Melanoma segmentation based on deep learning

Xiaoqing Zhang
College of Information Science and Technology, Donghua University, Shanghai, China

ABSTRACT
Malignant melanoma is one of the most deadly forms of skin cancer, which is one of the world’s fastest-growing cancers. Early diagnosis and treatment is critical. In this study, a neural network structure is utilized to construct a broad and accurate basis for the diagnosis of skin cancer, thereby reducing screening errors. The technique is able to improve the efficacy for identification of normally indistinguishable lesions (such as pigment spots) versus clinically unknown lesions, and to ultimately improve the diagnostic accuracy. In the field of medical imaging, in general, using neural networks for image segmentation is relatively rare. The existing traditional machine-learning neural network algorithms still cannot completely solve the problem of information loss, nor detect the precise division of the boundary area. We use an improved neural network framework, described herein, to achieve efficacious feature learning, and satisfactory segmentation of melanoma images. The architecture of the network includes multiple convolution layers, dropout layers, softmax layers, multiple filters, and activation functions. The number of data sets can be increased via rotation of the training set. A non-linear activation function (such as ReLU and ELU) is employed to alleviate the problem of gradient disappearance, and RMSprop/Adam are incorporated to optimize the loss algorithm. A batch normalization layer is added between the convolution layer and the activation layer to solve the problem of gradient disappearance and explosion. Experiments, described herein, show that our improved neural network architecture achieves higher accuracy for segmentation of melanoma images as compared with existing processes.

1. Introduction
Malignant melanoma is one of the deadliest forms of cutaneous cancer, and one of the world’s fastest-growing cancers. In the United States, approximately 76,690 new cases of newly invasive melanoma were diagnosed in 2013, resulting in an estimated 9,480 deaths. In Australia, more than 1,890 deaths occur due to skin cancer each year [1]. In terms of pricing, it has been estimated that the cost of medical treatment for non-melanoma skin cancer is $264 million and for melanoma it is $30 million. Early diagnosis and treatment of melanoma is critical; early treatment can achieve a nearly 95% cure rate. At the same time, dermatologic data show that even in professional institutions, accuracy of visual diagnosis alone is not high. Our study aims to present an optimal neural network structure that can enable automated means to become the basis for a broad and accurate diagnosis of skin cancer. Dermatoscopy is one of the most important tools for noninvasive skin imaging techniques to diagnose melanoma and other pigmentation skin lesions. It minimizes skin reflections through optical amplification and related devices, making it easier for experts to observe the organizational structure. This reduces screening errors and provides greater differentiation between difficult-to-understand lesions, such as pigmented spots, and small clinically unknown lesions, to improve diagnostic accuracy.

2. Previous work
Most image segmentation techniques utilize classical machine-learning processes, and extract features according to machine-learning methods while classifying features via specialized training programs. Previous literature referred to important techniques for early diagnosis of melanoma. For example, Green, et al. proposed the use of shape, color and skin damage boundaries as features for technological classification [2]. Lee et al. have developed a skin cancer diagnosis system which contains morphological features, and a
classification stage that implements artificial neural networks [3]. Aitken et al. classified the lesion area, perimeter, border, wavelength, and gradient [4]. Chang et al. proposed a heuristic method for feature extraction and lesion identification [5]. She et al. classified melanoma skin lesions according to characteristics such as symmetry, border, color variation and diameter [6]. And Fassihi et al. used morphological operators and feature extraction via a wavelet transform technique to achieve melanoma segmentation [7].

Deep neural networks have the ability to learn additional complex feature hierarchies from the data, and perform well at the learning level. Convolution neural networks (CNN) are suitable for image data segmentation, and were proposed by LeCun et al. in 1998 [8]. CNNs suppose that the inputs are images, to allow encoding of their properties in the architecture. This feature reduces the complexity of the network architecture for faster processing. Davy [9] and Zikic [10] et al. used 2D or 3D patches to train CNNs to predict the central pixel classes of medical images. In 2014, Urban [11] and Zikic et al. used a common CNN to achieve segmentation, consisting of a series of convolution layers, and a non-linear activation function between each layer and the softmax output layer. In 2014, Pinheiro and Collobert [12] used a basic CNN to predict each pixel, and improved the forecast by improving CNN. In 2013, Farabet [13] et al. proposed several different CNNs for image processing at different resolutions, and completed the pixel-level prediction. By regularization of the global super-pixel segmentation of the image, the smooth segmentation was finally realized. T. Brosch et al. proposed different CNN architectures [14,15], eliminating the overlapping patch calculations. Kang et al. introduced a complete convolution neural network (CNN) to monitor crowd segmentation in video series [15]. Long et al. [16] proposed the fusion of subnets generated by the lower layer of the network to the upper-generated upsampling segments. Ronneberger et al. [17] proposed the combination of different layers, using the u-net architecture to calculate the final segmentation directly at the lowest level, which replaces the pooling layer by an up-sampling operation, but still cannot completely solve the loss problem. In the field of medical imaging, in general, the use of CNN to segment images is relatively small. However, in 2013, Huang and Jain [18] used CNN to predict the boundaries of nerve tissue in electron microscopy images. We use the improved CNN framework to characterize and segment melanoma images. Among them, the neural network includes multiple convolution layers, dropout layer, softmax layer, multiple filters and activation function.

Using a non-linear activation function (ReLU and ELU) effectively mitigates the problem of gradient disappearance. Using RMSprop and Adam algorithm enables optimization of the loss algorithm. Between the convolution layer and the activation point, a batch normalization layer is added to solve the problem of gradient disappearance and explosion.

3. Method

3.1 The decision function

Neural network learning is done by allotting a set of training samples \((x_i, y_i), 1 \leq i \leq N\), \(x_i\) is handled by the input scalar, \(y_i\) is the target to be predicted, allowing the algorithm to seek \(f(\cdot)\) functions to establish the relationship between \(x\) and \(y\), with \(\hat{y}\) being a model \(\hat{y} = f(x, \theta)\) as an output, and \(\theta\) is a function parameter. A loss function \(L(y, f(x, \theta))\) can be defined to evaluate the risk of decision making on all training samples:

\[
R(\theta) = \frac{1}{N} \sum_{i=1}^{N} L(y^{(i)}, f(x^{(i)}, \theta))
\]

(1)

The principle of minimizing the risk of experience is to obtain the corresponding parameters by gradually converging to the minimum expected value.

3.2 Overfitting

As the training sample is only a small part of the real data, it cannot respond well to the real data distribution. Therefore, the minimization of experience risk is likely to lead to over-fitting, which is mainly due to the lack of training data. The generalization error is a measure of whether the learning model is well-generalized to unknown data. The scheme to solve the problem of overfitting is generally to add the parameterized regularization on the basis of minimizing the experience risk, which can reduce the parameter space:

\[
\theta^* = \arg\min_{\theta} R(\theta) + \lambda ||\theta||_2^2
\]

(2)

\[
= \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} L(y^{(i)}, f(x^{(i)}, \theta)) + \lambda ||\theta||_2^2
\]

where \(||\theta||_2\) is the regularization of the norm \(L_2\), and \(\lambda\) is used to control the regularization of the intensity.

3.3 Cross entropy loss function

For the classification problem, the prediction target \(y\) is a discrete category, and the model output \(f(x, \theta)\) is
the conditional probability for each class. \( f_{\theta}(x) \) can be viewed as a category of the likelihood function. The parameters can be optimized by maximum likelihood estimation, with a commonly-used negative logarithmic likelihood loss function:
\[
L(y, f(x, \theta)) = -\log f_{\theta}(x)
\] (3)

### 3.4 Gradient descent method

The input data set training process is via a learning algorithm for automatic learning parameters of the process, our commonly used parameter learning algorithm is a gradient descent method. The iterative formula of the gradient descent method is:
\[
a_{t+1} = a_t - \lambda \nabla a_{t} = a_t - \lambda \nabla f(a_{t})
\] (4)

where \( \lambda > 0 \) is the search step in the gradient direction. We obtain the parameters \( \theta \) by learning, in order to minimize the risk function:
\[
0^* = \arg \min_{\theta} R(\theta) = \arg \min_{\theta} \frac{1}{N} \sum_{i=1}^{N} L(y^{(i)}, f(x^{(i)}, \theta))
\] (5)

Learning parameters by gradient descent method are achieved using:
\[
a_{t+1} = a_t - \lambda \frac{\partial R(\theta)}{\partial a_{t}} = a_t - \lambda \sum_{i=1}^{N} \frac{\partial R(\theta; x^{(i)}, y^{(i)})}{\partial \theta}
\] (6)

where \( \lambda \) is termed the learning rate. The random gradient drop (SGD) is also called the incremental gradient drop, which is an improved gradient descent method, and SGD is updated for each sample:
\[
a_{t+1} = a_t - \lambda \frac{\partial R(\theta; x^{(i)}, y^{(i)})}{\partial \theta}
\] (7)

where \( x^{(i)}, y^{(i)} \) is the sample selected for the t-th iteration.

### 3.5 Learning rate

In the gradient decline, the learning rate when too large will not converge, and when too small will lead to a slow convergence rate or even a lack of convergence. The momentum method is the addition of the last iteration to the current iteration update; we note: \( \nabla \theta_t = \theta_t - \theta_{t-1} \). In the t-th iteration, \( \theta_t = \theta_{t-1} + (\rho \nabla \theta_t - \lambda g_t) \), where \( \rho \) is the momentum factor, and is usually set to a value of 0.9. At the beginning of the iteration, the previous gradient is used to accelerate the process, when near the convergence value of the iteration, because the two update directions are opposed, increasing the stability. The AdaGrad algorithm is a reference to the idea of L2 regularization. In the t-th iteration:
\[
\theta_t = \theta_{t-1} - \frac{\rho}{\sqrt{\sum_{t=1}^{t} g_t^2}} g_t
\] (8)

where \( \rho \) is the initial learning rate, and \( g_t \in \mathbb{R}^d \) is the gradient of the t iteration. As the number of iterations increases, the gradient decreases.

### 3.6 Softmax regression

Softmax regression is a common learning algorithm for multiple linear classification functions. The posterior probability of defining a target category \( y = c \) is:
\[
P(y = c | x) = \text{soft} \max (w_{t}^T x) = \frac{w_{c}^T x}{\exp \left( \sum_{c'} w_{c'}^T x \right)}
\] (9)

where the sample is \( (x, y) \) and the output target is of the form \( y = \{1, \ldots, C\} \).

### 3.7 Feedforward neural network

Neurons can belong to one of multiple layers. Each layer of neurons can receive signals from the previous layer of neurons and produce a signal output to the next layer. The first layer is the input layer, the middle layer is the hidden layer, and the last layer is the output layer, so that:
\[
a^{(l)} = f_{l}(z^{(l)})
\]
\[
z^{(l)} = W^{(l)} \cdot a^{(l-1)} + b^{(l)} = W^{(l)} \cdot f_{l}(z^{(l-1)}) + b^{(l)}
\] (10)

where: \( f_{l}(\cdot) \) represents a layer neuron activation function; \( W^{(l)} \in \mathbb{R}^{d_{l} \times d_{l-1}} \) is the weight of layer \( l-1 \) to the layer \( l \), the weight matrix \( b^{(l)} \in \mathbb{R}^{d_{l}} \) indicates an offset of the layer \( l-1 \) to the layer \( l \); \( z^{(l)} \in \mathbb{R}^{d_{l}} \) shows a state of neurons in the layer \( l \); and \( a^{(l)} \in \mathbb{R}^{d_{l}} \) represents neurons activity values in the layer \( l \).

### 3.8 Back propagation algorithm

Given a set of samples \( (x^{(i)}, y^{(i)}), 1 \leq i \leq N \), the output of the feedforward neural network is \( f(x | w, b) \), and the objective function is:
\[
J(W, b) = \sum_{i=1}^{N} L(y^{(i)}, f(x^{(i)} | W, b)) + \frac{1}{2} \lambda ||W||_F^2
\] (11)
where $W$ and $b$ contain the weight matrix and the offset vector for each layer, and $\|W\|_F^2 = \sum_{l=1}^{L} \sum_{i=1}^{n_l} \sum_{j=1}^{n_{l-1}} W_{ij}^{(l)}$. Use the gradient drop method to update the parameters:

$$W^{(l)} = W^{(l)} - \alpha \frac{\partial J(W, b)}{\partial W^{(l)}} = W^{(l)} - \alpha \sum_{i=1}^{N_l} \left( \frac{\partial J(W, b; x^{(l)}, y^{(l)})}{\partial W^{(l)}} \right) - \lambda W$$

$$b^{(l)} = b^{(l)} - \alpha \frac{\partial J(W, b)}{\partial b^{(l)}} = b^{(l)} - \alpha \sum_{i=1}^{N_l} \left( \frac{\partial J(W, b; x^{(l)}, y^{(l)})}{\partial b^{(l)}} \right)$$

(12)

while $\alpha$ is the rate of parameter update.

### 3.9 Convolution neural network (CNN)

CNN can extract increasingly complex features in hierarchical structures. Given an image $x_{ij}, 1 \leq i \leq M, 1 \leq j \leq N$ and a filter $f_{ij}, 1 \leq i \leq m, 1 \leq j \leq n$, the convolution output is: $y_{ij} = \sum_{u=1}^{m} \sum_{v=1}^{n} f_{uv} \cdot x_{i-u+1,j-v+1}$. Each neuron of the layer $l$ is connected only to the neurons in a local window of the layer $l-1$, forming a local connection network. The input of the i-th neuron of the layer $l$ is defined as:

$$a_{i}^{(l)} = f \left( \sum_{j=1}^{m} w_{ij}^{(l)} \cdot a_{i+j+m}^{(l-1)} + b^{(l)} \right)$$

(13)

The above formula can also be written as $a^{(l)} = f(w^{(l)} \otimes a^{(l-1)} + b^{(l)})$, where $\otimes$ represents the convolution operation. $w^{(l)}$ are the same for all neurons in layer $l$, that is, it undertakes a weight-sharing process. The image processing is entered in the form of a matrix, which is a two-dimensional convolution. The filter utilized is two-dimensional. In order to enhance the representation capability of the convolution layer, $k$ groups of outputs can be obtained using $k$ different filters. Each set of outputs shares a filter, with the filter being considered a feature extractor. Each set of outputs in a convolutional neural network is also called a set of feature mappings. Thus, the k-th feature of the layer $l$ is mapped to:

$$X^{(l,k)} = f \left( \sum_{p=1}^{m} (W^{(l,k,p)} \otimes X^{(l-1,p)}) + b^{(l,k)} \right)$$

(14)

where the filter $W^{(l,k,p)}$ representing the k-th set of feature maps from the p-th set of feature vectors to the layer $l-1$. Each group of feature maps of the first layer requires $n_{l-1}$ filters with an offset $b$.

Figure 1. The abscissa of the graph is the training step of the network model, and the ordinate is the accuracy. There are two sets of data sets: the test set and the training set. It can be seen from the figure that the accuracy of the training set increases as the number of training steps increases; however, the test set is stopped at an early stage due to the small amount of data, and then the training is over, the accuracy of training is reduced.

### 3.10 Early-stop

In the process of gradient-descent training, the parameters converged on the training sample are not necessarily optimal on the test set due to overfitting. We use the validation set to test the parameters of each iteration. It is then observed whether the results are optimal. If the error rate of the verifier does not decline, then iteration is halted; this strategy is termed ‘early-stop’. You can see Figure 1 for details.

### 3.11 Activation function

Within the neural network architecture, the activation function of the neuron node defines the mapping of the neuron output (for example, in the fully connected network, the output is equal to the inner product of the input vector and the weight vector plus the offset term). The main duty of the activation function is to provide nonlinear modeling capabilities to the network. It is generally a nonlinear function, because if the neural network can only express linear mapping, even if the depth of the network is increased, it still results in a linear mapping, and thus presents a difficulty to effectively model the actual distribution of nonlinear data. When the nonlinear activation function is joined, the deep neural network has a hierarchical nonlinear mapping learning ability. Hence, the activation function is an indispensable part of the deep neural network. The activation functions chosen are mainly ELU, ReLU, RMSprop, and Sigmoid. While the ELU fuses the sigmoid and ReLU, the right linear portion enables the ELU to mitigate the disappearance of the gradient, while the left side of the soft saturation can make the ELU more robust to the input or noise.
The ELU output remains close to zero, so that the convergence rate is faster. ReLU can effectively alleviate the problem of gradient disappearance as compared with the employment of the traditional sigmoid activation function, so as to train the deep neural network directly and in a supervised manner, without relying on unsupervised layer pre-training processes. ReLU is hard-saturated at \( x < 0 \), and when \( x > 0 \), the value of the derivative equals 1.0. Therefore, ReLU can maintain the gradient without decreasing at values of \( x > 0 \), thus alleviating the problem of gradient disappearance. However, upon further training, some inputs can be categorized as being a part of the hard saturation area, resulting in corresponding weights that cannot be updated. ReLU also has an offset phenomenon, that is, the mean value of the output is greater than zero, which will affect the convergence of the network. You can see Figure 2 for details.

3.12 Gradient disappearance

When the error propagates backwards from the output layer, the reciprocal of the activation function of the layer is multiplied at each layer. When we use the sigmoid function as the activation function, the error is continuously attenuated by each layer. When the network layer is very deep, the gradient will continue to decay, or even disappear, making the entire network difficult to train. This is a situation in which the gradient can disappear. One way to solve it is to use a linear activation function, such as a rectifier function, so that the activation function derivative has a value of unity or 1.0. The error problem can then be bypassed, and the training speed is improved.

4. Experimental setup

4.1 Preparation of data sets

The data set is provided by 2017 ISBI (International Skin Imaging Collaboration) Challenge on Skin Lesion Analysis Towards Melanoma Detection. The International Skin Imaging Collaboration (ISIC) is an international effort to improve melanoma diagnosis, sponsored by the International Society for Digital Imaging of the Skin (ISDIS). The ISIC Archive contains the largest publicly available collection of quality-controlled dermoscopic images of skin lesions. Dermatoscopy can eliminate skin surface reflection, enhance deep skin visualization, and improve diagnostic accuracy. The skin mirror dataset that was utilized for our study includes 2000 training data images and 600 test data-set images. Most images have relevant clinical metadata that have been reviewed and recognized by melanoma experts, and some of the images that have been approved are annotated and labeled (including skin features, etc.) by skin cancer specialists. The image size is 1024 \( \times \) 768 pixels. By increasing the size of the training set data, we can reduce the problem of overfitting, because the patch type is obtained via the central voxel. Through a rotation operation [19], data expansion is achieved; therefore, during the training process, the original patch rotation is utilized to generate new patches, thereby increasing our data set.

4.2 Implementation of CNN hardware and software configuration

All melanoma segmentation experiments were done in HP workstations with 16GB RAM, 1TB + 256GB hard drives, NVIDIA Geforce GTX 1070 8 G graphics processors, Intel Core i7-6700HQ processors, running the Ubuntu 16.04 operating system. The implementation of the software was done using Pycharm (Python 2.7), tensorflow + keras with gpu_device, cuda8.0.

4.3 Network structure and parameter settings

Our network architecture used a 3 \( \times \) 3 filter in the convolution layer and the ELU activation function. Although the number of connections can be significantly reduced, the number of neurons mapped to
each feature is not significantly reduced. The input dimension of the connected classifier is still high, and is prone to overfitting. Adding a pool layer (i.e., a subsampling layer) beneath the convolution layer can reduce the feature dimension, and assists in the avoidance of overfitting. The maxpool layer uses a $3 \times 3$ filter, the dropout layer parameter is set to 0.5, and the upsampling layer uses a $2 \times 2$ filter. Convolution layer 4 is combined with the convolution layer 6, while the convolution layer 3 is combined with the convolution layer 7, and convolution layer 2 is combined with the convolution layer 8, followed by convolution layer 1 being combined with convolution layer 9. Finally, the structure of the network is flattened and connected to the convolution output layer, which is followed by a sigmoid function, in order to perform the prediction of the segmentation tag. The convolution neural network has a total of 216 layers and 1,865,676 parameters. You can see Figure 3 for details.

The main building block used to construct the CNN architecture is the convolution layer; it can stack multiple layers above each other to form the feature hierarchy. Each layer can be understood as working by extracting the features of its previous layer and then using this as input to the hierarchical structure to which it is connected. This is done by passing the output profile of the convolution layer as an input channel to the subsequent convolutional processing. From the perspective of the neural network, the feature map corresponds to the hidden unit of the neuronal layer. Specifically, each coordinate in the feature map corresponds to a single neuron whose receiving field corresponds to the size of the filter. The value of the filter is also the weight of the connection between the neurons of the current layer and the neurons on the upper layer. Each feature map can be considered a topological permutation diagram: the filter responds to each spatial neighborhood of the input plane via a sliding window, by using a nonlinear feature extractor of the same specific local space (whose parameters are learned). Practice shows that the filter is similar to an edge detector, and each filter is adjusted to a different spatial frequency, scale and orientation, and is suitable for training data.

Our neural network model is affected by the number of melanoma images in the training data set. To increase the melanoma data and to generate more data, we randomly moved and rotated the existing melanoma medical image training set. This data conversion can also be performed extended to other models. In the training process, the choice of the weight value of our neural network model was done by initializing or transferring the learning approach (using the weight of the existing model). The learning rate of the network model decreased gradually with the increase of epoch. The purpose of the learning rate adjustment was to adjust the update the weight at the end of each batch. The momentum factor was used to control the effect of the last weight update on the current weight update, and was set to a value of 0.9.

Our algorithm for optimization loss of network models mainly uses RMSprop and Adam; RMSprop can be counted as a special case of Adadelta. RMSprop relies on the global learning rate, which is suitable for dealing with non-stationary targets. For the Adam algorithm, after bias correction, the iterative learning rate has a definite range, making the parameters more stable.

Batch size was made dependent upon the size of the memory. The size of our patch was assigned a value of 64, with the step size set to a value of 100. The KerasClassifier builder was used to obtain the default parameters (such as epoch number and batch
size), and to pass them to the ‘fit’ function. Our network architecture used the batch normalization layer several times during the convolution process. This improves learning efficiency and gradient flow, and it reduces initializer weights dependency instead of the use of dropout. Batch normalization essentially solves the gradient problem in the reverse propagation process; the gradient that passes through the layer is multiplied by the parameters of that layer. Since the network layer is deep, if the weight of the reverse propagation gradient is less than 1, then the gradient becomes very small, such as $0.9 \times 100$; and if most of the weights are greater than 1, then when the gradient arrives at this level, there will be a gradient explosion, such as $1.1 \times 100$. Batch normalization can make the back propagation value no longer relevant to the scale of the weights. That is, although we change the weight in the process of updating, the back-

Table 1. The table is the architecture of the improved convolution neural network, including the names of the layers, the functions used and the corresponding parameters.

| Name            | function | variant | Name            | function | variant |
|-----------------|----------|---------|-----------------|----------|---------|
| Convolution1    | Inception| 32      | Convolution6    | Inception| 256     |
| Pool1           | Nconvolution | 32$^3$3$^3$ | Convolution6    | Dropout | 0.5     |
| Pool1           | Dropout  | 0.5     | After_conv3     | Rblock   | 1,128   |
| Convolution2    | Inception| 64      | Up7             | Merge    | 2$^*2$ |
| Pool2           | Nconvolution | 64$^3$3$^3$ | Convolution7    | Inception| 128     |
| Pool2           | Dropout  | 0.5     | Convolution7    | Dropout  | 0.5     |
| Convolution3    | Inception| 128     | After_conv2     | Rblock   | 1,64    |
| Pool3           | Nconvolution | 128$^3$3$^3$ | Up8           | Merge    | 2$^*2$ |
| Pool3           | Dropout  | 0.5     | Convolution8    | dropout  | 0.5     |
| Convolution4    | Inception| 256     | After_conv1     | Rblock   | 1,32    |
| Pool4           | Nconvolution | 256$^3$3$^3$ | Up9             | Merge    | 2$^*2$ |
| Pool4           | Dropout  | 0.5     | Convolution9    | Inception| 32      |
| Convolution5    | Inception| 512     | Convolution9    | Dropout  | 0.5     |
| Convolution5    | Dropout  | 0.5     | Convolution10   | Convolution | 1$^*1$1 |
| After_conv4     | Rblock   | 1,256   | output          | Flatten, Densen | sigmoid |
| Up6             | Merge    | 2$^*2$  |                 |          |         |

Table 2. The table is the structure of the function Inception and Nconvolution.

| Inception:     | Convolution | ELU | Convolution | BN | ELU | MaxPooling | Convolution | Merge |
|----------------|--------------|-----|-------------|----|-----|------------|-------------|-------|
| Nconvolution:  | Convolution  | BN  | ELU         |    |     |            |             |       |

Figure 4. The ‘conv3$^3$3’ in this diagram represents the convolution calculation using the 3$^3$3 filter, and the ‘up’ indicates that the upper layer uses a 2$^*2$ filter and the ‘drop’ represents the dropout layer.
propagation gradient is not affected, so that the batch normalization solves the gradient problem in the reverse propagation process (e.g., the gradient disappears and explodes), and can enable the overall update of the weights of the different scales to be more consistent. The following Tables 1 and 2 and Figure 4 shows the network architecture of the improved neural network model that we used.

5. Experimental results and experimental evaluation

5.1 Qualitative results

Melanoma image segmentation is often used for clinical analysis in radiotherapy, and we should therefore ensure that the image segmentation profile provides the correct response for determining the shape and fidelity of the lesion. Any excessive containment can lead to a healthy substrate being irradiated, while any unqualified detection will cause the melanoma tumor area to remain untreated.

When the image segmentation process has a strong local deviation, it does not necessarily occupy a large volume, but will lead to larger shape differences. The segmentation of the melanoma image using the neural network automatic segmentation algorithm was compared in this study to the division of the image by an expert clinician. If the automatic segmentation algorithm is similar to that produced by the expert, then it is considered to be an acceptable alternative method for achieving expert-level results. A good automated medical image segmentation algorithm also requires less time and has better accuracy than

![Figure 5](image-url)

Figure 5. The figure is the result of the melanoma test set segmentation. The first picture is segmented by a clear red line, and the second is the binarized mask of the first graph.

![Figure 6](image-url)

Figure 6. The graph contains 20 melanoma image segmentation results. The first and fourth rows are the result of the segmentation of the 20 melanoma images using the improved CNN, the lesion area is circled with the red line; the second and fifth rows are the mask of the test image segmentation; the third and sixth rows are Experts’ manual segmentation, ie ground truth mask. The results of the test segmentation are compared with the ground truth.
the interactive expert segmentation. You can see Figure 5 for details.

5.2 Quantitative results

In this work, overlap was calculated by determining a set of statistics to measure the proximity of the melanoma test image as compared with its true value. One image is the melanoma “test” image (the mask of the test image obtained by training using a neural network), and the other is the melanoma “ground truth” (the binary image manually divided by the expert). Both test and real images are binarized masked images – the foreground color is white and the background is black. When the test image perfectly matches the ground truth value, the image is

Figure 7. The figure is the indicator of the melanoma test segmentation (FP, TN, TP, FN), by comparing the test segmentation results with the expert segmentation results.
determined to have the most complete overlap. In the test image, any foreground pixel that overlaps the foreground of the ground truth will be considered “true” (TP), because they are correctly marked as foreground. Background pixels that overlap the background of the real image of the ground are considered “false negatives” (FN) because they are correctly marked as background. Foreground pixels in the test image that overlap the background in the ground truth image are considered “false positive” (FP) because they should be marked as part of the background but are not. The foreground pixel in the ground truth that overlaps the background pixel in the test image will be treated as “false negative” (FN) (as it is marked as part of the background but should not be). We selected seven mask scenes from the results of the melanoma image test set and overlapped the scene with the ground truth mask image. The experimental results show that the melanoma segmentation results trained with our network structure were more accurate as compared with existing technologies reported in the literature. You can see Figures 6 and 7 for details.

In order to obtain the index value of each automatically segmented image, we randomly selected the partial image segmentation results from the melanoma image test set. The table shows that the sensitivity, specificity and accuracy of the segmentation results of our randomly selected seven images and the expert hand-split ground truth.

The indicators involved are as follows: True Positive Rate (TPR) = Sensitivity; TP/(TP + FN). False Positive Rate (FPR): FP/(FP + TN). True Negative Rate (TNR) = Specificity: TN/(FP + TN). False Negative Rate (FNR): FN/(TP + FN). Precision: TP/(TP + FP). Recall: TP/(TP + FN). F-factor: 2TP/(2TP + FP + FN). Rand Index: measures the similarity between two data clusters. The value of RI is [0,1]. The larger the value, the greater the clustering that occurs. The exact clustering returns a minimum score of 0, and perfect clustering returns a maximum score, i.e., a value of unity or 1.0. (TP + TN)/(TP + FP + FN + TN). Adjusted Rand Index: In order to achieve that “in the case of clustering results randomly generated, the indicators should be close to zero”, the adjusted rand index (ARI) was raised, so that it has a higher degree of distinction. It is a variant of the Rand index, taking into account that incorporation of randomness will lead some objects to occupy the same cluster, so that the Rand index will never reach a level of zero. ARI has a characteristic range of [-1,1]. Larger values signify that the clustering result is in harmony with the optimal solution. From a broad perspective, ARI measures the degree of coincidence of the two data distributions:

$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]}$$  \hspace{1cm} (15)

From the following table, we can see seven melanoma segmented images with a sensitivity at (0.886,1), specificity at (0.97, 1), F-factor at (0.93, 0.98), precision at (0.88, 1); the result of the segmentation has a higher accuracy. You can see Tables 3 and 4 for details.

6. Conclusion
Automatic division of the skin melanoma is a very challenging task because the lesion size, shape, intensity and position vary greatly. We use an improved CNN neural network structure, with the original pixels of the image formed as input, in a hierarchical way to learn a set of nonlinear transformations which represent the contents of the image. The model consists of a set of local filters and has a convolution layer, linear unit activation function, maximization layer, dropout layer, batch normalization layer, merge layer, flatten

| Table 3. The table is the indicators of the melanoma image test set segmentation results TPR, FPR, TNR, FNR, Precision, Recall, F-factor, RandIndex, ARI, where the seven image indicators correspond to the above seven mask images. |
|-----------------|----|----|----|----|----|----|
| ImageNo | Value Measurement | 1  | 2  | 3  | 4  | 5  | 6  | 7  |
| True Positive Rate (TPR) = Sensitivity | 0.886688 | 0.911996 | 0.94458 | 0.962661 | 0.965972 | 0.9659725 | 0.998674 |
| False Positive Rate (FPR) = FP/(FP + TN) | 0.000033 | 0 | 0.0010928 | 0.001526 | 0.005927 | 0.0059273 | 0.024422 |
| True Negative Rate (TNR) = Specificity | 0.999067 | 1 | 0.99889 | 0.994747 | 0.9949073 | 0.994073 | 0.975578 |
| False Negative Rate (FNR) = FN/(TP + FN) | 0.113312 | 0.088004 | 0.055420299 | 0.037039 | 0.034028 | 0.034028037 | 0.001326 |
| Precision = TP/(TP + FP) | 0.978084 | 1 | 0.997704963 | 0.989365 | 0.983625 | 0.983624606 | 0.883393 |
| Recall = TP/(TP + FN) | 0.886688 | 0.911996 | 0.944579701 | 0.962961 | 0.965972 | 0.965972 | 0.998674 |
| F-factor = 2 x (precision x recall)/(precision + recall) | 0.930146 | 0.953973 | 0.970415793 | 0.975985 | 0.974718327 | 0.937503 |
| RandIndex (RI) | 0.981212 | 0.952611 | 0.94402755 | 0.987895 | 0.973374 | 0.973375 | 0.983518 |
| AdjustedRandIndex (ARI) | 0.921 | 0.900282 | 0.965969847 | 0.964892 | 0.944108 | 0.944108 | 0.898418 |

Table 4. Lesion Segmentation Result. Higher results is better.
| TPR | TNR | AC = (TP + TN)/(TP + FP + FN + TN) |
|-----|-----|-------------------------------------|
| Our | 0.95 | 0.94 | 0.91 |
| SegNet [20] | 0.87 | 0.96 | 0.98 |
| FCN [21] | 0.85 | 0.70 | 0.82 |
layer, and sigmoid layer, in order to achieve satisfactory image feature learning and test segmentation results. We used the evaluation plan from the melanoma test image to evaluate our program, compared with ground truth, and calculated index parameters. It was shown that the method achieved a higher accuracy than existing architectures in most cases. In a few instances, the melanoma test results lagged due to the lack of image intensity, uniformity, and presence of imaging artifacts, which resulted in a few false positive results. However overall, our approach can incorporate a smaller melanoma training set, with shorter training and reasoning time, and still achieve an excellent segmentation result. Our findings suggest therefore, that this is a more advanced image segmentation method as compared with currently-used techniques, with the automatic segmentation method having the potential to also replace subjective, time-consuming, and expensive manual segmentation paradigms.

**Disclosure statement**

No potential conflict of interest was reported by the author(s).

**References**

[1] Australian CC. Cancer Council to launch new research/failure to monitor highlights cancer risk, http://www.cancer.org.au/cancersmartlifestyle/SunSmart/Skincancer-factsandfigures.htm, 2010.

[2] Green A, Martin N. etc., Computer image analysis in the diagnosis of melanoma. J Am Acad Dermatol. 1994;31:958–964.

[3] Lee HC. Skin cancer diagnosis using hierarchical neural networks and fuzzy logic. Department of Computer Science, University of Missouri, Rolla, 1994.

[4] Aitken JF, Pfitzner J, Battistutta D, et al. Reliability of computer image analysis of pigmented skin lesions of Australian adolescents. J. Cancer. 1996;78:252–257.

[5] Chang Y, Stanley RJ, Moss RH, et al. A systematic heuristic approach for feature selection for melanoma discrimination using clinical images. Skin Res Technol. 2005;11:165–178.

[6] She Z, Liu Y, Damatoa A. Combination of features from skin pattern and ABCD analysis for lesion classification. Skin Res Technol. 2007;13:25–33.

[7] Fassihi N, Shanbehzadeh J, Sarrafzadeh A, et al. Melanoma diagnosis by the use of wavelet analysis based on morphological operators. Proceedings of the International Multiconference of Engineers and Computer Scientists. 16–18, 2011.

[8] LeCun B, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition. Proc IEEE. 1998;86:2278–2324.

[9] Havaei M, Davy A, Warde-Farley D, et al. Brain tumor segmentation with deep neural networks, Proc. BRATS-MICCAI, 2014.

[10] Zikic I, Ioannou Y, Criminisi A, et al. Segmentation of brain tumor tissues with convolutional neural networks, Proc. BRATS-MICCAI, 2014.

[11] Urban B, Bendzus M, Hamprecht FA, et al. Multi-modal brain tumor segmentation using deep convolutional neural networks, Proc. BRATS-MICCAI, 2014.

[12] Pinheiro P, Collobert R. Recurrent convolutional neural networks for scene labeling, Proceedings of the 31st International Conference on Machine Learning, pp. 82–90, 2014.

[13] Farabet C, Camille Couprie C, Najman L, et al. Learning hierarchical features for scene labeling. IEEE Trans Pattern Anal Mach Intell. 2013;35:1915–1929.

[14] Brosch T, Yoo Y, Tang L, et al. Deep convolutional encoder networks for multiple sclerosis lesion segmentation,” International Conference on Medical Image Computing and Computer-Assisted Intervention – MICCAI; 2015. Vol. 9351, p. 3–11.

[15] Kang K, Wang X. Fully convolutional neural networks for crowd segmentation, [Online]. Available: arXiv:1411.4464, to be published, 2014.

[16] Long ES, Shelhamer E, Darrell T. “Fully convolutional networks for semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 3431–3440, 2015.

[17] Ronneberger O, Fischer P, Brox T. “U-net: Convolutional networks for biomedical image segmentation,” in Proc. 18th Int. Conf. MICCAI, p. 8, 2015.

[18] Huang J, Jain V. Deep and wide multiscale recursive networks for robust image labeling arXiv preprint arXiv:1310.0354, 2013.

[19] Amaral T, Silva LM, Alexandre LA, et al. Transfer learning using rotated image data to improve deep neural network performance, International Conference Image Analysis and Recognition, ICIAR: Image Analysis and Recognition pp. 290–300, 2014.

[20] Badrinarayanan V, Handa A, Cipolla R. A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling, arXiv preprint arXiv:1505.07293, 2015.

[21] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation, IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440, 2015.