manuscript No.
(will be inserted by the editor)

A Configurable Deep Learning Framework for Medical Image Analysis

Jianguo Chen and Nan Yang and Mimi Zhou and Zhaolei Zhang and Xulei Yang

Received: date / Accepted: date

Abstract Artificial Intelligence-based Medical Image Analysis (AI-MIA) has achieved significant advantages in accuracy and efficiency in various biomedical applications. However, most existing deep learning (DL) models have a fixed network structure and aim at single MIA tasks, making it difficult to meet the diverse and complex MIA requirements. To improve the universality of the AI-MIA DL models, we propose a configurable DL framework, which consists of a MIA task element library and a DL model component library. We establish a MIA task element library to accurately define various potential MIA tasks for numerous diseases. In addition, we collect DL computation modules that can be used in different steps of MIA, and built a DL model component library. Given a specific MIA demand of different diseases, the framework

Jianguo Chen
College of Computer Science and Engineering, Hunan University, Changsha, Hunan, 410082, China, and Donnelly Centre for Cellular and Biomolecular Research, University of Toronto, Toronto, ON M5S 3E2, Canada
E-mail: jianguochen@hnu.edu.cn

Nan Yang (corresponding author)
Department of Infectious Disease, the First Affiliated Hospital of Xi'an Jiaotong University, Xi'an, Shaanxi 710061, China
E-mail: nan.yang@xjtu.edu.cn

Mimi Zhou
Department of Infectious Disease, the First Affiliated Hospital of Xi'an Jiaotong University, Xi'an, Shaanxi 710061, China
E-mail: zhoumimiz@stu.xjtu.edu.cn

Zhaolei Zhang
Donnelly Centre for Cellular and Biomolecular Research, University of Toronto, Toronto, ON M5S 3E2, Canada
E-mail: zhaolei.zhang@utoronto.ca

Xulei Yang
Institute for Infocomm Research, Agency for Science Technology and Research, 138632, Singapore
E-mail: yang_xulei@i2r.a-star.edu.sg

can build a personalized DL model by defining MIA tasks and configuring DL model components. Moreover, by comparing with the existing machine learning (ML)/DL models, the structure of the DL model for each configuration can be further adjusted to improve its performance. To demonstrate the effectiveness of the proposed framework, two personalized DL models are designed for upper abdominal organ detection and colon cancer cell classification, achieving high accuracy and high performance. Experimental results on actual medical image datasets show that the configurable DL framework can define specific MIA requirements for different diseases, and flexibly configure DL model components to build suitable personalized DL models. Compared with the state-of-the-art methods, the significant advantage of our framework is that it can generate personalized DL models with a flexible structure, and can continuously fine-tune the model structure to improve model performance.

Keywords Artificial intelligence · configurable framework · deep learning · medical image analysis · personalized model.

1 Introduction

1.1 Motivation

With the rapid development of information science and medical imaging technology, various new medical imaging technologies and Medical Image Analysis (MIA) methods have been emerged [30]. Medical imaging can scan and feed back the state of internal tissues and organs of the human body in a non-interventional manner, and has evolved into an important auxiliary tool for clinical diagnosis and medical research [7, 29]. In recent years, Artificial Intelligence (AI) has achieved technological breakthroughs in the field of image recognition and classification thanks to its advantages such as no-preprocessing, high accuracy, and easy modeling. AI-based MIA (AI-MIA) is one of the principal applications of AI in intelligent medical diagnosis and treatment [1, 24, 40]. AI-MIA has achieved significant benefits in accuracy and efficiency in applications such as lesion feature extraction, lesion detection, tissue localization, and disease screening and diagnosis.

The explosive growth of medical imaging data has brought new opportunities and challenges to AI-MIA. On the one hand, large-scale images of various diseases contain rich disease characteristics and medical information, providing a large amount of training data for AI-MIA applications [30, 33, 39]. In this way, the AI-MIA solutions can train large-scale medical images and realize a high-performance and stable learning model, thereby effectively improving the accuracy of image analysis and disease diagnosis. On the other hand, large-scale medical imaging data sets have also promoted the development of AI-MIA in the breadth and depth directions [5, 12]. More and more diseases use AI technology for image analysis and disease diagnosis. In addition, the diagnosis of each disease may include more diverse and complex MIA tasks [4, 17].

As an important branch of AI, Machine Learning (ML) or Deep Learning (DL) has demonstrated unprecedented performance in many applications of AI-MIA. Facing the complex MIA requirements of various diseases and large-scale medical image data processing, the existing ML/DL methods mainly face the following challenges:

1. Diverse and complex MIA requirements. Different disease tissues and pathologies have different analysis requirements and require different analysis methods. In addition, many MIA applications often require diverse and complex processing steps, such as tissue and organ detection, lesion segmentation, and disease classification. Different steps may require distinct medical image data, resources, operating procedures, and analysis approaches.
2. Insufficient universality of ML/DL models. Although numerous ML/DL models have been proposed for different MIA applications, most existing models have a fixed network structure, which is only for single MIA tasks, and has insufficient universality to meet the needs of diverse and complex MIA applications. In addition, some existing work does not take into full consideration the particularity of medical imaging and the specific requirements of clinical applications. Although they can obtain high-performance results in simulation experiments, they face practical problems such as complex calculations or poor robustness in practical applications.

In this paper, we propose a configurable DL framework to flexibly create various suitable DL models for different MIA tasks, effectively meet the diverse needs of MIA applications, and improve the universality of DL models. The configurable framework is composed of a MIA task element library and a DL model component library, which are respectively created by subdividing the image analysis process and module component decomposition. Using the configurable DL framework, we can precisely define the specific MIA requirements of a certain disease, and create a corresponding DL model with a specific network structure by configuring appropriate model components. Experiments on actual MIA applications show that the proposed framework can construct relatively high-performance DL models for different MIA tasks. The contributions of this paper are summarized as follows.

- To accurately define various potential MIA tasks for numerous types of diseases, we create a MIA task element library. By subdividing common steps in the MIA process, medical image attributes and image analysis tasks are extracted from existing MIA solutions.
- To collect DL computation methods that can be used for MIA, we create a DL model component library. We extract and decompose DL model components from the existing DL models used for MIA and image processing, and establish corresponding formal models.
- Based on the configurable DL framework, we build specific DL models for the given MIA requirements of different diseases by configuring model components. To evaluate the effectiveness of our framework, we construct two DL models for upper abdominal organ detection and colon cancer cell detection and classification.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 introduces the configurable DL framework for MIA, including a MIA task element library and a DL model component library. Section 4 presents the detailed configuration process of the specific DL models of two actual MIA applications. Finally, Section 5 presents conclusions and future work.

2 Related Work

Medical image analysis is widely used in clinical examination and diagnosis of various diseases, such as tumor diagnosis, brain function detection, and cardiovascular disease screening [9, 19, 33]. The locating and detection of diseased tissues, the segmentation and classification of organs and lesions are currently hot research topics in the field of MIA [16]. It is necessary to label and interpret the content and characteristics of medical images in combination with medical expertise. In addition, there are different medical imaging principles in MIA, including Computer Tomography (CT), Magnetic Resonance Imaging (MRI), X-ray, ultrasound, Positron Emission Computed Tomography (PET), and pathological optical microscope [27, 32, 37]. Medical images with different imaging principles have different MIA processing requirements and tasks [30, 40]. However, there are few common image processing methods that can be simultaneously applied to MIA applications of different imaging principles.

In recent years, AI, ML, and DL technologies have achieved technological breakthroughs in image recognition and have been widely used in the field of MIA [12, 27, 36]. The current main research directions of ML/DL technology in MIA include tissue and lesion segmentation, tissue localization and detection, and lesion recognition and classification. For example, in [34], Yuan *et al.* proposed a semantic segmentation DL model for prostate cancer image segmentation. Fedorov *et al.* used a multi-task learning method to optimize the training of a Fully Convolutional Neural Networks (FCN) model and applied it to the segmentation of coronary CT images of the heart [33]. In [13], Kawahara *et al.* proposed an optimized Convolutional Neural Network (CNN) model to predict brain MRI images of children with autism. To solve the problem that deep CNN models are difficult to train, He *et al.* proposed a deep residual model (ResNet), which allows the deep convolutional layer to directly learn the residual mapping instead of the underlying mapping [10]. In terms of tissue localization and detection, Liao *et al.* designed a CNN model based on multiple sets of patches to detect CT images of lung nodules and obtained high recognition performance [17]. In [35], Zakaria *et al.* designed a hierarchical Adaboost neural network for image object detection. In [25], Redmon *et al.* proposed an efficient DL algorithmfor target detection (called "You Only Look Once", YOLO). However, most existing ML/DL models focus on single MIA tasks and have a fixed network structure, making them difficult to meet the diverse and complex MIA requirements.

To improve the universality of DL models, different configurable or adaptive methods have been proposed [3, 7, 21, 22, 28]. Focusing on adaptive DL methods, Sun *et al.* proposed an adaptive DL model for digital pre-distortion, which uses multiple regression neural networks and ensemble predicting modules to make the model more adaptive [28]. In [31], Tang *et al.* used an individual adaptive learning rate to improve the descent process of the DL model for fault diagnosis of rotating machinery. In [21], Ma *et al.* proposed an adaptive auto-encoder network that dynamically adjusts model weights and detects the head and neck cancers. In addition to adaptive DL methods, few studies have focused on configurable ML/DL models [3, 22]. For example, Jim *et al.* designed a configurable deep belief network and applied the proposed model to the field of dementia prediction [22]. In [3], Chakradhar *et al.* proposed a configurable CNN model to match specific parallel computing requirements by adjusting the parameters of the model. However, the existing adaptive and configurable DL methods mainly focused on adjusting the self-structure and parameters of the models, and cannot meet the adaptability to diverse target application requirements.

Due to the complexity of medical images, such as complex tissue shapes, heavy workload of training sample labeling, and imbalance of samples in various categories, medical images of many diseases cannot be directly trained and analyzed by using existing ML/DL models. In addition, most of the existing ML/DL models usually have a fixed network structure and are directed to single MIA tasks, resulting in insufficient universality. In actual application scenarios, it is sometimes difficult to meet diverse and complex MIA requirements.

3 Configurable Deep Learning Framework

The universality of the DL models is an important factor affecting the operation and development of AI-MIA. In this section, we propose a configurable DL framework, which consists of a MIA task element library and a DL model component library. The configurable DL framework can define specific MIA requirements for different diseases, configure DL model components, and build suitable personalized DL models. The overall architecture of the proposed configurable DL framework is illustrated in Fig. 1.

3.1 Construction of MIA Task Element Library

Although the MIA tasks for different diseases are varying, there are some common or similar steps in these tasks, such as Region Of Interest (ROI) recognition, lesion/tissue segmentation, and disease classification. We collect existing MIA application scenarios and extract medical image attributes and image analysis tasks, respectively. In addition, we decompose the common steps in the MIA processes to build a MIA task element library. Based on this,

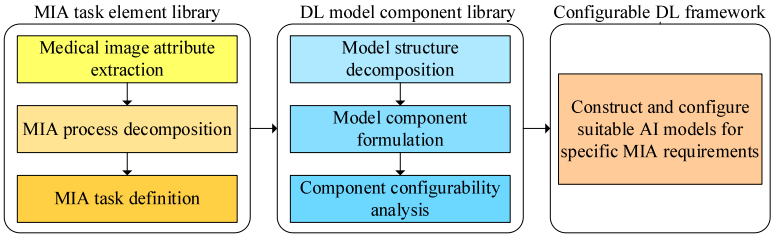


Fig. 1 Architecture of the proposed configurable DL framework for MIA. The framework consists of a MIA task element library and a DL model component library.

we can accurately define new MIA tasks for different diseases. The construction process of the MIA task element library is shown in Fig. 2.

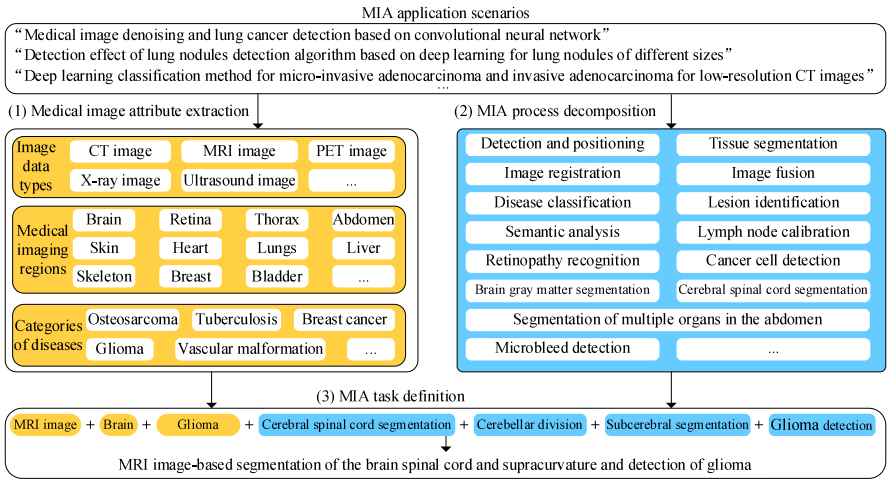


Fig. 2 The construction process of the MIA task element library, including medical image attribute extraction, MIA process decomposition, and MIA task definition. Medical image attributes and image analysis steps are extracted from existing MIA applications. Then, the common steps in the MIA process are decomposed. Based on the medical image attributes and the common steps of MIA, we can define specific MIA requirements for different diseases.

(1) Medical image attribute extraction.

Through literature review and on-site investigation, we collect and analyze existing MIA application scenarios to extract different medical image attributes, including image data types, medical image regions, and disease categories. Let $A = \{T, R, C\}$ be a set of medical image attributes, where T is a set of image data types, R is a set of medical image regions, and C is a set of disease categories. There are six main types of medical images, such as CT images, MRI images, PET images, X-ray images, ultrasound images, and OCT images. Medical imaging regions mainly include the brain, retina, chest cavity, abdomen, skin, breast, lung, and other areas. Disease categories

refer to various diseases for which MIA is used for examination and auxiliary diagnoses, such as tuberculosis, breast cancer, and glioma.

(2) MIA process decomposition.

In actual MIA applications, MIA tasks usually consist of multiple subtasks. By analyzing the specific process of MIA tasks in various diseases, we extract the key steps that have common or similar functions. Then, we evaluate the independence of each MIA step, namely, whether there are quantifiable inputs and outputs. In addition, we define each individual MIA step in detail, including input, analysis process, technical requirements, and output. In this way, a series of MIA steps can be extracted and precisely defined. These steps are used as MIA task elements, thereby building a MIA task element library defined as $E = \{e_1, \cdots, e_i, \cdots, e_{|E|}\}$, where $|E|$ is the number of task elements, and each task element e_i represents an independent MIA step.

(3) MIA task definition.

Based on medical image attributes and MIA task elements, we can flexibly and precisely define various MIA task requirements for specific diseases. Given a certain disease and the corresponding medical images, we set the medical image attributes and define the MIA requirement by combining different MIA steps (task elements) from the MIA task element library. For example, in Fig. 2, in order to detect and diagnose a patient's glioma, we set the image data type to "MRI image", the medical image region to "brain", and the target disease category to "glioma". Then, we select the required MIA task elements, such as cerebral spinal cord segmentation, cerebellar division, sub-cerebral segmentation, and glioma detection. On this basis, we define the MIA task as "Segmentation of brain spinal cord and supracurvature and detection of glioma based on MRI images". The defined MIA task will be used as image analysis requirements to build a corresponding specific DL model.

3.2 Construction of DL Model Component Library

We collect existing DL models in the fields of MIA and related image processing, analyze the model structure to extract common model components, and build a DL model component library. For each DL model component, we construct the corresponding formal model by defining the computation process, input interface, and output interface. The construction process of the DL model component library is shown in Fig. 3.

3.2.1 Model Component Extraction

Various DL models are used in MIA and image processing, including Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Fully Convolutional Neural Network (FCN), Stacked Auto-Encoder (SAE), Variational Auto-Encoder (VAE), Deep Q-Network (DQN), Faster R-CNN, Support Vector MAchine (SVM), Random Forest (RF), etc [13, 20, 33]. We analyze the structure and performance of each DL model, and extract model components

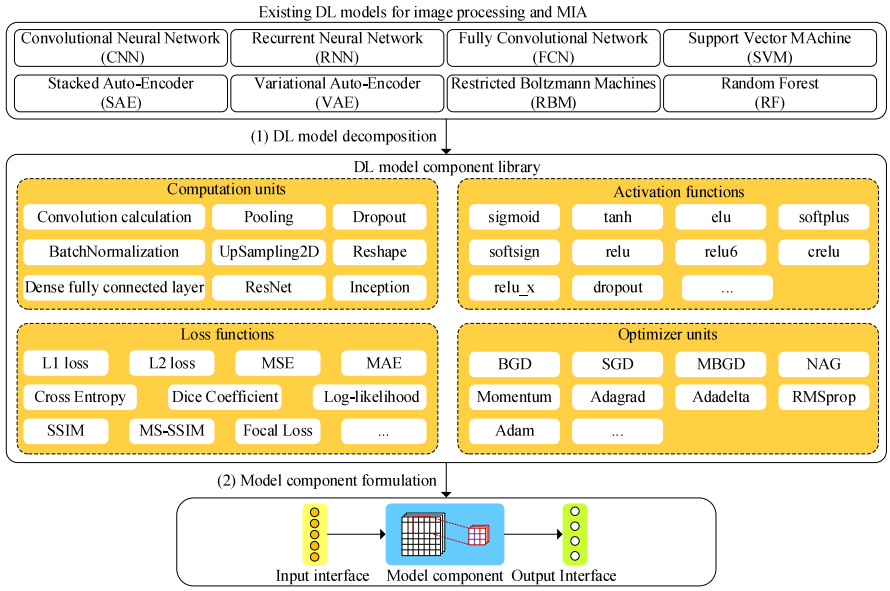


Fig. 3 The construction process of the DL model component library, including DL model decomposition and model component formulation. The structure of various DL models in MIA and image processing are analyzed for model decomposition and component extraction. Then, the formula of each model component is defined to perform model reconstruction.

with distinct structures and functions. In this work, we extract 4 types of DL model components, including computation units, activation functions, loss functions, and optimizer units. In addition, we also define our own novel custom model components according to specific MIA requirements.

1. **Computation units**. We extract various typical network structures and computation units from the existing DL models, including but not limited to: convolutional layer, pooling layer, dropout layer, batch normalization layer, dense fully connected network (FCN), reshape layer, upsampling2D, ResNet, and Inception units.
2. **Activation functions**. Different activation functions are used in existing DL models for image processing, including smooth nonlinear activation functions, continuous activation functions, and random regularization activation functions [26]. Smooth nonlinear activation functions such as sigmoid, tanh, elu, softplus, and softsign, are extracted. The continuous activation functions include relu, crelu, and relu_x. In this work, the dropout function is also regarded as a random regularization activation function.
3. **Loss functions**. Considering that different loss functions have a significant influence on the prediction accuracy and parameter convergence of the DL models, we extract various loss functions that are suitable for MIA [38]. The extracted loss functions include L1 loss function, L2 loss function, Mean Squared Error (MSE), Mean Absolute Error (MAE), Structural Similarity

(SSIM) index, Multi-scale SSIM (MS-SSIM), cross-entropy, dice coefficient, focal loss function, and log-likelihood functions.

4. **Optimizer units**. The DL model component library also involves multiple optimizer units [11], including Batch Gradient Descent (BGD), Stochastic Gradient Descent (SGD), Mini-Batch Gradient Descent (MBGD), Momentum Adaptive gradient algorithm (Adagrad), RMSprop, and Adaptive moment estimation (Adam).
5. **Custom model components**. Taking into account the unique characteristics of medical imaging data and MIA tasks (i.e., imbalanced samples in various categories, heterogeneous imaging data, and multi-modal data fusion), we define our own model components, such as data argumentation components, image transformer components, etc. In addition, we also provide an extensible component interface to allow users to further define their own computation components and add them to the model component library.

3.2.2 Model Component Formulation

For each extracted DL model component, we establish the corresponding formal model by defining its input interface, computation process, and output interface. Due to space limitations, we choose four components as examples to illustrate the construction process of the formal model.

(1) Computation unit - convolution computation component.

Take the convolution calculation component as an example to briefly explain the formulating process of the computation units. This component is used to perform convolution calculations on the input matrix for image feature extraction. The main model parameters of this component include activation function $f()$, convolutional kernel weight matrix F (with the size of $W_F \times H_F$), bias item w_b, convolution step size S, and zero-filling number P. The formal model of the convolution computation component is shown in Fig. 4.

As shown in Fig. 4, the convolution calculation component performs convolution on the elements in the input matrix of each medical image, and outputs the image feature matrix. The convolution function is defined as:

$$A = X \otimes F = f\left(\sum_{d=0}^{D-1} X_d \times F_d + w_b\right), \tag{1}$$

where D represents the depth of the input matrix, and X_d represents the d-th layer matrix. The input is an image matrix or feature matrix of size $W_{\text{In}} \times H_{\text{In}} \times D$, and the output is a feature matrix of size $W_{\text{Out}} \times H_{\text{Out}} \times D$. In this way, we can establish a formal model of the convolution computation component, and define a definite calculation process, model parameter set, input interface, and output interface.

(2) Custom image augmentation components

Although radiology departments in hospitals around the world produce massive amounts of clinical medical images, most of them cannot be directly

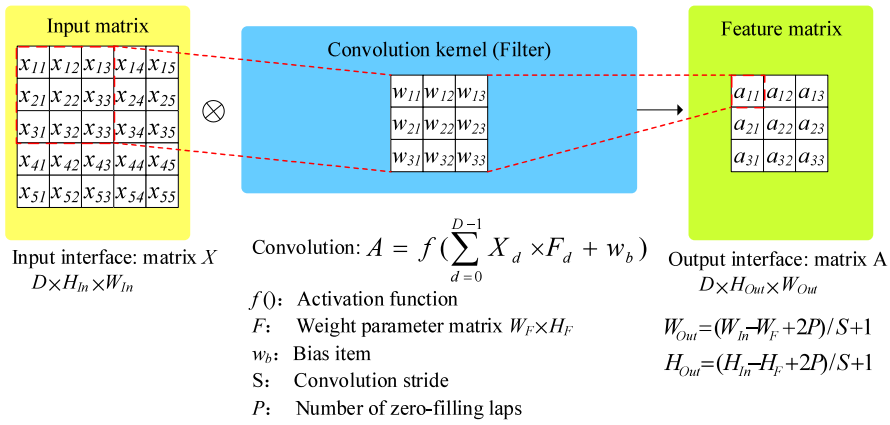


Fig. 4 Formal model of the convolution computation component, where the input interface is a medical image matrix, the computation process is a convolution operation, and the output interface is an image feature matrix.

used for DL model training and data analysis. Data preprocessing is an important step in MIA, including sub-processes such as multi-modal data fusion, image registration, and image augmentation. For medical image augmentation, most existing solutions only perform data augmentation on training samples, but ignore the corresponding ground truth or labels. In this work, in addition to the existing DL model components, we further design our own image augmentation components to expand the size of medical images. The designed data augmentation components can well solve the problem of the available small-size medical image data sets and the imbalance of samples between different categories. The structure of the designed image augmentation component is shown in Fig. 5.

We collect more than 24 transformation operations and use them as image augmentation elements, such as affine transformation, color transformation, pixel overlay, noise, blur, rotate, flip, invert, sharpen, crop, etc. Then, we design multiple transform-based image augmentation components for different types of medical images. The augmentation components can generate large-scale diverse and annotated training datasets with the corresponding segmentation ground truth for DL model training of medical image segmentation. Given a medical image, we can create an augmentation strategy by randomly selecting multiple transform elements and performing augmentation on the image to generate diverse images. At the same time, the augmentation strategy can be fixed and executed on the corresponding ground truth to generate matching ground truth for the augmented image. That is, for each image sample and its segmentation ground truth, we perform the same stochastic augmentation combination on them to generate different new samples and corresponding labels. In addition, we note that in medical images, the position and morphology of some organs or tissues have important medical significance, while some do not. For example, the position of the heart in chest CT and the position of

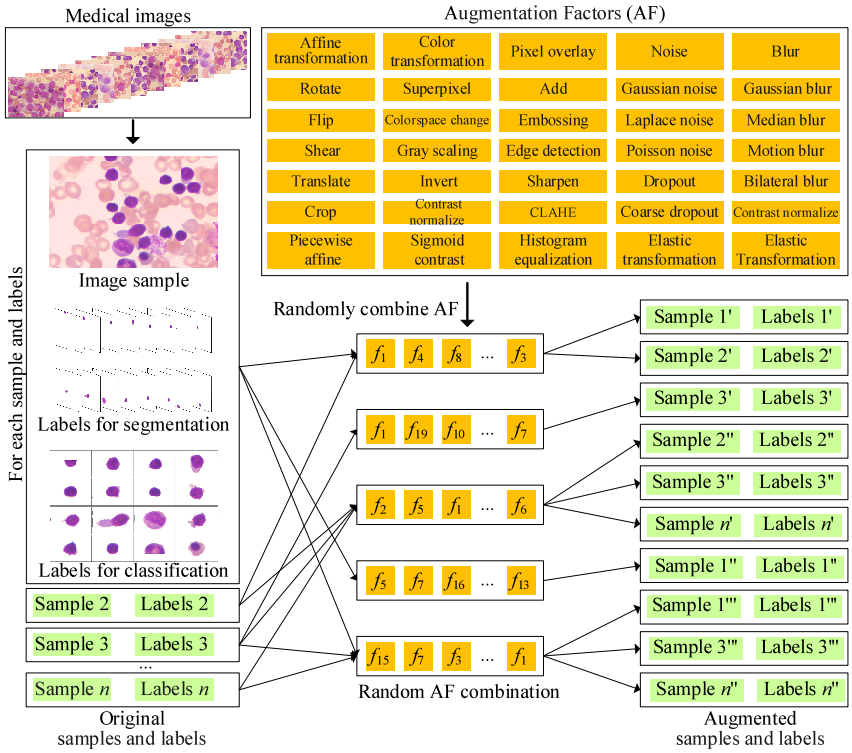


Fig. 5 Structure of the designed image augmentation components.

the liver in abdominal CT do not allow rotation and mirroring. In contrast, the position of the platelets in a bone marrow smear and the position of the lesion in a dermoscopic image allows rotation and mirroring. Therefore, when selecting augmentation elements, we care about whether image augmentation will change the medical characteristics and significance of medical images, and set the range of available elements. In this way, the augmented samples and labels also match each other, thereby realizing the synchronous augmentation of the samples and their labels.

(3) Activation function - ReLU.

The activation function in a DL model is responsible for generating a nonlinear decision boundary through a nonlinear combination of weighted inputs. For example, the Rectified Linear Unit (ReLU) is a typical activation function for many DL models. Assuming that the input of ReLU is x, ReLU is defined as:

$$f(x) = \begin{cases} 0, \text{ if } x \leq 0, \\ x, \text{ if } x > 0. \end{cases} \tag{2}$$

We further define the input and output interfaces of the ReLU function, where the input of ReLU is a real number x and the output is also a real number $f(x)$.

(4) Loss function - Softmax.

The loss function in a DL model is used to calculate the error between the predicted value and the actual value and to evaluate the accuracy of the model. Softmax is a loss function widely used in image classification solutions. Taking the Softmax function as an example, we explain the formula of the loss function. It distinguishes features of different categories by maximizing the posterior probability of the true label value. Given an input x_i and the corresponding label y_i, the Softmax loss function is defined as:

$$\begin{aligned} J_{\text{Softmax}} &= -\frac{1}{m}\sum_{i=1}^{m}\log\frac{\exp\left(f_{y_i}\right)}{\sum_{j=1}^{n}\exp\left(f_j\right)} \\ &= -\frac{1}{m}\sum_{i=1}^{m}\log\frac{\exp\left(W_{y_i}^{T}x_i+b_{y_i}\right)}{\sum_{j=1}^{n}\exp\left(W_j^{T}x_i+b_j\right)}, \end{aligned} \tag{3}$$

where x_i represents the feature of the i-th image, y_i is the category label of x_i, W_j is the weight of the category j, b_j is the deviation of j, and m and n represent the number and category of training samples, respectively. When the fully connected layer is activated, f_j represents the inner product between W_j and b_j, namely, $f_j = W_j^T x_i + b_j$. We further define the input and output interfaces of the Softmax function, where the input is the sample x_i and the label y_i, and the output is the real value $J_{Softmax}$.

(5) Optimizer unit - Stochastic Gradient Descent (SGD).

In a DL model, the optimization goal of the optimizers is the weight parameters of the entire model, and the objective function is the loss function. By minimizing the loss function of the DL model, the values of the weight parameters are optimized to stabilize the model. SGD, also known as incremental gradient descent, is an iterative method for optimizing the differentiable objective functions of the DL model. SGD calculates the gradient of the small-batch data loss function, and iteratively updates the weight parameters and bias items. Suppose there are m training samples $X = \{x^{(1)}, \cdots, x^{(i)}, \cdots, x^{(m)}\}$, where $y^{(i)}$ is the label of the i-th sample $x^{(i)}$; $\theta = (\theta_1, \cdots, \theta_j, \cdots, \theta_n)$ is the weight parameter set. n is the number of features of each sample, and the formula of SGD is defined as:

$$\theta_j = \theta_j + \frac{\alpha}{m}\left(y^{(i)} - h_\theta(x^{(i)})x_j^{(x)}\right), \tag{4}$$

where α is the learning rate of SGD. We further define the input and output interfaces of SGD. The input interface of SGD is $(X = \{x^{(1)}, \cdots, x^{(m)}\}, Y = \{y^{(1)}, \cdots, y^{(m)}\}, \theta = (\theta_1, \cdots, \theta_n))$, where θ is the initial weight parameter set. The output interface of SGD is $\theta' = (\theta'_1, \cdots, \theta'_n))$, where θ' is the optimized set of weight parameters.

We implement our DL model component library based on the PyTorch platform [23] in Python language, which is a popularly open-source DL platform. PyTorch provides two advanced functions, including tensor calculation with powerful GPU acceleration, and a deep neural network based on a tape

auto-tuning system, thereby accelerating research from prototypes to production deployment. Based on the MIA task element library and the DL model component library, we develop the configurable deep learning platform. The framework can build and configure personalized DL models by configuring model components for specific MIA requirements of different diseases. In the subsequent construction of each configurable DL model, we can add multiple components to a DL model, and set the component parameters and interfaces to meet specific MIA requirements.

4 Personalized Deep Learning Models for Medical Image Analysis

In this section, to demonstrate the effectiveness of the proposed configurable DL framework, we construct two personalized DL models for different actual MIA requirements. The first DL model is used for the detection of upper abdominal organs, and the second model is used for the detection and classification of colon cancer cells.

4.1 DL Model for Upper Abdominal Organ Detection

We use the configurable DL platform to build a DL model for upper abdominal organ detection by selecting and configuring model components based on the model component library. Based on the MIA task element library, we can accurately define the MIA requirements for upper abdominal organ detection and construct a special DL model to meet the MIA requirements. In addition, we conduct comparative experiments on actual abdominal CT images to evaluate the effectiveness of the constructed DL model.

4.1.1 MIA Requirement Definition

CT scan is currently the main imaging method for the diagnosis of liver, kidney, stomach, intestine, colorectum and other abdominal organ diseases. The abdomen is a place where the organs are densely distributed and the lesions are concentrated. The liver, spleen, stomach, kidney, and other organs are common disease organs in adults. There are many organs in the abdomen, and the shape, size, and position of each organ on an abdomen CT image are different at different angles. Therefore, the accurate detection of the regional position of each organ in abdominal CT images is an important preliminary work of MIA. Examples of the upper abdominal CT images are shown in Fig. 6.

Based on the acquired CT images, we can define the MIA requirement as follows.

- Image data type: CT images.
- Medical image regions: upper abdomen.
- Disease category: N/A.

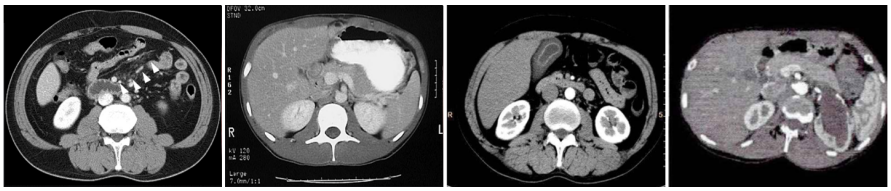


Fig. 6 CT images of the fetal upper abdomen.

– MIA task elements (steps): (1) liver detection, (2) gallbladder detection, (3) stomach detection, (4) intestine detection, (5) pancreas detection, and (6) kidney detection.

4.1.2 Upper Abdominal Organ Detection Model

According to the defined MIA requirements, we manually select the corresponding model components from the DL model component library and manually configure these components to build a personalized DL model (called UAOD model) for upper abdominal organ detection. The UAOD model is equipped with a data augmentation component, two fully convolutional residual networks, and an organ index unit. The network architecture of the UAOD model is illustrated in Fig. 7.

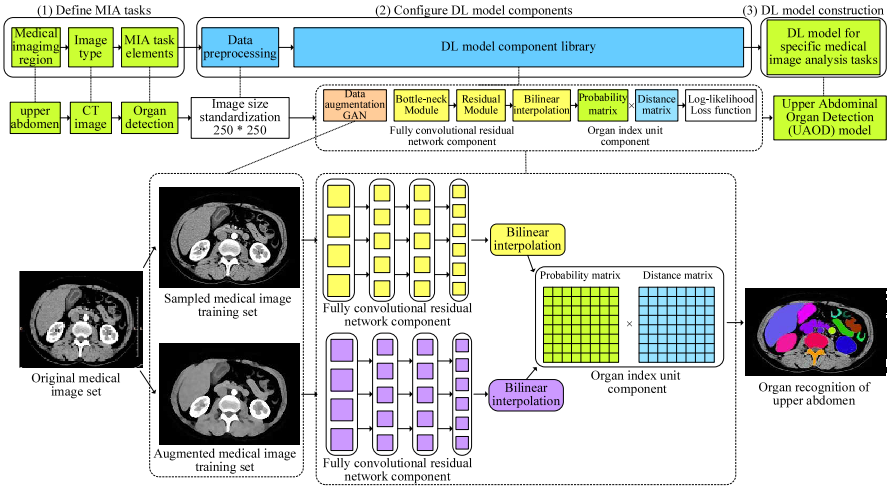


Fig. 7 Network architecture of the configured upper abdominal organ detection (UAOD) model. The MIA requirements for upper abdominal organ detection are defined by setting the medical image type, imaging region, and MIA task elements. Then, the UAOD model is created by configuring a data augmentation component, two fully convolutional residual networks, and an organ index unit.

Specifically, in order to solve the problem of small-size data samples and imbalance between categories, we use a data augmentation GAN component

Table 1 Component configuration of the UAOD model.

Level	Component name	Component parameters	Input interface	Output interface
L_0	Input		512×512	
L_1	Data augmentation component	GAN model	original CT images	augmented CT images
L_2	Fully convolutional residual network 1	$4 \times 4 \times 12$	512×512	$127 \times 127 \times 12$
L_3	Fully convolutional residual network 2	$4 \times 4 \times 12$	512×512	$127 \times 127 \times 12$
L_4	Bilinear interpolation		$127 \times 127 \times 12$	$127 \times 127 \times 12$
L_5	Organ index unit	32×32	$127 \times 127 \times 12$	32×32

to expand the small sample data set. Then, we use a bottle-neck component and a residual component to extract the structural characteristics of multiple target tissues from the input image. On this basis, a bilinear interpolation component is further used to fuse the structural characteristics from the two sub-networks. A probability matrix and a distance matrix are used to calculate the error between the structural characteristics of each target tissue and the ground truth. Finally, we use the log-likelihood loss function to train the model.

Table 1 describes the detailed component configuration of the proposed UAOD model, including component names, kernel parameters, input interfaces, and output interfaces. As shown in Fig. 7 and Table 1, the original CT images of the upper abdomen are preprocessed, and the size of each image is normalized to 250×250 pixels. The components in the UAOD model are designed below.

(1) Data augmentation component.

To reduce the impact of data imbalance (that is, the severely inconsistent proportions of normal samples and diseased samples) on the UAOD model, we use a data augmentation component to increase the number of small-category samples. The Generative Adversarial Nets (GAN) [8] is attached to the UAOD model to train small-category samples in upper abdominal images and generate simulated images. The GAN component contains a generator and a discriminator. The generator G receives the input random variable z, which follows an artificially selected prior probability distribution. Then, G uses a multi-layer perceptron network to parameterize the derivable map $G(z)$ and map the input space to the sample space. The discriminator D receives the input of the real sample x, the generated sample $G(z)$, and the real and fake labels. Then, D uses a parameterized multi-layer perceptron network to calculate the probability $D(x)$ of $G(x)$ from the real sample x. The generator G and discriminator D play the roles of two competitors in the mini-max game, as defined as:

$$\begin{aligned} &\min_G \max_D V(D,G) = \\ &\mathbb{E}_{x,\ p(x)}[\log D(x)] + \mathbb{E}_{z,\ p(z)}[\log\left(1 - D\left(G(z)\right)\right)]. \end{aligned} \tag{5}$$

If the input of D comes from a real image, the output is marked as 1, otherwise the output is 0. Based on the GAN component, each sample in a small category of the upper abdominal CT images is trained to generate more augmented samples, thereby balancing the proportion of each category in the training set.

(2) Fully convolutional residual network component.

Two fully convolutional residual network (FCRN) components [15] are added to the UAOD model to extract the features of the organ areas in the upper abdominal images. The first FCRN is trained on the sampled training subset, and the second FCRN is trained on the augmented subset. Each FCRN consists of a bottleneck module and a residual module, where the former contains 3 convolutional layers and the latter contains 6 convolutional layers. They use jumper connections to directly transmit forward and backward signals from one module to another.

(3) Organ index unit component.

Based on the FCRN component, we add an Organ Index Unit (OIU) component to calculate the contribution of each pixel in the organ region to the organ classification. We perform bilinear interpolation on the organ features obtained from the FCRN component, and create a probability matrix based on the prediction results. Let $v_i(x, y)$ be the value of the organ region (x, y) in the i-th probability matrix, and the standardized possibility ρ_i of the organ region is defined as:

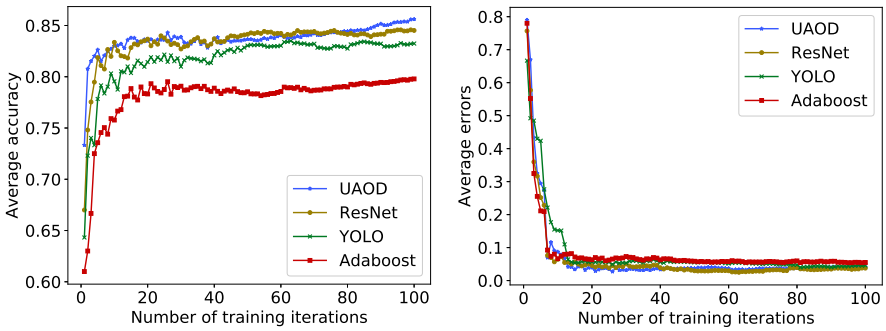
$$\rho_i(x, y) = \frac{v_i(x, y) - \min_{i \in R} v_i(x, y)}{\sum_{i \in R} v_i(x, y)}. \tag{6}$$

The distance from each pixel in the target organ region to the nearest boundary indicates the importance of the pixel to organ classification. In this way, we can obtain the distance matrix. Then, we multiply the distance matrix by the standardized possibility to get the probability distribution matrix for classification. Finally, the classification probabilities of the organ region in the probability distribution matrix are averaged to obtain the values of different types of organs.

4.1.3 Performance Evaluation of UAOD Model

To evaluate the performance of the constructed UAOD model, we conduct experiments by comparing the UAOD model with other ML/DL models used for the same image analysis task. We perform the Adaboost [35], YOLO [25], ResNet [10], and UAOD models on the upper abdomen CT images from Affiliated Shenzhen Maternal and Child Healthcare. 87,920 images are collected, of which 70,336 images are used as the training set and 17,584 images are used as the test set. We use 5-fold cross-validation to assess the performance and versatility of the comparison models. The experimental results of the comparison models are shown in Fig. 8 and Table 2.

It can be observed in Fig. 8 and Table 2 that the proposed UAOD model can achieve high detection accuracy similar to the state-of-the-art DL models.



(a) Average accuracy (b) Average errors

Fig. 8 Performance of DL models for upper abdominal organ detection.

Table 2 Performance comparison of comparison models.

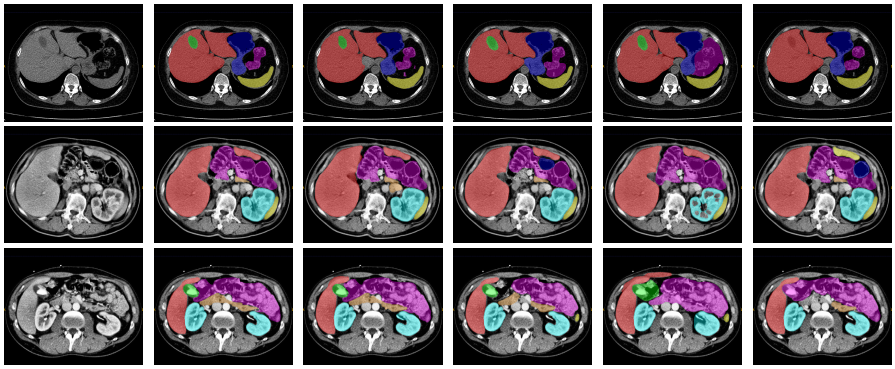
Data sets	UAOD	ResNet	YOLO	Adaboost
Training set	85.52%	83.33%	81.65%	78.84%
Validation set	82.21%	80.79%	78.11%	74.42%

Benefiting from the residual mapping operation, UAOD can retain gradient information in the deep network structure, and can effectively understand the characteristics of organs and their changes. The proposed UAOD model achieves a high accuracy of 85.52% on the training set and 82.21% on the validation set. In contrast, the Adaboost method uses cascaded classification to cascade multiple weak classifiers to form a strong classifier based on organ features, so as to achieve the lowest accuracy when the organ features are slightly different. As shown in Table 2, the Adaboost model achieves the lowest accuracy of 78.84% on the training set and 74.42% on the validation set. In the YOLO model, the prediction of bounding boxes enhances the control constraints, and each grid unit can predict only two bounding boxes. The objects in the two bounding boxes belong to the same category, which limits YOLO's accuracy in predicting small organs that are close to each other. The experimental results prove that the UAOD model can complete the MIA task of organ detection and has high effectiveness. Moreover, by adjusting and optimizing the structure of the UAOD model, we can further obtain a relatively efficiency DL model for upper abdominal organ detection.

4.1.4 Comparison Experiments on Actual CT Images

We conduct comparative experiments on actual CT images of the upper abdomen, and evaluate the effectiveness of the established UAOD model by comparing with ResNet [10], YOLO [25], and Adaboost [35] algorithms. We collect CT images of the upper abdomen from the Department of Infectious Disease of the First Affiliated Hospital, Xi'an Jiaotong University, China. The

MIA tasks of the UAOD model is to detect the abdominal organs of the liver, gallbladder, pancreas, spleen, stomach, intestines, and kidneys. We use 17,584 images as the test set to compare the detection accuracy of the comparison models. Examples of detection results provided by the comparison algorithms are shown in Fig. 9 and Table 3.



(a) Input (b) Ground truth (c) UAOD (d) ResNet (e) YOLO (f) Adaboost

Fig. 9 Visualization of detection results of abdominal organs. The first column is the three original CT images of the upper abdomen, and the second column is the corresponding ground truth. The third to sixth columns are the detection results of UAOD, ResNet, YOLO and Adaboost algorithms.

Table 3 Detection performance of comparison models on abdominal organs.

Organs	UAOD	ResNet	YOLO	Adaboost
Liver	84.19%	82.48%	83.14%	79.67%
Gallbladder	82.01%	80.36%	79.56%	75.77%
Pancreas	80.78%	74.81%	72.30%	63.00%
spleen	79.33%	67.00%	64.32%	61.00%
kidney	80.31%	81.71%	79.45%	79.52%

Table 3 describes the detection performance of the comparison models for abdominal organs. It can be seen from Fig. 7 and Table 3 that the proposed UAOD algorithm can identify each organ in the three images with the highest accuracy. In contrast, ResNet incorrectly identifies an intestinal organ like the stomach in the second case, and an incomplete intestine in the third case. The YOLO algorithm incorrectly identifies the extra tissue (such as aorta and intestine) as the liver in the first case, and the tissue around the gallbladder as the gallbladder in the third case. The Adaboost algorithm has the lowest recognition accuracy, where part of the intestines in the first case is incorrectly identified as the pancreas, the air cavity in the intestines in the second case is

incorrectly identified as the stomach, and the gallbladder in the third case is ignored. Therefore, the comparative experimental results show that the proposed UAOD algorithm can effectively complete the detection of abdominal organs and achieve the highest accuracy.

4.2 DL Model for Colon Cancer Cell Detection and Classification

In this section, we use the MIA task element library and the DL model component library to create a personalized DL model to accurately detect and classify cancer cells in histopathological images. Then, we conduct comparative experiments on actual colon histopathological images to evaluate the performance of the constructed DL model.

4.2.1 MIA Requirement Definition

Identifying cancer cells under microscopes is the main content of pathological examination and the key to cancer diagnosis. We collect colon cancer histopathological images from the Catalogue Of Somatic Mutations In Cancer (COSMIC) [2]. COSMIC is the world's largest source of somatic mutation information related to human cancer. Examples of collected colon cancer histopathological images are shown in Fig. 10.

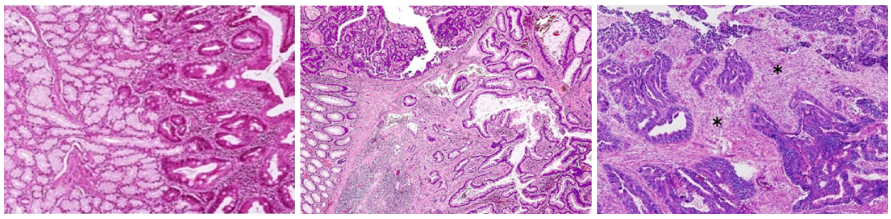


Fig. 10 Colon cancer histopathological images.

In colon tissues, atypical hyperplasia and malignant epithelial cells often have irregular chromatin structure and appear to be highly chaotic without clear boundaries. In addition, changes in the appearance of the same type of cells inside and between samples are also factors, which makes it difficult to classify cells. Based on the collected histopathological images of colon cancer, we can define the MIA requirement as follows.

- Image data type: histopathological images.
- Medical imaging regions: colon tissues.
- Disease category: colon cancer.
- MIA task elements (steps): (1) colon cancer cell detection, and (2) cell classification.

4.2.2 Colon Cancer Cell Detection and Classification Model

According to the defined MIA requirements, we manually select model components from the DL model component library to build a personalized DL model (called C3DC model) for the detection and classification of colon cancer cells. The C3DC model is equipped with a cancer cell detection component and a cancer cell classification component. The network architecture of the proposed C3DC model is illustrated in Fig. 11 and the detailed configuration is described in Table 4.

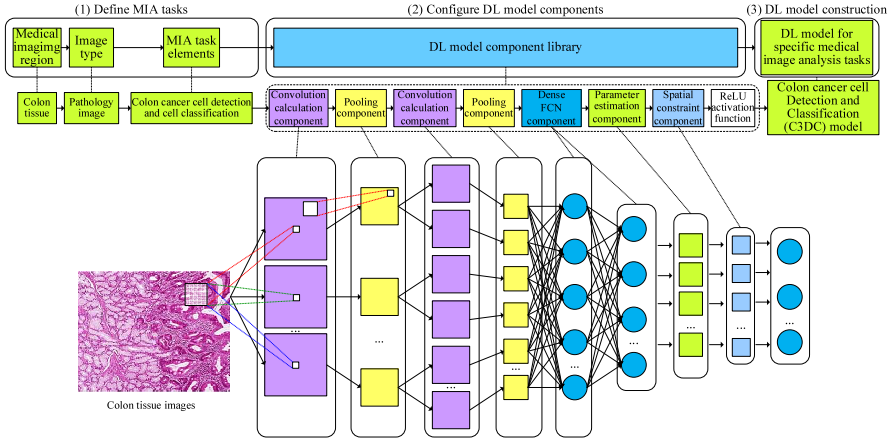


Fig. 11 Network architecture of the configured colon cancer cell detection and classification (C3DC) model. The MIA requirements for colon cancer cell detection and classification are defined by setting the medical image type, imaging region, and MIA task elements. Then, the C3DC model is created by configuring a cancer cell detection component and a cancer cell classification component.

Specifically, we use a convolutional layer to receive the input colon tissue images and extract the image features of the original data. Then, we use a pooling component to reduce the dimensionality of the features. On this basis, we further use a convolutional layer and a pooling layer to extract abstract image features. Based on the extracted features, the classification index of the image is obtained through the dense fully connected layer. Finally, the parameter estimation component and the spatial constraint component are attached to the model to obtain the final classification result.

(1) Input of the C3DC model.

Because the nucleic acid in colon cancer cells has been stained with hematoxylin, this feature is a fairly good indication of cell location. When detecting colon cancer cells, hematoxylin intensity is selected as the input feature of each patch of the C3DC training model. In cell classification, cell morphology (including shape, size, color, and texture) is used to distinguish different types of cancer cells. The input patch size is set to $W \times H \times D = 27 \times 27 \times 1$, and

Table 4 Component configuration of the C3DC model.

Level	Component name	Component parameters	Input interface	Output interface
L_0	Input		$27 \times 27 \times 1$	
L_1	Convolutional layer 1	$4 \times 4 \times 1 \times 36$	$27 \times 27 \times 1$	$24 \times 24 \times 36$
L_2	Pooling layer 1	2×2	$24 \times 24 \times 36$	$12 \times 12 \times 36$
L_3	Convolutional layer 2	$3 \times 3 \times 36 \times 48$	$12 \times 12 \times 36$	$10 \times 10 \times 48$
L_4	Pooling layer 2	2×2	$10 \times 10 \times 48$	$5 \times 5 \times 48$
L_5	Fully connected layer 1	$5 \times 5 \times 48 \times 1024$	$5 \times 5 \times 48$	1×1024
L_6	Fully connected layer 2	$1 \times 1 \times 1024 \times 512$	1×1024	1×512
L_7	Parameter estimation component	$1 \times 1 \times 512 \times 20$	1×512	1×20
L_8	Spatial constraint component		1×20	10×10
L_9	Neighbor integrated predictor		10×10	$1 \times M$

the output image size of the C3DC model is set to $W' \times H' = 1 \times M$, where M is the number of colon cancer cell categories.

(2) Cancer cell detection component.

Given the input patch $x \in \mathbb{R}^{H \times W \times D}$ of a histopathological image, the height is H, the width is W, and the deep is D. We add a cancer cell detection module to the C3DC model to detect the central position of each cell in x. The output $y \in [0,1]^{H' \times W'}$ is defined as a probability matrix of size $H' \times W'$. Suppose $\Omega = H' \times W'$ is the spatial domain of y, d is the constant radius, and x and x_m^0 represent the output value y_j and the coordinates of the m-th cell in the spatial domain Ω, respectively. Each element y_j ($j \in \{1, \cdots, |\Omega|\}$) in y is defined as:

$$y_j = \left(1 + \frac{(\|x - x_m^0\|_2^2)}{2}\right)^{-1}, \tag{7}$$

if $\forall m \neq m', \|x - x_m^0\|_2 \leq \|x - x_{m'}^0\|_2 \leq d$, otherwise, $y_j = 0$. It is clear from the probability matrix y that there is a peak probability near the center of each cell, while it is flat elsewhere.

The parameter estimation component L_7 and the spatial constraint component L_8 are attached to the C3DC model to predict the probability that each pixel is the center of the cell. The corresponding predicted output value $\hat{y}$ is generated from the spatial constraint component (layer L_8) of the C3DC model. According to the known structure of the training output probability map, the i-th element of the predicted output is defined as:

$$\begin{aligned} \hat{y_j} &= g(z_j, \hat{z}_1^0, \cdots, \hat{z}_M^0, h_1, \cdots, h_M) \\ &= \begin{cases} \frac{1}{1+\frac{(\|z_j - z_m^0\|_2^2)}{2}} & \text{if } \forall m \neq m', \|z_j - z_m^0\|_2 \leq \|z_j - z_{m'}^0\|_2 \leq d \\ 0 & \text{other} \end{cases} \end{aligned} \tag{8}$$

where $\hat{z}_M^0 \in \Omega$ is the estimated cell center, $h_m \in [0,1]$ is the height mask of the m-th probability mask, and M represents the maximum probability mask in $\hat{y}$.

(3) Cancer cell classification component.

Based on cell detection, we continue to use the designed C3DC model to classify these cells. For cell classification, a neighborhood integrated prediction strategy is introduced, and cell morphology (including shape, size, color, and texture) is used to distinguish different types of cancer cells. Let $X \in \mathbb{R}^{H\times W\times D}$ be the set of all patches ($H \times W \times D$) of a histopathological image, and $x_i \in X$ is the histopathological image, which contains a certain cell to be classified near the center. Let $c_i \in \{1, \cdots, C\}$ be the label of patch x_i, where C is the total number of cell categories. The output layer of the spatial constraint component produces an output vector $\hat{y}(x_i) \in \mathbb{R}^C$, and the label c' is predicted as:

$$c_i' = \arg\max_j \hat{y}_j(x_i), \tag{9}$$

where $\hat{y}_j(x_i)$ represents the j-th element in $\hat{y}$. We train the C3DC model by using a log loss function:

$$\ell\left(c_i, c_i'\right)) = -\log\left(p_j\left(\hat{y}(x_i)\right)\right), \tag{10}$$

where $p_j(\hat{y}(x_i))$ is a Softmax activation function:

$$p_j(\hat{y}(x_i)) = \frac{\exp\left(\hat{y}(x_i)\right)}{\sum_{x_i\in X}\exp\left(\hat{y}(x_i)\right)}. \tag{11}$$

In addition, we use a neighbor integrated predictor to predict the class label of each cancer cell. For each patch $x_i \in X$, $z(x_i) \in \Omega$ is expressed as an area centered on x_i. Therefore, we define the adjacent blocks of X as:

$$B(X) = \left\{x_k \in X : \|z(x_k) - z(x_i)\|_2 \le d_\beta\right\}. \tag{12}$$

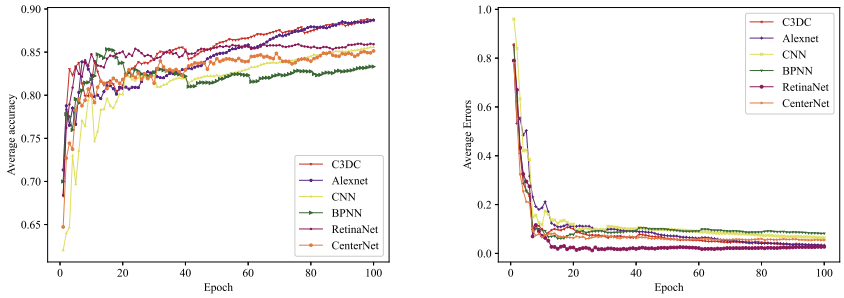
Then, all patches in the circle with radius d_β centered on $z(x_i)$ belong to the adjacent blocks of x_i. For the input patch x_i, according to the output $\hat{y}(x_i)$ of the C3DC model, the final classification $\hat{c}_i$ of x_i can be updated for:

$$\hat{c}_i = \arg\max_j \sum_{x_i\in B(X)} w\left(z(x_i)\right) \times p_j\left(\hat{y}(x_i)\right), \tag{13}$$

where $w(\cdot) : \Omega \to \mathbb{R}$ is a function that assigns different weights to patches according to the centrality $z(x_i)$ in x_i. In this way, a personalized DL model for the detection and classification of colon cancer cells is established.

4.2.3 Performance Evaluation of C3DC Model

To evaluate the performance of the constructed C3DC model, we conduct experiments by comparing it with the CNN [13], BPNN [14], RetinaNet [18], CenterNet [6], and Alexnet [1] models. All experiments are performed on public colon cancer histopathological images from COSMIC [2]. 65,400 images are collected, of which 52,320 images are used as the training set and 13,080 images are used as the test set. In addition, we use 5-fold cross-validation to assess



(a) Average accuracy (b) Average errors

Fig. 12 Performance of DL models for colon cancer cell detection and classification.

Table 5 Performance comparison of comparison models.

Data sets	C3DC	Alexnet	CNN	BPNN	RetinaNet	CenterNet
Training set	87.68%	87.86%	84.61%	82.32%	85.93%	84.17%
Validation set	82.73%	80.01%	75.79%	80.74%	82.06%	79.21%

the performance and versatility of the comparison models. The experimental results are shown in Fig. 12 and Table 5.

As shown in Figs. 12 and Table 5, after 30 training iterations, the accuracy of the proposed C3DC model exceeds 80.00%, which is superior to the comparison models. In addition, as the number of training iterations increases, the accuracy of C3DC continues to improve and converges. The proposed C3DC model achieves a high accuracy of 87.68% on the training set and 82.73% on the validation set. On the contrary, because CNN and BPNN models cannot identify densely distributed cancer cells in the cell detection process, the accuracy of subsequent cell classification is low. The CNN model achieves a low accuracy of 84.61% on the training set and 75.79% on the validation set. Fig. 12(b) shows the error convergence diagram of the comparison models during model training. As the training iteration progresses, the error values of all comparison models can be quickly reduced and converged, indicating that these models are highly efficient.

5 Conclusions

This paper proposed a configurable MIA DL framework to satisfy diverse MIA requirements and improve the versatility of the DL models. The MIA task element library and DL model component library were constructed, respectively. On this basis, we can define new specific MIA requirements for different diseases, and create corresponding DL models with specific network structures by configuring different model components. The actual experimen-

tal results on the medical image data sets show that the proposed configurable DL framework can construct multiple relatively high-performance DL models for different MIA tasks. The constructed DL models can efficiently perform diversified and complex MIA tasks for various diseases. This work will promotes the intersection of MIA and DL research and development, and the gradual maturity of the smart medical industry based on big data.

As future work, we will further expand the model component library to meet more practical MIA demands and provide the public with an online configurable DL platform.

Acknowledgment

This work is partially funded by the National Natural Science Foundation of China under Grant 62002110, and the International Postdoctoral Exchange Fellowship Program under Grant 20180024.

Conflict of interest The authors declared that they have no conflicts of interest to this work.

References

1. Akmal, T.A.M.T.Z., Than, J.C.M., Abdullah, H., Noor, N.M.: Chest x-ray image classification on common thorax diseases using glcm and alexnet deep features. International Journal of Integrated Engineering **11**(4) (2019)
2. Cancer, C.O.S.M.I.: Cosmic. Website (2020). `https://cancer.sanger.ac.uk/cosmic`
3. Chakradhar, S., Sankaradas, M., Jakkula, V., Cadambi, S.: A dynamically configurable coprocessor for convolutional neural networks. In: Annual International Symposium on Computer Architecture (ISCA), pp. 247–257. IEEE (2010)
4. Chen, Z., Xu, Y., Chen, E., Yang, T.: Sadagrad: Strongly adaptive stochastic gradient methods. In: International Conference on Machine Learning (ICML), pp. 913–921 (2018)
5. Ciresan, D., Giusti, A., Gambardella, L.M., Schmidhuber, J.: Deep neural networks segment neuronal membranes in electron microscopy images. In: Advances in Neural Information Processing Systems, pp. 2843–2851 (2012)
6. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q.: Centernet: Keypoint triplets for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6569–6578 (2019)
7. Gao, F., Wu, T., Chu, X., Yoon, H., Xu, Y., Patel, B.: Deep residual inception encoder–decoder network for medical imaging synthesis. IEEE J. Biomed. Health Inform. **24**(1), 39–49 (2019)
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)

9. Guo, X., Liu, X., Zhu, E., Yin, J.: Deep clustering with convolutional autoencoders. In: International Conference on Neural Information Processing, pp. 373–382. Springer (2017)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. IEEE (2016)
11. He, S., Wu, Q., Saunders, J.: A group search optimizer for neural network training. In: International Conference on Computational Science and its Applications, pp. 934–943. Springer (2006)
12. Huang, B., Tian, J., Zhang, H., Luo, Z., Qin, J., Huang, C., He, X., Luo, Y., Zhou, Y., Dan, G., et al.: Deep semantic segmentation feature-based radiomics for the classification tasks in medical image analysis. IEEE J. Biomed. Health Inform. (2020)
13. Kawahara, J., Brown, C.J., Miller, S.P., Booth, B.G., Chau, V., Grunau, R.E.: Brainnetcnn: Convolutional neural networks for brain networks; towards predicting neurodevelopment. NeuroImage **146**, 1038–1049 (2017)
14. Khalifa, I.A., Zeebaree, S.R., Ataş, M., Khalifa, F.M.: Image steganalysis in frequency domain using co-occurrence matrix and bpnn. Science Journal of University of Zakho **7**(1), 27–32 (2019)
15. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks. In: International Conference on 3D Vision (3DV), pp. 239–248. IEEE (2016)
16. Li, X., Dou, Q., Chen, H., Fu, C.W., Qi, X., Belavỳ, D.L.: 3d multi-scale fcn with random modality voxel dropout learning for intervertebral disc localization and segmentation from multi-modality mr images. Med. Image Anal. **45**, 41–54 (2018)
17. Liao, F., Liang, M., Li, Z., Hu, X., Song, S.: Evaluate the malignancy of pulmonary nodules using the 3-d deep leaky noisy-or network. IEEE Trans. Neural Netw. Learn. Syst. **30**(11), 3484–3495 (2019)
18. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision, pp. 2980–2988 (2017)
19. Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., Alsaadi, F.E.: A survey of deep neural network architectures and their applications. Neurocomputing **234**, 11–26 (2017)
20. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440. IEEE (2015)
21. Ma, L., Lu, G., Wang, D., Qin, X., Chen, Z.G., Fei, B.: Adaptive deep learning for head and neck cancer detection using hyperspectral imaging. Visual Computing for Industry, Biomedicine, and Art **2**(1), 1–12 (2019)
22. O'Donoghue, J., Roantree, M., Van Boxtel, M.: A configurable deep network for high-dimensional clinical trial data. In: International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2015)
23. PyTorch: Pytorch. Website (2021). `https://www.pytorch.org`

24. Qian, J., Yang, J., Xu, Y., Xie, J., Lai, Z., Zhang, B.: Image decomposition based matrix regression with applications to robust face recognition. Pattern Recognit. **102**, 107204 (2020)
25. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788. IEEE (2016)
26. Sharma, S.: Activation functions in neural networks. Towards Data Science **6** (2017)
27. Sinha, A., Dolz, J.: Multi-scale self-guided attention for medical image segmentation. IEEE J. Biomed. Health Inform. (2020)
28. Sun, J., Wang, J., Gui, G.: Adaptive deep learning aided digital predistorter considering dynamic envelope. IEEE Trans. Veh. Technol. **69**(4), 4487–4491 (2020)
29. Sun, L., Wang, J., Huang, Y., Ding, X., Greenspan, H., Paisley, J.: An adversarial learning approach to medical image synthesis for lesion detection. IEEE J. Biomed. Health Inform. (2020)
30. Tajbakhsh, N., Jeyaseelan, L., Li, Q., Chiang, J.N., Wu, Z., Ding, X.: Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. Med. Image Anal. p. 101693 (2020)
31. Tang, S., Shen, C., Wang, D., Li, S., Huang, W., Zhu, Z.: Adaptive deep feature learning network with nesterov momentum and its application to rotating machinery fault diagnosis. Neurocomputing **305**, 1–14 (2018)
32. Wang, Z., Zineddin, B., Liang, J., Zeng, N., Li, Y., Du, M., Cao, J., Liu, X.: A novel neural network approach to cdna microarray image segmentation. Computer methods and programs in biomedicine **111**(1), 189–198 (2013)
33. Xiong, Z., Fedorov, V.V., Fu, X., Cheng, E., Macleod, R., Zhao, J.: Fully automatic left atrium segmentation from late gadolinium enhanced magnetic resonance imaging using a dual fully convolutional neural network. IEEE Trans. Med. Imaging **38**(2), 515–524 (2018)
34. Yuan, Y., Fang, J., Lu, X., Feng, Y.: Spatial structure preserving feature pyramid network for semantic image segmentation. ACM Trans. Multimed. Comput. Commun. Appl. **15**(3), 1–19 (2019)
35. Zakaria, Z., Suandi, S.A., Mohamad-Saleh, J.: Hierarchical skin-adaboost-neural network (h-skann) for multi-face detection. Appl. Soft. Comput. **68**, 172–190 (2018)
36. Zeng, N., Li, H., Wang, Z., Liu, W., Liu, S., Alsaadi, F.E., Liu, X.: Deep-reinforcement-learning-based images segmentation for quantitative analysis of gold immunochromatographic strip. Neurocomputing **425**, 173–180 (2021)
37. Zhang, J., Liu, M., Shen, D.: Detecting anatomical landmarks from limited medical imaging data using two-stage task-oriented deep neural networks. IEEE Trans. Image Process. **26**(10), 4753–4764 (2017)
38. Zhao, H., Gallo, O., Frosio, I., Kautz, J.: Loss functions for image restoration with neural networks. IEEE Trans. Comput. Imaging **3**(1), 47–57 (2016)

39. Zhao, Z., Wang, Z., Zou, L., Liu, H.: Finite-horizon h_∞ state estimation for artificial neural networks with component-based distributed delays and stochastic protocol. Neurocomputing **321**, 169–177 (2018)
40. Zheng, G., Liu, X., Han, G.: A review of computer-aided detection and diagnosis system for medical imaging. Journal of Software **29**(5), 1471–1514 (2018)