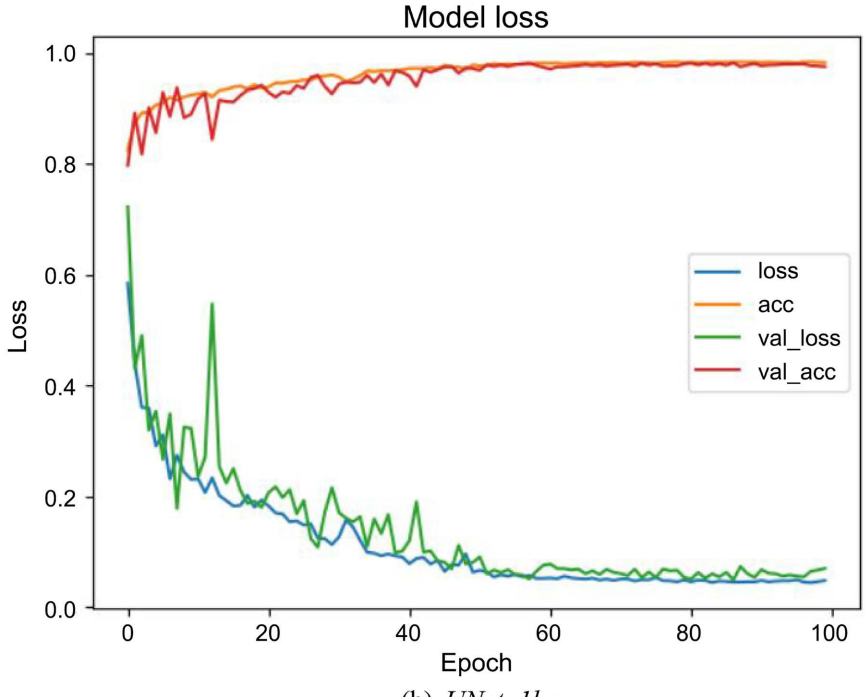


Figure 6. The training diagram (without deep-supervision).

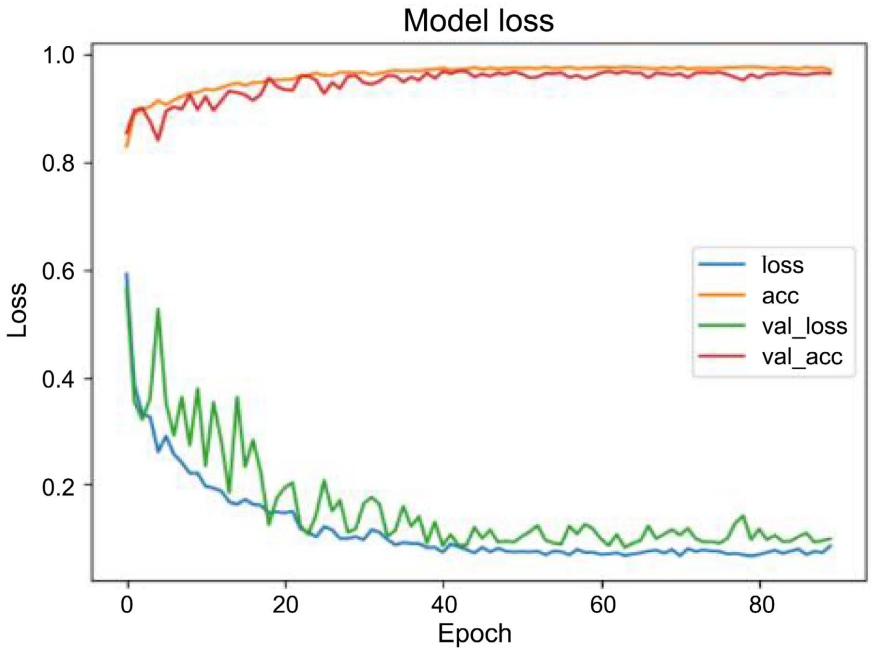
Figure 7 shows the original and the improved loss accuracy curves after 100 epochs of training for different versions of the *UNet* neural network. Here *UNet_1bn* is to add one Batch Normalization, *sUNet_1bn* is simplified to one convolution each stage and adds one Batch Normalization, *sUNet_1bn_dilation12312* uses dilated convolution instead of the convolution in *sUNet_1bn*. Table 1 records the accuracy results for the last best state, *i.e.* stop learning. Network training



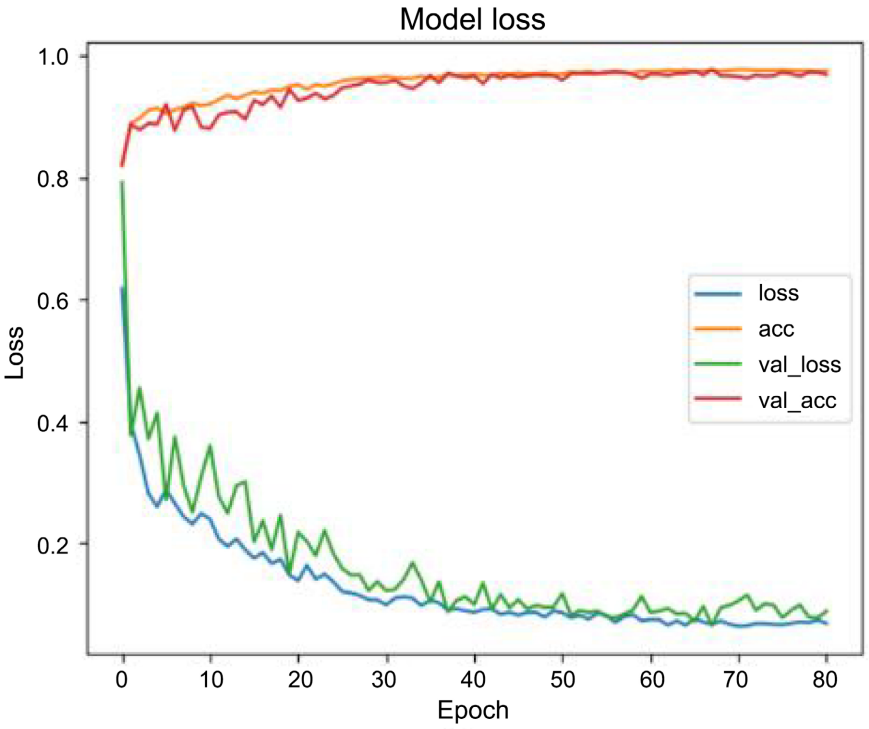
(a) *UNet*



(b) *UNet_1bn*



(c) *sUNet_1bn*



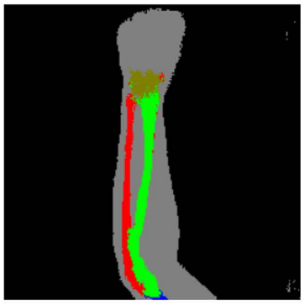
(d) *sUNet_1bn_dilation12312*

Figure 7. The original and improved loss-accuracy curves of various versions of the *UNet* neural network after 100 rounds of training: (a) the original *UNet* network, (b) the modified *UNet_1bn*, (c) modified *sUNet_1bn*, and (d) modified *sUNet_dilated*.

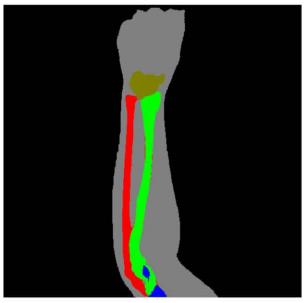
sets the starting learning rate to $1 \times e^{-4}$ and the ending learning rate to $1 \times e^{-8}$. Due to the limitation of hardware conditions such as video memory, *UNet* only uses 6 images for each batch for training and verification.

The test environment settings for our experiment were as follows:
windows10_21H1_19043.1766;
Intel(R) Core(TM) i7-8750H CPU @ 2.20 GHz 2.21 GHz;
GeForce GTX 1060, 6 GB, Driver Version:436.48, CUDA Version: 10.1;
python3.5.6;
Numpy1.15.2;
Scipy1.1.0;
H5py2.8.0;
Opencv-python4.4.0;
Keras2.22.2;
tensorflow_gpu 1.10.0;

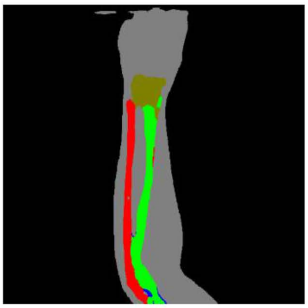
The original X-ray image



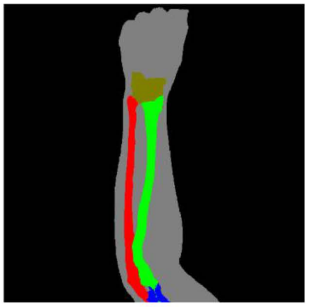
(a) *UNet*



(b) *UNet_1bn*



(c) *sUNet_1bn*



(d) *sUNet_1bn_dilation12312*

Figure 8. The test results of the segmentation by using the original and modified Neural Networks after 100 epochs of training (use early stop).

The segmentation results are shown in Figures 7(a)-(d). The results show that the total numbers of parameters used for the *UNet* with the improved networks, which include both *sUNet_1bn* and *sUNet_1bn_dilation12312*, have been reduced from 7,759,690B to 3,832,326B. However, the accuracy and cross entropy error are not much lower than the original network, which can meet our application needs. The segmentation results using these networks are shown in Figure 8.

3. Discussions

One of the main goals of this paper is to obtain a network with few parameters and comparable performance. White some combinations may be better in accuracy than the network we ultimately choose, they also have multiple hardware costs. Due to the limited number of samples, it is not possible to include all X-ray images in these acquisition cases, so the trained network may not be fully applicable to all cases.

The original *UNet* network used for comparison failed to converge correctly under the same conditions. After normalizing the training set image, we got the UNet Training curve shown in Figure 7(a).

4. Conclusion

For the segmentation of human forearm bones, especially the radius, we have improved and adapted the *UNet* network, added the normalization preprocessing before fide to the neural network, and modified the specific convolution sublayer, so that it can provide satisfactory high-level semantic classification capabilities and pixel level segmentation performance, while reducing the number of parameters, which can prove the degree of automation and reduce manual intervention to meet the needs of the scene.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Chen, J., Fan, Y., Li, P. and Huang, S. (2018) Clinical Application of Intelligent Radio Absorptiometry Measurement of Human Arm Bone Mineral Density and Assessment of Osteoporosis. *Chinese Medical Devices*, **33**, 31-35.

[2] Badrinarayanan, V., Kendall, A. and Cipolla, R. (2017) SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39**, 2481-2495. https://doi.org/10.1109/TPAMI.2016.2644615

[3] Shelhamer, E., Long, J. and Darrell, T. (2017) Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39**, 640-651. https://doi.org/10.1109/TPAMI.2016.2572683

[4] Chen, L., Papandreou, G., Kokkinos, I., Kevin, M. and Yuille, A. (2018) DeepLab:

Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **40**, 834-848. https://doi.org/10.1109/TPAMI.2017.2699184

[5] Chen, L., Zhu, Y., Papandreou, G., Schroff, A. and Adam, H. (2018) Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C. and Weiss, Y., Eds., *Computer Vision—ECCV* 2018. *Lecture Notes in Computer Science*, Springer, Cham. https://doi.org/10.1007/978-3-030-01234-2_49

[6] Liu, S., Huang, D. and Wang, Y. (2018) Receptive Field Block Net for Accurate and Fast Object Detection. In: Ferrari, V., Hebert, M., Sminchisescu, C. and Weiss, Y., Eds., *Computer Vision—ECCV* 2018. *Lecture Notes in Computer Science*, Springer, Cham, 404-419. https://doi.org/10.1007/978-3-030-01252-6_24

[7] Ronneberger, O., Fischer, P. and Brox, T. (2015) U-Net: Convolution Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W. and Frangi, A., Eds., Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015. 18th International Conference, Munich, Germany, Proceedings, Part III.

[8] WHO Study Group (1994) Assessment of Fracture Risk and Its Application to Screening for Postmenopausal Osteoporosis. *World Health Organization Technical Report Series*, Geneva.

[9] Wang, S., Jiang, J. and Lu, X. (2020) Advances on Tumor Image Segmentation Based on Artificial Neural Network. *Journal of Biosciences and Medicines*, **8**, 55-62. https://doi.org/10.4236/jbm.2020.87006

[10] Wang, Y., Dong, M., Shen, J., Lin, Y. and Pantic, M. (2022) Dilated Convolutions with Lateral Inhibitions for Semantic Image Segmentation. Cornell University Library, Ithaca, arXiv.org.