9 Artificial Intelligence and Deep Learning in Malaria Diagnosis

Min Fu and Zuodong Li

Abstract

Microscopic examination is the gold standard for malaria diagnosis. However, microscopic diagnosis of malaria relies heavily on pathologists or technicians with specialized knowledge and expertise, and manual microscopic screening is an unreliable method in nonexpert situations. In addition, diagnosing malaria requires considerable time and effort, overwhelming experts when too many cases need to be tested, and for the same test, different results can occur due to different testers. The same test can yield different results for different testers. Therefore, automated image analysis systems can break the dependence of accurate *Plasmodium* detection on specialized technicians, improve the accuracy and efficiency of detection and reduce the cost of *Plasmodium* detection. Traditional methods use image-based algorithms for extracting features (e.g., color, shape, texture, and gradient changes) for image detection, image segmentation, feature extraction, and classification. However, the accuracy rate of traditional methods is 80–90%, which cannot meet clinical requirements. With the re-emergence of artificial neural networks, deep learning algorithms based on convolutional neural networks and other deep learning algorithms driven by massive data have gained tremendous development in the direction of computer vision, such as image classification, target localization, target detection, and image segmentation, which has led to an increasing interest in the applicability of deep learning methods based on convolutional neural networks in medical image analysis. After applying the deep learning algorithm, compared with the traditional algorithm, the accuracy can be increased by 10–20%, which has

M. Fu (✉)
School of Aerospace Engineering, Xiamen University, Xiamen, Fujian, China

Z. Li
Wuhan Foreland Intelligent Technology Co., Ltd., Xiamen, Fujian, China

H. Mehlhorn et al. (eds.), *Malaria Control and Elimination in China*, Parasitology Research Monographs 18, https://doi.org/10.1007/978-3-031-32902-9_9

reached the level of human doctors. It has become a reality that the detection of malaria parasites can be replaced by machines in the clinic.

Keywords

Microscopic diagnosis · Malaria diagnosis · Traditional methods · Deep learning · Feature extraction · Artificial intelligence

9.1 The Inevitability of Computer-Aided Diagnosis

Since the latter part of the last century, various attempts and studies have been made on the self-detection and classification of malaria, and remarkable results have been achieved. In the twenty-first century, information technology is becoming increasingly developed, and machine vision, machine learning, and image processing technologies are increasingly widely used in scientific research. These technologies are also widely used in disease detection, and the traditional methods of malaria detection have also changed. The use of computer vision for malaria diagnosis is considered to be a method to simplify the diagnostic process by increasing diagnostic accuracy, saving diagnostic time, reducing the required manpower, and minimizing human error (Tek et al. 2009; Díaz et al. 2009; Mandal et al. 2010; Kumarasamy et al. 2011; Jan et al. 2018).

Microscopic examination is the gold standard for malaria diagnosis, with thin blood smears identifying the species; thick blood smears, which are slide samples with large amounts of blood, are most useful for characterization. Thick blood films are primarily used to detect the presence of malaria parasite infection and to assess malaria parasitemia but not to detect which species the infection belongs to. In contrast, thin blood smears are 1–2 microliters of blood spread over most of the slide, dried and fixed in methanol, and when fixing thin blood smears, care should be taken to avoid exposing thick smears to methanol. Thin blood smears help physicians detect which Plasmodium species are causing the infection. Table 9.1 discusses the

Table 9.1 Variation in blood smears

Thick smears	Thin smears
Suitable for detecting the presence of malaria parasites	Beneficial for *plasmodium* species
The sample requirement is blood for no less than 4–5 microliters	No more than 1–2 microliters of blood spread over a larger area of the slide
RBCs must be ruptured but WBCs, platelets, and malaria parasites must be visible before or during the staining	Observation of *plasmodium* in RBCs and analysis of the differences between infected and normal RBCs
Improving the detection efficiency of malaria parasites	Low detection efficiency and easy to miss at low density
No methanol fixation required	Requires methanol fixation prior to staining
For confirming the diagnosis	Able to identify *plasmodium* species

differences between thick and thin blood smears in the evaluation of malaria parasite detection. The advantages of microscopic examination are the ability to differentiate between malaria parasite species, quantify parasitemia, observe the different stages of parasite characteristics, and have a low material cost. However, microscopic diagnosis of malaria relies heavily on pathologists or technicians with specialized knowledge and expertise, and manual microscopic screening is an unreliable method in nonexpert situations. In addition, diagnosing malaria requires considerable time and effort, overwhelming experts when too many cases need to be tested, and for the same test, different results can occur due to different testers. The same test can yield different results for different testers. Therefore, automated image analysis systems can break the dependence of accurate Plasmodium detection on specialized technicians, improve the accuracy and efficiency of detection and reduce the cost of *Plasmodium* detection. Computers are playing an increasingly important role in the medical field, and without computer technology, medical proficiency and productivity would be significantly reduced. Computers already play an important role in various medical diagnostic applications, such as digital X-ray, magnetic resonance imaging (MRI), computed tomography (CT), and ultrasound. Computer-aided diagnosis of malaria is a microscopic diagnostic technique through the use of computer vision and machine learning algorithms. It can replace manual microscopic detection for semiautomatic or fully automatic detection of *Plasmodium*. With the development of picture technology and vision technology, malaria detection has gradually evolved from purely manual to semiautomated or even fully automated. Screening for malaria parasites by light microscopy is still considered the main method of malaria detection in health testing laboratories and hospitals worldwide, and incorrect diagnostic cutoffs will lead to patients infected with malaria not receiving timely and effective treatment, resulting in serious consequences. In recent years, malaria self-testing systems have been shown to be systematic, and the steps are more uniform. The main steps to acquire images are image preprocessing, image segmentation, feature extraction, and detection and classification.

9.2 Introduction of Traditional Methods to Assist in the Intelligent Diagnosis of Malaria

Traditional methods use image-based algorithms for extracting features (e.g., color, shape, texture, and gradient changes) for image detection, image segmentation, feature extraction, and classification. Makkapati and Rao (2009) proposed a method to segment red blood cells and stained masses from Leishmania-stained blood smears by detecting the primary color range and calculating the optimal saturation threshold to segment erythrocytes and parasites. The method is performed in HSV space, where the background in the image is first determined by detecting the primary color range and the remaining pixels are the erythrocytes and chromatin dots to be segmented. Then, a method is proposed to determine the optimal saturation threshold to segment the erythrocytes and chromatin dots. This work illustrates the potential of color image processing techniques in providing diagnostic solutions for severe infectious diseases. Ross et al. (2006) proposed an automated image processing and classification technique to detect red blood cells infected with malaria parasites and to differentiate the species of infected Plasmodium. The authors used morphological techniques combined with a new threshold selection technique with local and global thresholds to segment potentially infected red blood cells from thin blood smears and then designed tree classifiers with two nodes based on thin blood smear image features such as texture, color, and geometry of the image, using feedback feedforward neural networks to identify whether a red blood cell is infected and, if infected, to determine the malaria species, eliminating the reliance on technician skills and experience. Panchbhai et al. (2012) proposed a model based on RGB color space to achieve segmentation of erythrocytes and parasites by first dividing the thin blood smear RGB image into three different layers, R, G, and B. Then, the G layer was further processed, and the Otsu algorithm was used to determine the optimal threshold value to segment all infected erythrocytes, reducing the detection time and the chance of human error in malaria. May and Aziz (2013) proposed a method for the automatic quantification and classification of *P. falciparum*-infected erythrocytes. This method uses the Otsu method to find local thresholds to segment out infected erythrocytes. Since each parasite infects one erythrocyte, the thin blood smear images are classified according to the number of infected cells detected. This method finally obtained a good classification result, but the limitation of this study is that it only targets *Plasmodium interrogans* in the trophozoite stage, so the results are relatively poor for other parasite species. Somasekar and Reddy (2015) proposed an edge-based segmentation method to segment erythrocytes infected with *Plasmodium*. First, the brightness difference of the image was corrected by reducing the effect of image color through color space transformation and γ homogenization, and then the infected erythrocytes were extracted using the fuzzy C-mean clustering method, and the infected erythrocytes were segmented using the minimum perimeter polygon algorithm. This work provides a consistent and robust method for edge segmentation of infected erythrocytes in microscopic images. In 2017, Somasekar and Reddy (2017) also proposed a two-stage threshold segmentation method for segmenting malaria-infected cells in microscopy images. The method performs segmentation in two stages, where the first stage maximizes the interclass variance of the original image and then iterative threshold selection is performed on the thresholded image of the first stage to segment malaria parasites according to suitable stopping conditions. This work was able to successfully detect *Plasmodium* without prior knowledge of the image content, without parameter adjustment, and to extract malaria-infected cells in both thick blood smears and thin blood smears.

9.3 Introduction of Deep Learning in the Assisted Intelligent Diagnosis of Malaria

With the re-emergence of artificial neural networks, deep learning algorithms based on convolutional neural networks and other deep learning algorithms driven by massive data have gained tremendous development in the direction of computer vision, such as image classification, target localization, target detection, and image segmentation (Krizhevsky et al. 2012; He et al. 2016), which has led to an increasing interest in the applicability of deep learning methods based on convolutional neural networks in medical image analysis. Meanwhile, an increasing number of deep learning models have been applied to malaria diagnosis for automatic classification, detection, and segmentation tasks of cells in malaria microscopy images.

The essence of deep learning is to build deep networks with multiple hidden layers based on large-scale sample data sets and powerful computing power and train the networks through multiple iterations to learn the effective features in the data to obtain higher prediction or classification accuracy. Compared with traditional machine learning, the most important feature of deep learning is its network structure with multiple hidden layers, which can automatically learn the features in the data without human involvement.

Deep learning is the latest trend in machine learning, and it has already achieved outstanding performance in many nonmedical fields. In addition to using more network layers, deep learning is seen as an extension of the well-known multilayer neural network classifier trained using backpropagation, using different kinds of layers in a typical succession. Deep learning usually requires a large training set, which is one of the reasons why the medical field has been slower to introduce deep learning methods, as is by the difficulty in harvesting annotated training images in terms of expert knowledge requirements and considering privacy issues.

Hung and Carpenter (2017) applied for the first time a target detection model previously used for natural images on blood smear images of Plasmodium infection to identify *Plasmodium* cells and recognize the growth stage they are in based on Faster R-CNN for malaria detection, which avoids the segmentation task and does not rely on general features for classification. Rajaraman et al. (2018) evaluated the use of convolutional neural network-based pretrained models to extract image features for the classification of infected and uninfected cells. The paper evaluated the performance of pretrained convolutional neural networks, including AlexNet, VGG16, ResNet-50, and DenseNet-121, in extracting features of infected and uninfected cells and experimentally determined the optimal layer for feature extraction for each model to help improve the classification. The results showed that the pretrained convolutional neural network exhibited good performance for extracting image features. Subsequently, Rajaraman et al. (2019) built on the literature to detect infected cells in blood smears by constructing an integration of multiple deep learning models. The performance of custom and pretrained convolutional neural networks was first evaluated, and an optimal integration model was constructed for the challenge of classifying infected and normal cells in thin blood smear images. The integration by VGG-19 and SqueezeNet outperformed the classification performance of a single model.

9.3.1 Introduction of Neural Network

Artificial neural networks (ANNs), abbreviated as neural networks, are mathematical models based on the principle of biological neural networks, which use the network topology principle to simulate the response mechanism of the human brain nervous system to various information by analyzing and studying the structure of the human brain and the feedback mechanism of external stimuli. ANNs have successfully solved many real-world problems in many fields, such as video detection, smart driving, and smart medical care, and have shown good performance. In ANN, a neuron is the smallest unit of operation, which receives input parameters from other neurons and outputs the final result after computation. Figure 9.1 shows an example diagram of the neuron structure.

$x1, x2..., x$n are the inputs of this neuron, $w1, w2..., wn$ are the weights and b is the bias. The output of this neuron can be expressed by the mathematical formula as:

$$y = f\left(\sum_{i=1}^{n} x_i w_i + b\right) \tag{9.1}$$

where $f(x)$ denotes the activation function.

A neural network consists of multiple neurons and neural layers, and Fig. 9.2 shows a basic neural network structure.

For each neuron in the nth layer of the ANN, its input is the output of the neuron in the nth-first layer, and the output of that neuron will continue to be the input of the neuron in the nth + first layer. In an ANN9 containing multiple intermediate hidden layers, the output of the nth layer neuron and the input of the nth + first layer neuron are interconnected by an activation function. Since many real-world problems are very complex nonlinear problems and the activation function can introduce nonlinear properties to form a nonlinear mapping between the input and output of neurons as a way to increase the nonlinearity of the network, enhance the expressiveness of

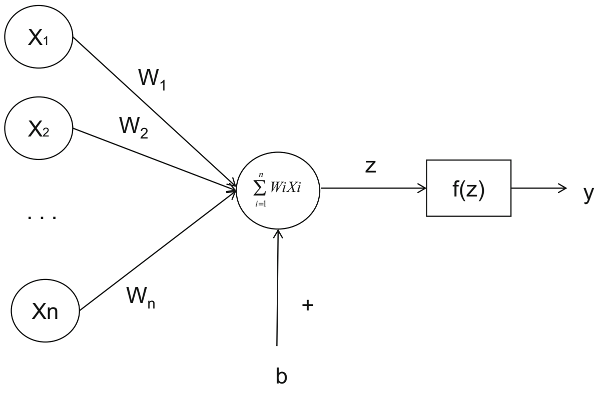


Fig. 9.1 Neuron structure diagram

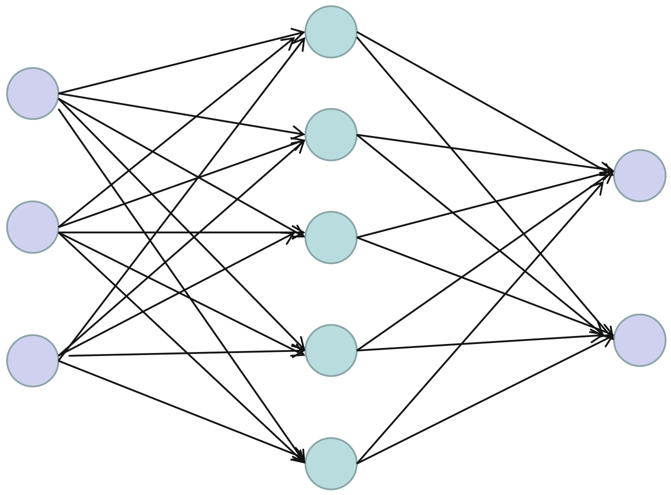


Fig. 9.2 Structure diagram of the Neural Network

the network, and make the network more powerful, the selection of the correct and suitable activation function is crucial to build the ANN model.

The following describes the characteristics and mathematical formulas of several more commonly used activation functions.

9.3.1.1 Activation Function

Sigmoid Function

Sigmoid is the most common nonlinear activation function, and its main role is to map arbitrary real numbers between (0,1), which is usually used for binary classification tasks. Its mathematical formula is:

$$\sigma(x) = \frac{1}{1 + e^x} \tag{9.2}$$

Tanh Function

Tanh is similar to the function curve of a sigmoid. When the input of both functions is small or large, their outputs are close to smooth when the gradient is infinitely close to 0, which will not be conducive to the update of parameters, resulting in the training of the network not proceeding. The output interval of Tanh is (−1,1), and its mathematical formula is expressed as:

$$tanh\ (x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{9.3}$$

ReLU Function

Commonly used in the output of the hidden layer of the network, this function avoids the problem of gradient disappearance that occurs in the sigmoid function and the tanh function, but as the network continues to be trained iteratively, there may be neuron death, which results in parameters that cannot be updated. Its mathematical formula is:

$$f(x) = max\ (0, x) \tag{9.4}$$

LeakyReLU Function

From Eqs. (9.4), it can be seen that when the input of the ReLU function is negative, its function values are all zero, which may lead to neuron death. In the LeakyReLU function, when the input is negative, it sets a nonzero slope to the input so that its function value is no longer 0 as a way to correct the data distribution and avoid the problem of neuron death caused by the ReLU function, whose mathematical formula is:

$$f(x) = \begin{cases} x, & x \geq 0 \\ ax, & x < 0 \end{cases} \tag{9.5}$$

where α is a very small constant, taken between 0 and 1.

9.3.1.2 Backpropagation Algorithm

Currently, the most commonly used and effective network training method in neural networks is the backpropagation algorithm. 10 (Back propagation Algorithm, BP algorithm), whose core technique is the chain derivation rule. The BP algorithm consists of two processes, namely, forward propagation of data and backpropagation of errors.

1. Forward propagation: Mainly used for the transmission of data feature information, the data are first input to the neural network from the input layer and then transmitted toward the output layer by layer. In the neural network, forward propagation can be expressed by the following equation:

$$x^l = f\left(u^l\right) \tag{9.6}$$

$$u^l = w^l x^{l-1} + b^l \tag{9.7}$$

where l is the current layer, x^1 is the output of layer l, b^1 and w^1 denote the bias and weight of the lth layer, respectively, and $f(x)$ is the activation function. Then, the forward propagation of the jth neuron in the lth layer can be expressed as:

$$z_j^l = \sum\nolimits_i a_i^{l-1} * w_{jk}^l + b_j^l \tag{9.8}$$

$$a_j^l = f\left(\sum\nolimits_i a_i^{l-1} * w_{jk}^l + b_j^l\right) \tag{9.9}$$

where z_j^l and a_j^l denote the input and output of the jth neuron in the lth layer, respectively, w_{jk}^l denotes the weight of the kth neuron in the $(l-1)$ layer connected to the jth neuron in the *l*th layer, andb_j^l denotes the bias of the *j*th neuron in the *l*th layer.

2. Backpropagation: Since there is a certain error between the desired result and the actual result, backpropagation passes the error back from the output layer of the network toward the input layer in the opposite direction while updating the values of the parameters in the neural network according to the error value to minimize the error, and backpropagation reflects the correction and fine-tuning of the model based on the error information.

Suppose that the neural network loss function is defined in the form of a squared error function:

$$J(w, b, x, y) = \frac{1}{2} \|h_{W,\ b}(x) - y\|^2 \tag{9.10}$$

where x is the input to the network, y is the corresponding desired output result of input x,$h_{W,\ b}(x)$ is the actual output result of the neural network, and the weights W and bias b are the parameters of the neural network. For a training set containing m data, the overall loss function is defined as:

$$J(W,\ b) = \frac{1}{m}\Sigma_{i=1}^m J\left(w,\ b,\ x^i,\ y^i\right) = \frac{1}{m}\Sigma_{i=1}^m \left(\frac{1}{2}\|h_{w,\ b}(x^i) - y^i\|^2\right) \tag{9.11}$$

The final training objective of a neural network is to minimize the error between the desired and actual output results of the network by continuously updating the optimization parameter vector *W, b*, i.e., minimizing the loss function *J*(*W,b*) of the network. Usually, the gradient descent algorithm is used to update *W* and *b* with the following update formula:

$$W_{ij}^l = W_{ij}^l - \alpha \frac{\partial}{\partial W_{ij}^l} J(W,\ b) \tag{9.12}$$

$$b_i^l = b_i^l - \alpha \frac{\partial}{\partial b_i^l} J(W, b) \tag{9.13}$$

where α is the learning rate and $\frac{\partial}{\partial W_{ij}^l} J(W, b)$ and $\frac{\partial}{\partial b_{ij}^l} J(W, b)$ denote the loss function to the parameters to *W* and *b* bias derivatives, respectively. The steps of the backpropagation algorithm are as follows: first, the error value of the *j*th neuron in the *l*th layer is calculated according to Eqs. (9.9), (9.10), and (9.12) as follows:

$$\begin{aligned}\delta_j^l = \frac{\partial J}{\partial z_j^l} &= \sum\nolimits_k \frac{\partial J}{\partial z_k^{l+1}} \cdot \frac{\partial z_k^{l+1}}{\partial a_j^l} \cdot \frac{\partial a_j^l}{\partial z_j^l} \\ &= \sum\nolimits_k \delta z_k^{l+1} \cdot \frac{\partial \left(w_{jk}^{l+1} a_j^l + b_j^{l+1} \right)}{\partial a_j^l} \cdot f'\left(z_j^l\right) \\ &= \sum\nolimits_k \delta z_k^{l+1} \cdot w_{jk}^{l+1} \cdot f\left(z_j^l\right)\end{aligned} \tag{9.14}$$

The partial derivatives of the weights are then calculated as follows:

$$\frac{\partial J}{\partial w_{jk}^l} = \frac{\partial J}{\partial z_j^l} \cdot \frac{\partial z_j^l}{\partial w_{jk}^l} = \delta_j^l \cdot \frac{\partial \left(w_{jk}^l a_k^{l-1} + b_j^l \right)}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l \tag{9.15}$$

Next, the partial derivatives of the bias are calculated as follows:

$$\frac{\partial J}{\partial b_j^l} = \frac{\partial J}{\partial z_j^l} \cdot \frac{\partial z_j^l}{\partial b_j^l} = \delta_j^l \cdot \frac{\partial \left(w_{jk}^l a_k^{l-1} + b_j^l \right)}{\partial b_j^l} = \delta_j^l \tag{9.16}$$

Finally, the parameter vectors *W* and *b* are updated according to Eqs. (9.13) and (9.14).

9.3.1.3 Neural Network Model Optimization Algorithm

The goal of neural network learning is to find suitable parameter values that make the value of the loss function as small as possible, and optimization algorithms are a class of algorithms that address this problem. The gradient descent algorithm is the most basic and common optimization algorithm. The gradient descent algorithm continuously updates the parameters in the neural network according to certain rules until the value of the loss function no longer changes significantly, i.e., it converges to a minimum value to achieve the training purpose of the neural network. In a function, the direction of the fastest decrease in the value of the function is the negative gradient direction of the function, so the gradient descent algorithm gradually iteratively updates the parameters along the negative gradient direction, which allows the value of the loss function to converge to the minimum value at the fastest speed. Equations (9.13) and (9.14) are the iterative formulas of the gradient descent algorithm. The condition of using the gradient descent algorithm to obtain the global optimal solution is limited to the loss function being a convex function, but in practical applications, the loss function is relatively complex and not always convex, so there will be saddle points and local minima. In addition, the computation time of the gradient descent method is relatively long; in the actual training data, it is usually a lot, and the loss function is the sum of the losses on all the training set data. In addition, the whole training process is quite time consuming. In view of the above problems, the following will introduce several improved optimization algorithms based on gradient descent, which can make the network converge more stably and faster than the gradient descent method and have better optimization effects in applications.

1. Stochastic gradient descent algorithm

Stochastic Gradient Descent (SGD) can speed up the update of parameters in each round. The improvement over the gradient descent algorithm is that the algorithm randomly selects the loss function on training data for parameter optimization in each iteration round instead of directly optimizing the loss function on all 12 training data. For large linear models that need to be trained on large-scale data, SGD is a necessary optimization algorithm, but SGD also has the shortcoming that a small value of the loss function on a certain training data does not mean a small value of the loss function on all training data, which means that SGD only considers the gradient of a single data, which can easily lead to a locally optimal result for the network. To balance the performance of both the above gradient descent algorithm and the stochastic gradient descent algorithm, in practice, each iteration usually selects a small portion of the loss function on the training data for optimization, and this small portion of the training data is treated as a batch, so the method is also known as the small batch gradient descent algorithm.

2. Gradient descent algorithm with Momentum. The principle of the gradient descent algorithm with Momentum is to calculate the exponentially weighted average of the gradients and then use this value to update the parameter optimization network, and the parameter update formula is as follows:

$$v_t = \beta v_{t-1} + (1-\beta) g_t \tag{9.17}$$

$$w_t = w_{t-1} - \alpha v_t \tag{9.18}$$

where α and β are two hyperparameters, α denotes the learning rate, which controls the exponentially weighted average, and β denotes the effect of the previous gradient on the present, the larger β means the greater the effect of the previous gradient on the present.g_t denotes the original gradient of a parameter in a certain iteration, v_t is the gradient calculated by the exponentially weighted average, and the initial value of v_t is 0. The above formula can be interpreted as speeding up the update rate when the gradient direction is constant, which reduces the oscillation and speeds up the convergence of the network.

3. Adaptive learning rate algorithm

If the learning rate is set too small, the convergence speed will be very slow for parameters with large gradients; if the learning rate is set too large, the parameters that will be optimized may be unstable. Adagrad is the most classic adaptive learning rate algorithm, which automatically adjusts the learning rate of each parameter by calculating the sum of squares of historical gradients, and its adjustment law is such that parameters that are updated more frequently have a smaller learning rate and parameters that are updated rarely have a larger learning rate. The updated formula of the Adagrad algorithm is:

$$V_t = V_{t-1} + {g_t}^2 \tag{9.19}$$

$$w_t = w_{t-1} - \frac{\alpha}{\sqrt{v_t + \epsilon}} g_t \tag{9.20}$$

where g_t is synonymous with Eq. (9.18), $V_t$is the cumulative squared gradient sum with an initial value of 0. α is the global learning rate, and ε is a tiny quantity used to prevent the denominator from being zero.

One drawback of the Adagrad algorithm is that as V_t gradually increases cumulatively, the learning rate decreases, eventually leading to the cessation of updates. Therefore, an improved RMSprop algorithm is proposed. The RMSprop algorithm introduces a decay factor β that makes V_t decay at each iterative update, similar to the approach in Momentum. Its updated equation is as follows:

$$V_t = \beta V_{t-1} + (1-\beta){g_t}^2 \tag{9.21}$$

$$w_t = w_{t-1} - \frac{\alpha}{\sqrt{v_t + \epsilon}} g_t \tag{9.22}$$

Adam's algorithm combines the RMSprop and Momentum algorithms, considering both the sum of squares of historical gradients to achieve adaptive adjustment of the learning rate of each parameter and retaining the average of historical gradients as momentum, so that the update of each parameter is more independent during the training process of the network, which speeds up the training speed of the network and improves the stability of the network training at the same time. Its update equation is as follows:

$$v_t = \beta_1 v_{t-1} + (1-\beta_1)\, g_t \tag{9.23}$$

$$V_t = \beta_2 V_{t-1} + (1-\beta_2){g_t}^2 \tag{9.24}$$

$$w_t = w_{t-1} - \frac{\alpha}{\sqrt{V_t + \epsilon}} v_t w_t = w_{t-1} - \frac{\alpha}{\sqrt{V_t + \epsilon}} v_t$$

$$w_t = w_{t-1} - \frac{\alpha}{\sqrt{V_t + \epsilon}} v_t \tag{9.25}$$

9.3.1.4 Convolutional Neural Networks

Convolutional neural networks (CNNs) are one of the most classical and commonly used neural networks, and they have made many breakthroughs in image analysis and processing. Since convolutional neural networks were proposed, many models based on convolutional neural networks have also been successively proposed for solving image-related problems, such as image denoising, action recognition, and target detection, and all of these models have shown good performance. Compared with previous image processing algorithms, the main advantage of CNN is that it reduces the complicated preprocessing steps, especially the image preprocessing that requires human involvement, and CNN can directly input the image to be processed into the network for subsequent work, which has been widely used in various fields of image-related applications. In 1959, Hubel and Wiesel, in their study of how the cat brain processes visual information, discovered a complex structure of cells that responded to localized areas of visual information, thus introducing the "receptive field." Inspired by this research and based on it, Fukushima proposed the neurocognitive machine, which is called the predecessor of CNN, which uses an alternating structure of local feature extraction and feature exchange layers to maintain the ability to recognize targets even when they are deformed or displaced. Although the neurocognitive machine model did not use error backpropagation for supervised learning, it is still considered the first discovery of convolutional neural networks. Later, based on the neurocognitive machine, YannLeCun et al. proposed the convolutional neural network model, the famous LeNet-5 network, using the error backpropagation method, and numerous CNN models proposed since then have been improved on this basis. 14 Compared with the ordinary ANN, the hidden layer of CNN usually consists of a convolutional layer, an activation layer, a pooling layer, and a fully connected layer. From input to output, the layers of CNN create connections between each other through different neuron nodes and transfer the input data along the network structure layer by layer; successive convolutional and pooling layers are used to extract the features of the input data and compress the features, and the final fully connected layer will classify the data based on the extracted features.

1. Convolutional layer
   The convolutional layer is mainly used for extracting image features. Two techniques are applied in the convolutional operation, namely, local perceptual field and weight sharing, which allow the network to better utilize the local features of the image and significantly reduce the number of parameters to be optimized in the network.
2. Local perceptual field: In convolutional neural networks, the neuron nodes in each layer are no longer connected to each other in a fully connected manner, but the neurons in the nth layer are only connected to some neurons in the n-1th layer,

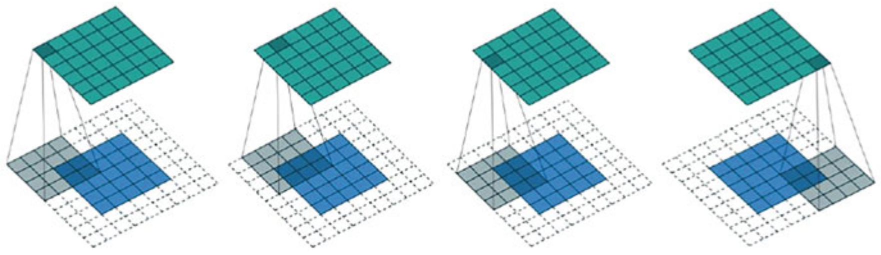


Fig. 9.3 Example of the convolution operation process

i.e., locally connected, thus reducing the number of parameters in the network training. Fig. 2.4 shows an example of the local connection of neurons, in which each neuron in layer $n + 1$ is connected to only 3 neurons in layer n instead of all neurons in layer n. This connection method reduces the number of parameters for network training from $5 \times 5 = 15$ to $3 \times 3 = 9$, which is a 40% reduction in the number of parameters, and the same connection method is used for neurons in layers $n + 2$ and $n + 1$. The use of local connections in CNN greatly reduces the number of parameters, speeds up the learning rate of the network, and reduces the possibility of overfitting to a certain extent.

3. Weight sharing: During the convolutional operation, each convolutional kernel is repeatedly applied to the whole perceptual field, i.e., weight sharing, which can also reduce the number of parameters. As shown in Fig. 9.3, there are three different sets of weights 1, 2, and 3. If only local connectivity is applied, the network needs a total of 9 parameters.

4. Convolution operation

   The convolution operation is the core operation of the convolution layer. Figure 9.3 shows part of the process of a convolution operation. There are several important parameters in the convolution operation process.

5. Pooling layer.

   The pooling layer is usually found between successive convolutional layers and is mainly used to compress the feature map to extract important features of the image on the one hand and reduce the size of the feature map on the other hand, thus simplifying the complexity of computation. The pooling operation can ignore the influence of the target due to rotation, tilt, and other relative position changes, which can reduce the dimensionality of the feature map while improving the accuracy of the feature mapping and, to a certain extent, can also avoid overfitting.

   The pooling operation is usually calculated in two ways: maximum value pooling and mean value pooling. Figure 9.4 shows two examples of the pooling operation process. The basic idea of maximum value pooling is to take the maximum value in each subregion of the feature map as the mapping result; only one of the weights of the convolution kernel is 1, and the rest are 0. The position

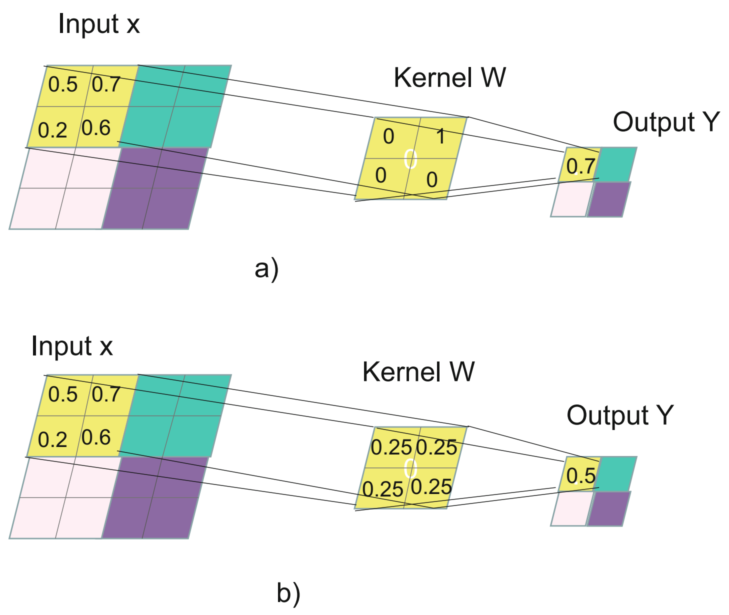


Fig. 9.4 Example of the pooling operation process. (**a**) Maximum pooling. (**b**) Mean pooling

with the weight of 1 in the convolution kernel corresponds to the position with the largest value in the overlap region between the input image and the convolution kernel. In the above figure, the size of the convolution kernel is 2×2, and its moving step on the input image Input X is 2. The result of maximum pooling is to reduce the input image to 1/4 of the original size and to obtain the maximum value feature in each 2×2 region. Mean pooling takes the average value of the subregions as the mapping result. In Fig. 9.4b, the weights of the convolutional kernels are all 0.25, and the moving step of the convolutional kernels on the input image Input X is 2. Then, the result of the mean pooling is to shrink the input image to 1/4 of the original size, and at the same time, the input image is blurred.

9.4 Steps of Deep Learning Intelligent-Assisted Malaria Detection Algorithms

Most computer-aided malaria detection techniques include four main image processing stages, as follows:

(a) Image preprocessing, where image processing is performed for brightness, sharpness, and size of the image.
(b) Image detection, where the location of red blood cells or the location of the malaria parasite is detected.

(c) Image segmentation to obtain the size of red blood cells and the size and shape of the abnormal region.
(d) Image classification to determine whether it is a worm and the type of worm.

9.4.1 Image Preprocessing

The main purpose of the image preprocessing step is to generate images with low noise and high contrast for further processing. Due to the variable stainability of blood smears and camera adjustments, the light brightness of the microscope capturing the image can also change; this particular problem makes it difficult to classify blood cells and find malaria, as it is difficult to accurately segment and classify things with very similar colors for processing. Many researchers have proposed different preprocessing methods, such as illumination, noise reduction, and image enhancement. Different combinations of filters can be used to reduce the light brightness problems from microscope and camera movements; nevertheless, the production of blood slides due to the nonuniform and nonstandard staining concentration and appearance still requires human involvement.

9.4.2 Image Detection

Image detection is the second step in malaria identification and generally involves the detection of two regions:

1. Detection of areas of red blood cells and then determining whether there are worms inside the red blood cells.

Detecting the abnormal region and then judging whether it is a worm.

9.4.2.1 YOLO

YOLO uses a convolutional network to extract the features and then uses fully connected layers to obtain the predicted values. The network structure refers to the GooLeNet model, which contains 24 convolutional layers and 2 fully connected layers, as shown in Fig. 9.5. For the convolutional layers, 1×1 convolution is mainly used to perform channel reduction, and then 3×3 convolution is immediately followed. For the convolutional and fully connected layers, the leaky ReLU activation function is used. However, the last layer uses a linear activation function (Fig. 9.5). The network structure shows that the final output of the network is a tensor of size. This is consistent with the previous discussion. The specific meaning represented by this tensor is shown in Fig. 9.6. For each cell, the first 20 elements are the category probability values, then 2 elements are the bounding box confidence, which can be multiplied to obtain the category confidence, and the last 8 elements are the bounding box. You may wonder why the confidence and the bounding box are arranged separately instead of in this way, but it is purely for the convenience of calculation, because in fact the 30 elements are corresponding to a cell, and their arrangement is arbitrary. However, separating the arrangement makes it easy to extract each part. Here, to explain, first, the predicted value of the network is a two-dimensional tensor whose shape is. Using slicing, the category probability part is the confidence part, and the last remaining part is the prediction result of the bounding box. In this way, it is very convenient to extract each part, which will facilitate the computation during training and prediction later.

9.4.3 Image Segmentation

Segmentation is an important task in image processing and computer vision research; the other is defined as the process of segmenting a graph into a set of regions that do not overlap. The most important and difficult stage in the automatic classification procedure for analyzing malaria parasites is the accurate segmentation of blood smear images into various elements, such as red blood cells and white blood cells. The purpose of image segmentation is to better segment the different regions and allow us to focus more on the determination of worms. The classical image segmentation algorithms are as follows:

9.4.3.1 FCN (Fully Convolutional Networks)

In general, the last few layers of CNN consist of fully connected layers, and the main function is to transform the feature map output from the convolution layer into a feature vector of specific length and then complete the classification task by the softmax function. In this way, a category prediction can be generated for each pixel in the feature map while preserving the details of the input image, i.e., the feature map can be classified pixel by pixel to complete the segmentation of the image. In short, the fundamental difference between FCN and CNN is that FCN replaces the fully connected layer at the end of CNN with a convolutional layer, and the output of FCN is an image that finely segments the target object, i.e., an image with a target segmentation mask, rather than a probability value. Figure 9.5 shows the comparison

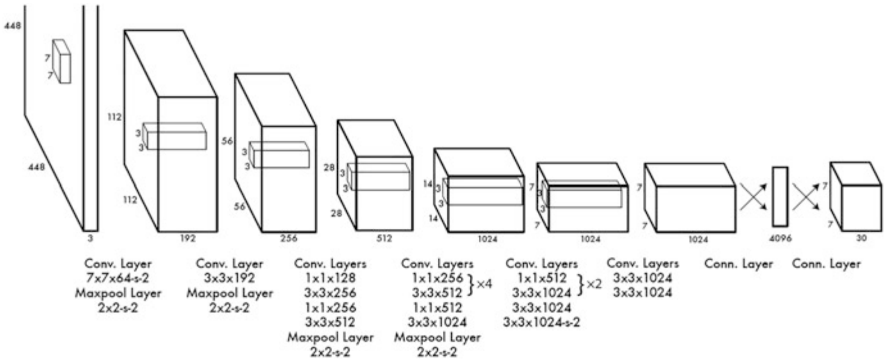


Fig. 9.5 YOLO Network Architecture

of the CNN and FCN network structures. The upper part of the figure is the CNN network structure, and the lower part is the FCN network structure.

The first 5 layers are convolutional layers, and after the convolutional layers, 3 fully connected layers are connected, which is a typical AlexNet network. The output of the 3 fully connected layers is a one-dimensional vector of fixed length, where the length of the last output one-dimensional vector corresponds to the number of categories contained in the data set. The FCN replaces layers 6, 7, and 8 of the CNN with convolutional layers with kernel sizes of (4096,1,1), (4096,1,1), and 21 (1000,1,1), respectively. Successive convolution and pooling of the images generate feature maps with continuously smaller sizes, where the last layer of convolution outputs a feature map called a heatmap. To restore the low-resolution heatmap to the original image size and thus classify each pixel point, a deconvolution operation, or upsampling, is performed on the heatmap. After upsampling, 1000 images of the same size as the original image are obtained. Finally, to perform class prediction for each pixel point, the approach taken in FCN is to find the class with the maximum probability of the location of that pixel point on these 1000 images pixel by pixel and use it as the classification result for that pixel.

Since the feature map is too small, if the original size image is segmented by upsampling the feature map directly, many image details will be lost, making the final segmentation result not fine, so FCN introduces a hopping structure to fuse the prediction with more global information in the last layer and the prediction with more local details in the shallow layer so that both global information and local details can be taken into account when making classification prediction. In this way, both global information and local details can be taken into account in classification prediction to improve segmentation accuracy. Figure 9.6 shows the hopping structure of FCN, in which five successive convolution and pooling operations are performed on the original image to reduce the image to 1/2, 1/4, 1/8, 1/16, and 1/32 of the original image, and then the feature maps of 1/8, 1/16, and 1/32 size of the original image are retained, and finally the fully connected layers in the original CNN are replaced with convolution layers conv6 and conv7. The generated feature map is still 1/32 of the original image, and the feature map is the heatmap. At this time, the network has a 1/32 size heatmap, 1/16 size, and 1/8 size feature map of the original image, and the 1/32 size feature map is directly reduced to the original image size by the deconvolution operation, which will lose some image details, so here we continue to iterate forward, and the 1/16 size and 1/8 size feature maps are also deconvolved in turn to supplement the image details. Finally, the result images of these three deconvolution operations are fused to improve the segmentation accuracy.

FCN is a representative work of deep learning technology applied to image segmentation, which can realize end-to-end image segmentation, although there are some drawbacks of FCN, such as the segmentation results are not fine enough; not sensitive enough to the details in the image; when classifying the image pixel by pixel, the connection between each pixel is not fully considered and lacks spatial 22 consistency. However, since its proposal, FCN has become the basic network

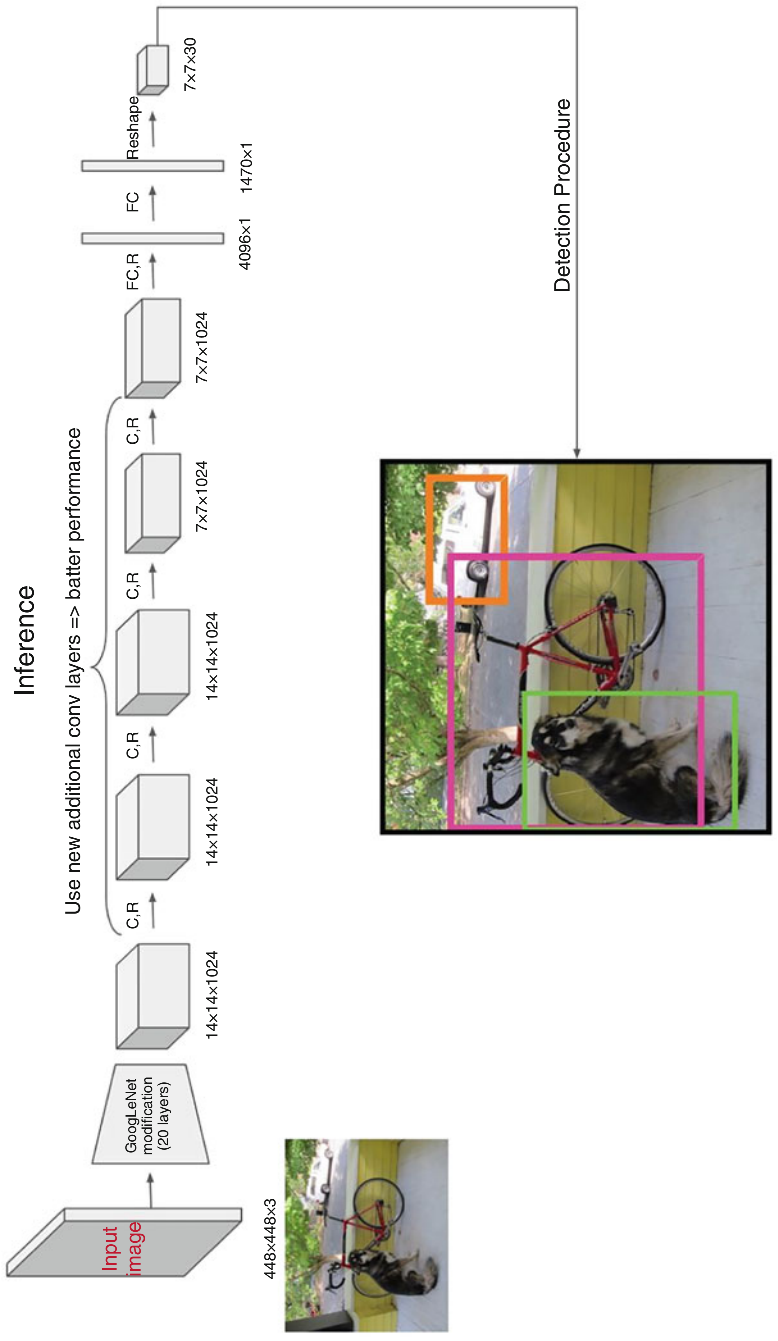


Fig. 9.6 YOLO process

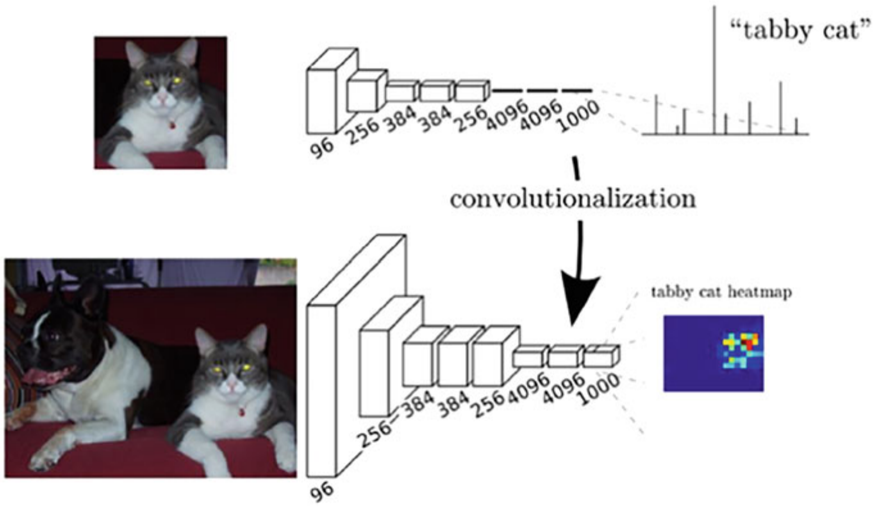


Fig. 9.7 Network structure of the CNN and FCN

framework for semantic segmentation, and many subsequent segmentation algorithms are actually improved on the basis of this framework.

9.4.3.2 U-Net

U-Net was proposed in 2015 and belongs to an improvement of FCN, which can realize end-to-end segmentation. The structure of U-Net is simple but effective and can be regarded as an encoding-decoding structure, where the encoding process gradually reduces the structural information of the image through convolutional and pooling layers while extracting feature information, while the decoding process gradually recovers the structural information of the image through upsampling. The U-Net network structure is shown in Fig. 9.7.

As shown in Fig. 9.7, the network structure of U-Net is symmetrical and shaped like the letter "U", hence the name U-Net. pooling for feature map compression; green arrows represent upsampling for image size recovery; cyan arrows represent 1 × 1 convolution, based on RU-Net's cell segmentation24 for outputting results; gray numbers on the left side of blue and white rectangles indicate image size, and gray numbers on the top indicate the number of channels.

U-Net is a classical fully convolutional network, and the left side of the network (the green rectangular box part in Fig. 9.9) is a CNN architecture consisting of convolutional and pooling layers, i.e., the encoding process, also known as contracting ath. Each block consists of two convolutional layers and one pooling layer, where the convolutional layers use a 3 × 3 convolutional kernel with a ReLU activation function, and the pooling layer uses a maximum pooling of size 2 × 2 with a step size of 2 for downsampling, increasing the number of feature channels to twice the original number after each downsampling. The right side of the network (the red rectangular box in Fig. 9.7) is the decoding process, also called Expansive ath, which also consists of 4 blocks, each of which first increases the size of the feature map to twice the original size by deconvolution and reduces the number of feature channels to half the original size, and then connects the result of deconvolution with the symmetric feature map in the encoding process to form a new feature map. The result of the deconvolution is then connected with the symmetric feature map in the encoding process to form a new feature map. Since the size of the feature map is slightly larger in the encoding process, it is first cropped and then connected, and then the connected feature map is convolved twice in 3×3, and the last layer is convolved in 1×1. The purpose is to convert the feature map of multiple channels into a fixed depth (number of categories, e.g., 2 for binary classification).

The two most important features of the U-Net network are the symmetric "U" structure and hopping connection. The encoding process of U-Net is downsampled 4 times, and symmetrically, its decoding process is also upsampled 4 times to restore the high-level semantic feature map obtained from the encoding process to the resolution of the original image. In addition, U-Net uses jump connections to construct more feature channels in the symmetric phase of encoding and decoding so that the network can fuse the information of the lower-level feature maps with that of the higher-level feature maps through the feature channels to improve the accuracy of image segmentation.

9.4.4 Image Classification

Image classification refers to the classification of unknown images based on the extracted features. In malaria recognition tasks, it is usually the ROI images after image detection and image segmentation, and occasionally the images are classified directly. The classical image classification algorithms are as follows.

Since the breakthrough of the convolutional neural network AlexNet in image classification competition in 2012, some classical convolutional neural network models have been proposed successively and widely used in the field of computer vision in the next 17 years, and several classical convolutional neural network models are introduced below.

9.4.4.1 AlexNet

AlexNet is the first convolutional neural network that has been widely used in the field of computer vision, especially AlexNet, which won the ILSVRC (Image Large Scale Visual Recognition Challenge) in 2012.

In particular, AlexNet won first place in image classification of the ILSVRC (Image Large Scale Visual Recognition Challenge) competition in 2012 and surpassed second place by 10.9%, which has caused a research boom of CNN in the field of computer vision since then.

AlexNet consists of five convolutional layers and three fully connected layers, and the model was initially used to classify images on a data set containing 1000 categories. The AlexNet model architecture is shown in Fig. 9.8.

The AlexNet network has the following features:

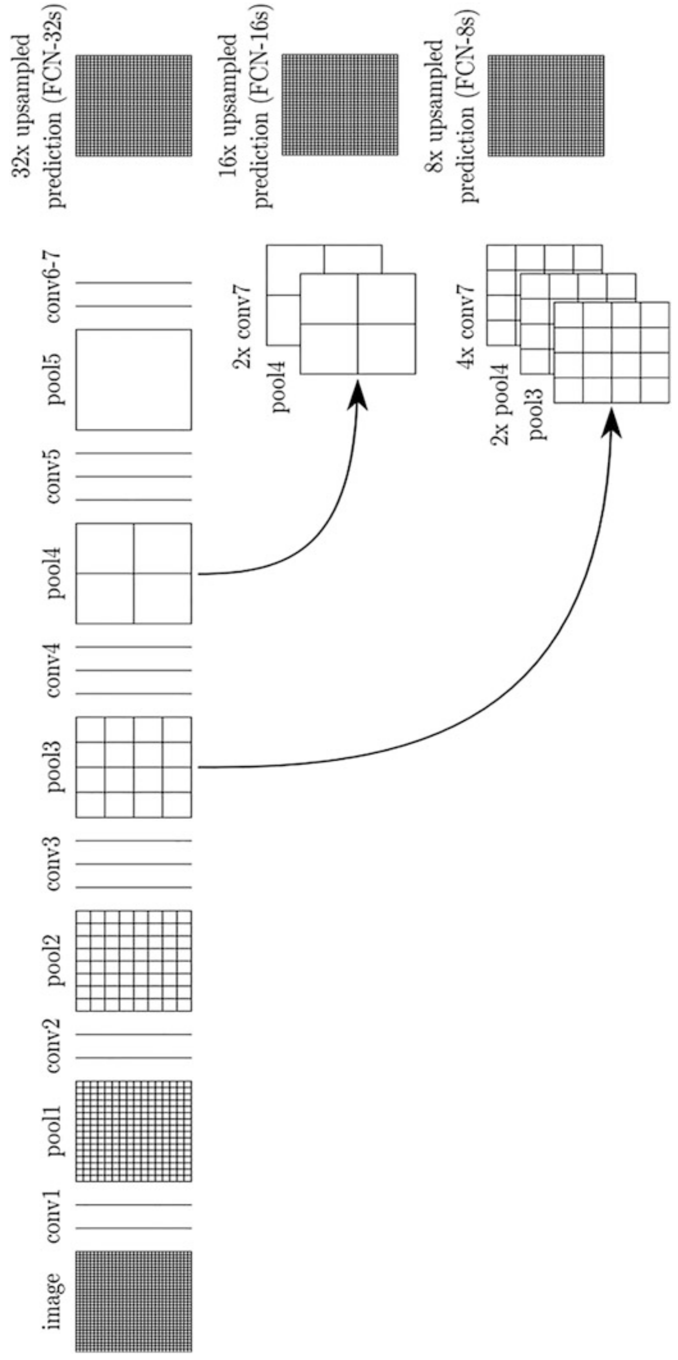


image
conv1
pool1
conv2
pool2
conv3
pool3
conv4
pool4
conv5
pool5
conv6-7
32x upsampled prediction (FCN-32s)
2x conv7
pool4
16x upsampled prediction (FCN-16s)
4x conv7
2x pool4
pool3
8x upsampled prediction (FCN-8s)

Fig. 9.8 Schematic diagram of the FCN skip structure

1. Successful application of the ReLU function as the activation function avoids the problem of gradient disappearance when the sigmoid function is deeper in the network.
2. proposed LocalResponseNormalization (LRN): increase the weights of neurons that contribute more to the classification, while suppressing other neurons that contribute less to the classification, to further improve the generalization ability of the model.
3. Use dropout in the fully connected layer to randomly discard some neurons to avoid overfitting to some extent.
4. Maximum pooling is used to overcome the blurring results caused by mean pooling. In addition, the step size of pooling is smaller than the size of the pooling kernel, which makes the feature maps that have undergone the pooling operation have overlapping parts with each other, thus increasing the feature richness.

9.4.4.2 VGG-Net

VGG-Net, proposed by the Visual Geometry Group, a research group at the University of Oxford, is the underlying network in the model of the winner of the target localization event and the runner-up of the image classification event in the 2014 ILSVRC competition. VGG-Net focuses on the relationship between the network depth of a convolutional neural network and its performance, and it uses a convolutional layer with repeatedly superimposed 3 × 3 small convolutional kernels. It uses the method of repeatedly stacking 3 × 3 convolutional layers with small convolutional kernels and 2 × 2 maximum pooling layers to construct convolutional neural networks with different numbers of layers from 16 to 19.

The whole VGG-Net can be divided into five parts. Each part is connected with multiple 3 × 3 convolutional layers in series, each part is followed by a 2 × 2 maximum pooling, and finally 3 fully connected layers with a softmax layer. There are various structures of VGG-Net; the more common ones are VGG16 and VGG19, among which the VGG16 network is simpler and has good performance. The VGG16 network is simpler, has good performance and is the most widely used. Figure 9.9 shows the network structure of VGG16.

Network structure features:

1. More convolutional layers, deeper network, simpler and more regular structure.
2. The size of the convolutional kernels of VGG-Net is 3 × 3. Its main innovation is to use multiple convolutional layers with 3 × 3 small convolutional kernels instead of one convolutional layer with larger convolutional kernels, which increases the depth of the network and reduces the number of parameters of the network. Because each convolutional layer will be followed by an activation layer, using multiple convolutional layers with small convolutional kernels is equivalent to the network performing more nonlinear mapping, thus improving the nonlinear representation of the network.
3. The LRN layer proposed in AlexNet is discarded. vGG-Net proves that a deeper network can extract features better; VGG-Net also becomes the base network for many subsequent models.

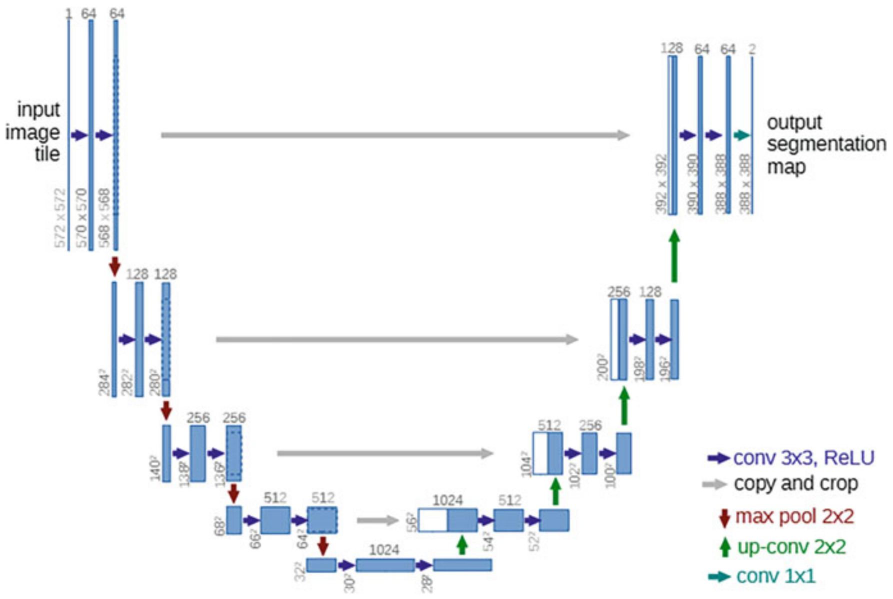


Fig. 9.9 U-Net network structure

9.4.4.3 GoogLeNet

GoogLeNet is a "deep" and "wide" model with 22 layers. Unlike AlexNet and VGG-Net, which only rely on deepening the network to improve its performance, it takes a different approach by considering not only the depth but also the width of the network, introducing the inception module, which uses convolutional kernels of different sizes on the same layer19 to convolve the input image side by side to generate different feature maps and combine these different feature maps in the depth direction. The structure of the Inception module network is shown in Fig. 9.10.

There are four parallel branches in the module. The first three branches use convolutional layers with kernel sizes of 1 × 1, 3 × 3, and 5 × 5 to extract information at different spatial scales, where the middle two branches first perform a 1 × 1 convolution on the input to reduce the number of channels of the input and thus reduce the model complexity. The fourth branch uses a 3 × 3 maximum pooling layer and then takes a 1 × 1 convolution to change the number of channels. Appropriate padding is used in all four branches to keep the input and output sizes consistent in height and width. Finally, the output of each branch is concatenated and fed into the next layer. The inception module extracts semantic features at different levels by different branches, thus maximizing the expressiveness of the network.

GoogLeNet is suitable for processing large amounts of data, especially in the case of limited computational resources or memory, and it has the advantages of high computational efficiency and high classification accuracy (Figs. 9.11, 9.12, and 9.13).

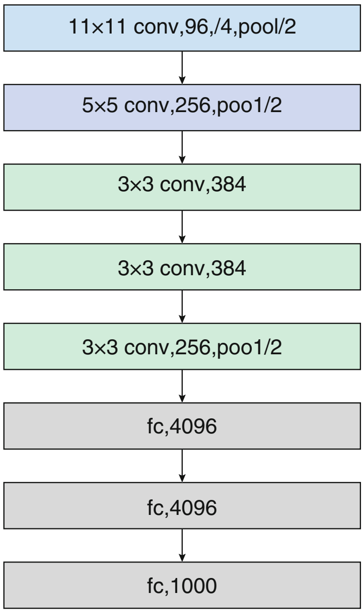


Fig. 9.10 Network structure of AlexNet

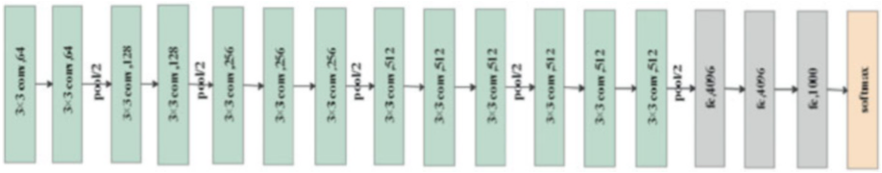


Fig. 9.11 Network structure of VGG16 VGG-Net

9.4.4.4 ResNet

ResNet, proposed by Kaiming He et al. in 2015, was the winner of the ILSVRC competition for the classification task in 2015. The largest contribution of ResNet is the introduction of the residual block to prevent the problem of gradient disappearance due to the increase in network depth, thus further deepening the network depth.

The basic idea of the residual block is to add a constant mapping to the network that allows the input information to be transferred directly to the next layer of the network, and it changes the learning goal of ResNet. Suppose the input of one segment of the network is x and the expected output is $H(x)$. If the output of x is directly transferred to the next segment of the network, then the learning objective becomes $\mathrm{F}(x)=H(x)-x$, the residual at that point.

When $\mathrm{F}(x) = 0$, the input and expected output are equal, i.e., a constant mapping, so the training objective of ResNet is to make $\mathrm{F}(x)$ infinitely close to 0, making the network depth increase without the final accuracy decrease.

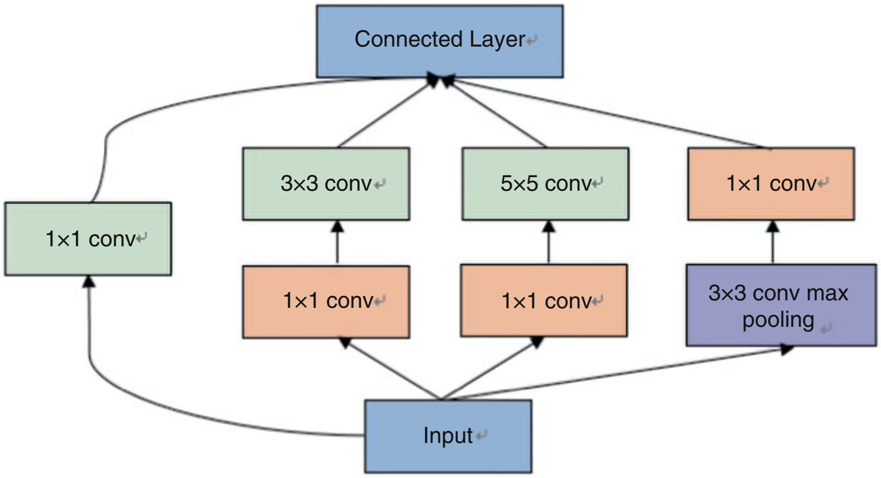
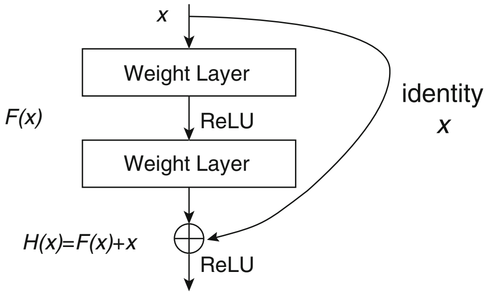


Fig. 9.12 Inception module structure Inception

Fig. 9.13 Residual block structure



9.5 Conclusion

Most of the current segmentation and classification of malaria microscope image cells are based on traditional machine learning or a combination of machine learning and deep learning; both segmentation and classification are carried out in stages, and their classification is generally carried out on the basis of segmentation. For classification, traditional methods require manual participation, human refinement and cleaning of data, and manual construction of features; similarly, for segmentation, traditional image segmentation methods are affected by subjective human operations such as manual selection of initial seed points and initial contours.

References

Díaz G, González FA, Romero E (2009) A semiautomatic method for quantification and classification of erythrocytes infected with malaria parasites in microscopic images[J]. J Biomed Inform 42(2):296–307

He K, Zhang X, Ren S et al (2016) Deep residual learning for image recognition[C]. In: Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition, pp 770–778

Hung J, Carpenter A (2017) Applying faster r-cnn for object detection on malaria images[C]. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp 56–61

Jan Z, Khan A, Sajjad M et al (2018) A review on automated diagnosis of malaria parasite in microscopic blood smears images[J]. Multimed Tools Appl 77(8):9801–9826

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks[C]. Adv Neural Inf Proces Syst 60:1097–1105

Kumarasamy SK, Ong SH, Tan KSW (2011) Robust contour reconstruction of red blood cells and parasites in the automated identification of the stages of malarial infection[J]. Mach Vis Appl 22(3):461–469

Makkapati VV, Rao RM (2009) Segmentation of malaria parasites in peripheral blood smear images[C]//2009. In: IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, pp 1361–1364

Mandal S, Kumar A, Chatterjee J et al (2010) Segmentation of blood smear images using normalized cuts for detection of malarial parasites[C]//2010. In: Annual IEEE India Conference. IEEE, pp 1–4

May Z, Aziz S S A M. Automated quantification and classification of malaria parasites in thin blood smears[C]//2013 IEEE International Conference on Signal and Image Processing Applications. IEEE, 2013: 369-373

Panchbhai VV, Damahe LB, Nagpure AV et al (2012) RBCs and parasites segmentation from thin smear blood cell images[J]. Int J Image Graphics Signal Proc 4(10):54

Rajaraman S, Antani SK, Poostchi M et al (2018) Pretrained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images [J]. PeerJ 6:e4568

Rajaraman S, Jaeger S, Antani SK (2019) Performance evaluation of deep neural ensembles toward malaria parasite detection in thin-blood smear images[J]. PeerJ 7:e6977

Ross NE, Pritchard CJ, Rubin DM et al (2006) Automated image processing method for the diagnosis and classification of malaria on thin blood smears[J]. Med Biol Eng Comput 44(5): 427–436

Somasekar J, Reddy BE (2015) Segmentation of erythrocytes infected with malaria parasites for the diagnosis using microscopy imaging[J]. Computers & Electrical Engineering 45:336–351

Somasekar J, Reddy BE (2017) A novel two-stage thresholding method for segmentation of malaria parasites in microscopic blood images[J]. Journal of Biomedical Engineering and Medical Imaging 4(2):31–31

Tek FB, Dempster AG, Kale I (2009) Computer vision for microscopy diagnosis of malaria [J]. Malar J 8(1):153