Research Article

A Novel Radial Basis Neural Network-Leveraged Fast Training Method for Identifying Organs in MR Images

Min Xu ⬡,[1,2,3] **Pengjiang Qian** ⬡,[3,4] **Jiamin Zheng,**[4] **Hongwei Ge** ⬡,[2] **and Raymond F. Muzic Jr.**[5]

[1]*School of Internet of Things Technology, Wuxi Institute of Technology, Wuxi 214121, China*
[2]*School of Internet of Things, Jiangnan University, Wuxi 214122, China*
[3]*Jiangsu Key Lab of Media Design and Software Technology, Jiangnan University, Wuxi 214122, China*
[4]*School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China*
[5]*Department of Radiology and Case Center for Imaging Research, University Hospitals, Case Western Reserve University, Cleveland, OH 44106, USA*

Correspondence should be addressed to Pengjiang Qian; qianpjiang@jiangnan.edu.cn

We propose a new method for fast organ classification and segmentation of abdominal magnetic resonance (MR) images. Magnetic resonance imaging (MRI) is a new type of high-tech imaging examination fashion in recent years. Recognition of specific target areas (organs) based on MR images is one of the key issues in computer-aided diagnosis of medical images. Artificial neural network technology has made significant progress in image processing based on the multimodal MR attributes of each pixel in MR images. However, with the generation of large-scale data, there are few studies on the rapid processing of large-scale MRI data. To address this deficiency, we present a fast radial basis function artificial neural network (Fast-RBF) algorithm. The importance of our efforts is as follows: (1) The proposed algorithm achieves fast processing of large-scale image data by introducing the ε-insensitive loss function, the structural risk term, and the core-set principle. We apply this algorithm to the identification of specific target areas in MR images. (2) For each abdominal MRI case, we use four MR sequences (fat, water, in-phase (IP), and opposed-phase (OP)) and the position coordinates (x, y) of each pixel as the input of the algorithm. We use three classifiers to identify the liver and kidneys in the MR images. Experiments show that the proposed method achieves a higher precision in the recognition of specific regions of medical images and has better adaptability in the case of large-scale datasets than the traditional RBF algorithm.

1. Introduction

Magnetic resonance imaging (MRI) is a new type of high-tech imaging examination fashion in recent years. It has the advantages of no ionizing radiation, no bone artifacts, and multidirectional and multiparameter imaging [1]. Therefore, the generation of an end-to-end intelligent disease diagnosis system based on MRI is an inevitable direction for the development of intelligent medicine. To achieve the goal of effective intelligent medical treatment, this paper studies the classification of abdominal organs based on MRI.

There are many techniques for medical image processing [2–5]. Gordillo et al. [6] divided the existing MR image pro-

cessing technologies into the following three categories: The first type is threshold-based methods, which classify the segmentation objects (such as pixels) of the MR image by comparing them with different thresholds [7–9]. The second type is region-based methods, which divide several mutually exclusive regions according to preset rules and then categorize pixels with the same attributes into the same region [10, 11]. The third type is pixel-based classification methods, which mainly classify the objects according to the MR multimodal attributes of each pixel. According to whether the training set is labeled or not, they can be subdivided into unsupervised, semisupervised, and supervised methods [12–14].

Of the third type of methods, an artificial neural network as a supervised learning model is applied to the field of medical imaging [15]. It is suitable for image processing without prior distribution assumptions; its application can be divided into three categories: The first type is to apply artificial neural networks directly to MR image processing. Lucht et al. [16] applied a neural network to the dynamic segmentation of MR breast images. Egmont-Petersen et al. [17] used neural networks and multiscale pharmacokinetic features to segment bone tumors in MR perfusion images. Zhang et al. [18] proposed a visual encoding model based on deep neural networks and transfer learning for brain activity measured by functional magnetic resonance imaging. The second type is to use a convolutional neural network or its improved algorithm to segment MR images [19–22]. Khalilia et al. [20] used convolutional neural networks to automatically perform brain tissue segmentation in fetal MRI. Wang et al. [21] used dynamic pixelwise weighting-based fully convolutional neural networks for left ventricle segmentation in short-axis MRI. The third type is to use hybrid neural networks to segment MR images. Glass et al. [23] used a hybrid artificial neural network to segment the inversion recovery image of a normal human brain. Alejo et al. [24] used a hybrid artificial neural network to design an accurate computer-aided method capable of performing region segmentation. Reddick et al. [25] used a hybrid neural network to propose a fully automatic method for segmentation and classification of multispectral MR images.

Based on the review of the above literature, great progress has been made in the use of artificial neural networks for medical image segmentation. However, with the higher resolution requirements of MR images and the increasing size of the dataset, research on fast artificial neural network training for large medical image datasets is still lacking. In response to this phenomenon, this paper proposes the Fast-RBF algorithm, which has fast processing capabilities for large datasets. We applied this method to MRI-based abdominal organ classification and segmentation. The results showed that this method achieved significant results. The main contributions of this paper are as follows:

(1) The Fast-RBF algorithm with a large-sample-processing capability is proposed by introducing the ε-insensitive loss function and structural risk term and using the core-set principle [26]. This method not only retains the strong nonlinear fitting ability and simple learning rules of RBF artificial neural networks but can also process a large dataset quickly, which improves the processing speed and efficiency.

(2) For each abdominal MRI case, we use four MR sequences (fat, water, IP, and OP) and the position coordinates (x, y) of each pixel as the input of the algorithm. We use three classifiers to identify the liver, kidneys, and other tissues. The proposed algorithm has better adaptability and runs faster in large dataset scenarios than the traditional RBF neural network algorithm.

The remainder of this paper is divided into four parts: Section 2 introduces RBF neural networks and the relation-
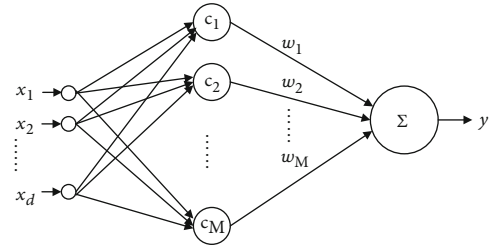

Figure 1: The model of an RBF neural network.

ship between RBF neural networks and linear models; Section 3 introduces the Fast-RBF neural network with its large-sample-processing ability; Section 4 verifies the validity of the proposed algorithm on medical image processing; and Section 5 summarizes the full text.

2. Related Work

2.1. RBF Neural Network.
RBF neural networks consist of an input layer, an implicit layer, and an output layer, as shown in Figure 1. Among them, $\mathbf{x}_i \in R^d, y \in R$, the number of hidden layer nodes is M, and the nonlinear mapping $f : R^d \longrightarrow R$ is performed by the RBF neural network.

In an RBF neural network, the input layer receives the training samples; the hidden layer node performs a nonlinear transformation through the radial basis function that maps the input space to a new space. If the radial basis function is defined as a Gaussian function, let $\mathbf{c}_i \in R^d$ denote the center of the Gaussian function and let δ_i represent the kernel width of the Gaussian function. This function can be expressed as

$$\varphi(\|\mathbf{x} - \mathbf{c}_i\|) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{\delta_i}\right). \tag{1}$$

The nodes of the output layer implement a linear weighted combination in this new space. Let w_i be the connection weight of the hidden layer and the output layer and $\varphi(\bullet)$ be the radial basis function; then, the mapping function of $R^d \longrightarrow R$ is

$$y = f(\mathbf{x}) = \sum_{i=1}^{M} w_i \varphi(\|\mathbf{x} - \mathbf{c}_i\|). \tag{2}$$

2.2. RBF Neural Network and Linear Model.
According to the introduction above, the RBF neural network has 3 parameters: the center vector of the radial basis function $\mathbf{c}_i = [c_{i1}, c_{i2}, \cdots c_{id}]^{\mathrm{T}}$, the kernel width δ_i, and the weight of the output layer w_i. Among them, $\mathbf{c}_i$ and δ_i can be determined by the fuzzy C-means (FCM) clustering algorithm [27], and w_i is obtained by the gradient descent learning algorithm. Let μ_{ji}, which is obtained by the FCM clustering algorithm, denote the fuzzy membership of sample $\mathbf{x}_j$ for the ith class, n represent the size of the training sample, and M indicate the number of hidden layer nodes; then, the center of the radial basis function $\mathbf{c}_{ik}$ and the kernel width δ_i can be expressed by equations (3) and (4):

$$c_{ik} = \frac{\sum_{j=1}^{n} \mu_{ji} x_{jk}}{\sum_{j=1}^{n} \mu_{ji}}, \tag{3}$$

$$\delta_i = \frac{\sum_{j=1}^{n} \mu_{ji} \|\mathbf{x}_j - \mathbf{c}_i\|^2}{\sum_{j=1}^{n} \mu_{ji}}. \tag{4}$$

Let $\tilde{x}_i = \varphi(\|\mathbf{x} - \mathbf{c}_i\|)$, $i = 1, 2, \cdots, M$.

$$\tilde{\mathbf{x}} = [\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_M]^{\mathrm{T}}. \tag{5}$$

The center $\mathbf{c}_i$ and the kernel width δ_i of the radial basis function are obtained by equations (3) and (4), the input sample is mapped to the new space $f : R^d \longrightarrow R^M$, and the conversion from the input layer to the hidden layer is a nonlinear mapping.

Let $\mathbf{p} = [w_1, w_2, \cdots w_M]^{\mathrm{T}}$; then, the neural network function can be expressed as

$$y = \mathbf{p}^{\mathrm{T}} \tilde{\mathbf{x}}. \tag{6}$$

It can be seen from equation (6) that when the radial basis function hidden layer is estimated, the output of the network can be converted into a linear model.

3. Fast-RBF Algorithm

3.1. Fast-RBF Principle. First, the ε-insensitive loss function corresponding to the RBF linear model is introduced. To minimize the value of the ε-insensitive loss function, ε is solved as the constraint term of the optimization problem. Then, the structural risk term and the Gaussian kernel are introduced to construct the RBF neural network optimization model with large-sample processing. The specific steps are as follows.

Step 1. From equations (3) and (4), the values of $\mathbf{c}_i$ and δ_i are obtained; then, from equation (5), the model input $\tilde{\mathbf{x}}$ is obtained.

Step 2. Introducing the ε-insensitive loss function.

First, the definition of the ε-insensitive loss function is given:

Definition 1. The ε-insensitive loss function is defined as [28]

$$L^\varepsilon(\mathbf{x}, y, f) = |y - f(\mathbf{x})|_\varepsilon = \max\left(0, |y - f(\mathbf{x})|_\varepsilon\right), \tag{7}$$

where $\mathbf{x} \in R^d, y \in R$.

For the linear model of equation (6), its corresponding ε-insensitive loss function can be expressed as

$$\sum_{i=1}^{n} |y_i^o - y_i|_\varepsilon = \sum_{i=1}^{n} \max(0, |y_i^o - y_i| - \varepsilon) = \sum_{i=1}^{n} \max\left(0, |\mathbf{p}^{\mathrm{T}} \tilde{\mathbf{x}}_i - y_i| - \varepsilon\right), \tag{8}$$

where y_i^o represents the neural network output value and y_i represents the real output value.

For equation (8), the constraints of $\mathbf{p}^{\mathrm{T}} \tilde{\mathbf{x}}_i - y_i < \varepsilon$ and $y_i - \mathbf{p}^{\mathrm{T}} \tilde{\mathbf{x}}_i < \varepsilon$ are not always satisfied, so the relaxation factors ξ_i and ξ_i^* are introduced, and the following constraints can be obtained:

$$\begin{cases} y_i - \mathbf{p}^{\mathrm{T}} \tilde{\mathbf{x}}_i < \varepsilon + \xi_i, & \xi_i \geq 0, \\ \mathbf{p}^{\mathrm{T}} \tilde{\mathbf{x}}_i - y_i < \varepsilon + \xi_i^*, \xi_i^* \geq 0. \end{cases} \tag{9}$$

The purpose of this algorithm is to minimize the value of the ε-insensitive loss function represented by equation (8). The value of the ε-insensitive parameter will directly affect the accuracy of the modeling. Therefore, the parameter λ is introduced, and ε is used as the constraint term in the optimization problem. Combined with equation (9), the optimization problem can be expressed equivalently as

$$\min 2\lambda\varepsilon + \frac{\lambda}{\mu n} \sum_{i=1}^{n} \left(\xi_i^2 + \xi_i^{*2}\right), \quad \text{s.t.} \begin{cases} y_i - \mathbf{p}^{\mathrm{T}} \tilde{\mathbf{x}}_i < \varepsilon + \xi_i, \\ \mathbf{p}^{\mathrm{T}} \tilde{\mathbf{x}}_i - y_i < \varepsilon + \xi_i^*, \end{cases} \tag{10}$$

where the parameter μ is the balance factor and $\xi_i, \xi_i^* \geq 0$ is automatically satisfied.

Step 3. Introducing structural risk items and kernel functions.

A support vector machine is an implementation of the principle of structural risk minimization [28]; the method proposed in this paper learns the implementation method of the support vector machine and introduces a regularization term to minimize the risk in the algorithm structure. The kernel method is an important component of a support vector machine [28], which is used to improve the computational ability of the linear learner. The method proposed in this paper also introduces a kernel function. After introducing the regular term and the kernel function, the optimization problem can be expressed by

$$\min_{\mathbf{p}, \varepsilon, \xi_i, \xi_i^*} \|\mathbf{p}\|^2 + 2\lambda\varepsilon + \frac{\lambda}{\mu n} \sum_{i=1}^{n} \left(\xi_i^2 + \xi_i^{*2}\right), \quad \text{s.t.} \begin{cases} y_i - \mathbf{p}^{\mathrm{T}} \varphi(\tilde{\mathbf{x}}_i) < \varepsilon + \xi_i, \\ \mathbf{p}^{\mathrm{T}} \varphi(\tilde{\mathbf{x}}_i) - y_i < \varepsilon + \xi_i^*, \\ i = 1, 2, \cdots, n. \end{cases} \tag{11}$$

Step 4. Formula derivation.

By introducing the Lagrange multiplier, the Lagrangian function of equation (11) can be expressed as

$$\begin{aligned} L = &\|\mathbf{p}\|^2 + 2\lambda\varepsilon + \frac{\lambda}{\mu n} \sum_{i=1}^{n} \left(\xi_i^2 + \xi_i^{*2}\right) \\ &+ \sum_{i=1}^{n} \alpha_i \left(y_i - \mathbf{p}^{\mathrm{T}} \varphi(\tilde{\mathbf{x}}_i) - \varepsilon - \xi_i\right) \\ &+ \sum_{i=1}^{n} \alpha_i^* \left(\mathbf{p}^{\mathrm{T}} \varphi(\tilde{\mathbf{x}}_i) - y_i - \varepsilon - \xi_i^*\right). \end{aligned} \tag{12}$$

The matrix form of the corresponding dual problem of equation (12) is

$$\left\{ \max \begin{bmatrix} \alpha^T & \alpha^{*T} \end{bmatrix} \begin{bmatrix} \frac{2}{\lambda}y \\ -\frac{2}{\lambda}y \end{bmatrix} - \begin{bmatrix} \alpha^T & \alpha^{*T} \end{bmatrix} \tilde{\mathbf{K}} \begin{bmatrix} \alpha \\ \alpha^* \end{bmatrix}, \quad \text{s.t.} \begin{bmatrix} \alpha^T & \alpha^{*T} \end{bmatrix} \mathbf{1} = 1, \quad \alpha, \alpha^* \geq 0, \right. \tag{13}$$

where α, α^* are the Lagrange coefficients and $\tilde{K}$ is the kernel function. They are

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix},$$

$$\alpha = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix},$$

$$\alpha^* = \begin{bmatrix} \alpha_1^* \\ \vdots \\ \alpha_n^* \end{bmatrix}, \tag{14}$$

$$\tilde{\mathbf{K}} = \begin{bmatrix} \tilde{k}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j) \end{bmatrix} = \begin{bmatrix} \mathbf{K} + \dfrac{\mu n}{\lambda}\mathbf{I} & -\mathbf{K} \\ -\mathbf{K} & \mathbf{K} + \dfrac{\mu n}{\lambda}\mathbf{I} \end{bmatrix},$$

where K is the Gaussian kernel function.

The values of the variables obtained by the solution are

$$\left\{ \begin{aligned} & \mathbf{p} = \lambda \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\varphi(\tilde{\mathbf{x}}_i), \\ & \xi_i = \alpha_i \mu n, \\ & \xi_i^* = \alpha_i^* \mu n. \end{aligned} \right. \tag{15}$$

In addition, because $\sum_{i=1}^{n}(\alpha_i + \alpha_i^*) = 1, \mu = \sum_{i=1}^{n}(\xi_i + \xi_i^*)/n$.

Step 5. Prediction.

The prediction function is shown in the following equation:

$$\begin{aligned} y &= \mathbf{p}^T \varphi(\tilde{\mathbf{x}}_{\text{test}}) = \lambda \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\varphi^T(\tilde{\mathbf{x}}_i)\varphi(\tilde{\mathbf{x}}_{\text{test}}) \\ &= \lambda \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\tilde{K}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_{\text{test}}). \end{aligned} \tag{16}$$

If it is used for classification,

$$y = \text{sign}\left(\mathbf{p}^T \varphi(\tilde{\mathbf{x}}_{\text{test}})\right). \tag{17}$$

If $y > 0$, it belongs to the positive class, and if $y < 0$, it belongs to the negative class.

It can be seen from this section that the algorithm proposed in this paper is a quadratic programming problem.

3.2. The Center-Constrained MEB Problem. In 2002, Bădoiu and Clarkson proposed a minimum enclosing ball (MEB) $(1 + \xi)$-approximation algorithm based on the core set in the literature [26]. The algorithm uses a subset of the input set, which is called the core set, to solve the optimization problem. The algorithm can obtain the same good approximation results as the original input set to improve the efficiency of the algorithm. Tsang et al. [29] suggested that the MEB problem is related to many kernel problems. Eligible quadratic programming (QP) problems can be solved quickly by the core-set algorithm. The following section briefly introduces the center-constrained minimum enclosing ball (CC-MEB) algorithm. Next, we will introduce the relationship between the proposed algorithm and CC-MEB and implement the fast algorithm proposed in this paper.

Given the training sample $S = \{\varphi(\mathbf{x}_i)\}_{i=1}^{m}$, where $\mathbf{x}_i \in R^d$ and φ is the kernel mapping associated with a given kernel $\mathbf{K}$, adding one dimension δ_i to each $\varphi(\mathbf{x}_i)$ forms a set $S = \{(\varphi^T(\mathbf{x}_i), \delta_i)\}_{i=1}^{m}$. By setting the coordinate of the last one-dimensional center point to be 0, that is, the CC-MEB's coordinate is $[\mathbf{c}, 0]$, then the optimization problem of finding the smallest CC-MEB that contains all the samples in the set S is

$$\left\{ \min_{\mathbf{c}, R} R^2, \quad \text{s.t.}(\varphi(\mathbf{x}_i) - \mathbf{c})^2 + \delta_i^2 \leq R^2, \right. \tag{18}$$

where $i = 1, 2, \cdots m$.

Let $\mathbf{\Delta} = [\delta_1^2, \cdots, \delta_m^2]^T \geq 0$; then, the matrix form of the corresponding dual problem of equation (18) is

$$\left\{ \max_{\boldsymbol{\beta}} \boldsymbol{\beta}^T (\text{diag}(\mathbf{K}) + \mathbf{\Delta}) - \boldsymbol{\beta}^T \mathbf{K} \boldsymbol{\beta}, \quad \text{s.t.} \boldsymbol{\beta} \geq 0, \boldsymbol{\beta}^T \mathbf{1} = 1, \right. \tag{19}$$

where the kernel matrix is

$$\mathbf{K}_{m \times m} = \begin{bmatrix} k(\mathbf{x}_i, \mathbf{x}_j) \end{bmatrix} = \begin{bmatrix} \varphi^T(\mathbf{x}_i)\varphi(\mathbf{x}_j) \end{bmatrix}. \tag{20}$$

Using the optimal solution $\boldsymbol{\beta}$ to obtain the values for the radius R and center point $\mathbf{c}$,

$$\left\{ \begin{aligned} & R = \sqrt{\boldsymbol{\beta}^T (\text{diag}(\mathbf{K}) + \mathbf{\Delta}) - \boldsymbol{\beta}^T \mathbf{K} \boldsymbol{\beta}} \\ & \mathbf{c} = \sum_{i=1}^{m} \beta_i \varphi(\mathbf{x}_i). \end{aligned} \right. \tag{21}$$

The formula for the distance from any point to the center point is

$$\|\mathbf{c} - \varphi(\mathbf{x}_l)\|^2 + \delta_l^2 = \|\mathbf{c}\|^2 - 2(\mathbf{K}\boldsymbol{\beta})_l + k_{ll} + \delta_l^2. \tag{22}$$

Because $\boldsymbol{\beta}^T \mathbf{1} = 1$, any real number η is added to equation

---

Input: Dataset$\{Z_i = (\tilde{\mathbf{x}}_i, y_i)\}_{i=1}^n$, approximation parameter ξ, parameter η, parameter λ, parameter μ, and kernel width δ_i, where $\xi = 1e - 6$ and $\eta = \max\left(0, \max\left(\mathrm{diag}\,(\tilde{\mathbf{K}}) - (2/C)\begin{bmatrix} y \\ -y \end{bmatrix}\right)\right)$

Output: Core-set Q, Lagrangian coefficient $\tilde{\alpha}$

Training steps:

Step 1: Randomly select 20 samples to form the initial core set Q_0;

Generate the center $\mathbf{c}_0$ and radius R_0 of the initial CC-MEB according to equation (21) and set the number of iterations t to be 0

Step 2: Randomly select 59 samples and calculate the distance from any sample to the center of the CC-MEB according to equation (22). If there is no sample $\tilde{\mathbf{x}}$ outside $CC\text{-}MEB(\mathbf{c}_t, (1 + \xi)R_t)$, proceed to step 6

Step 3: Find the farthest sample from the center $\mathbf{c}_t$ in step 2 and add the sample to core-set $Q_{t+1} = Q_t \cup \{\tilde{\mathbf{x}}\}$

Step 4: Solve the new CC-MEB, recorded as $\mathrm{MEB}(Q_{t+1})$, and $\mathbf{c}_{t+1} = \mathbf{c}_{\mathrm{MEB}}(Q_{t+1})$, $R_{t+1} = R_{\mathrm{MEB}}(Q_{t+1})$

Step 5: Set $t = t + 1$ and return to step 2

Step 6: Terminate the training and return the required output

Prediction step:

Input the test sample $\tilde{\mathbf{x}}_{\mathrm{test}}$ into the following:

$y\_\mathrm{test} = \mathbf{p}^T \varphi(\tilde{\mathbf{x}}_{\mathrm{test}}) = \lambda \mathbf{c}^T \varphi(\tilde{\mathbf{x}}_{\mathrm{test}})$

ALGORITHM 1: The Fast-RBF algorithm.

(19), which does not affect the value of β. The original dual form is changed to

$$\begin{cases} \max_{\beta} \beta^T (\mathrm{diag}\,(\mathbf{K}) + \Delta - \eta \mathbf{1}) - \beta^T \mathbf{K} \beta, & \text{s.t.}\, \beta \geq 0, \beta^T \mathbf{1} = 1, \Delta \geq 0. \end{cases} \tag{23}$$

Reference [29] pointed out that any QP problem that satisfies equation (23) can be regarded as an CC-MEB problem, which can be solved by the core-set fast algorithm

3.3. Relationship between Fast-RBF and CC-MEB. Equation (13) is the QP form of Fast-RBF. Let $\tilde{\alpha} = \begin{bmatrix} \alpha^T & \alpha^{*T} \end{bmatrix}^T$; then,

$$\Delta = -\,\mathrm{diag}\,(\tilde{\mathbf{K}}) + \eta \mathbf{1} + \frac{2}{\lambda}\begin{bmatrix} y \\ -y \end{bmatrix}, \tag{24}$$

where the real number η should be large enough so that $\Delta \geq 0$. Thus, equation (13) can be written as follows:

$$\begin{cases} \max_{\tilde{\alpha}} & \tilde{\alpha}^T \left(\mathrm{diag}\,(\tilde{\mathbf{K}}) + \Delta - \eta \mathbf{1}\right) - \tilde{\alpha}^T \tilde{\mathbf{K}} \tilde{\alpha} \\ s.t. & \tilde{\alpha}^T \mathbf{1} = 1. \end{cases} \tag{25}$$

This form is equivalent to the CC-MEB problem from equation (23), and the problem can be solved using the core-set fast algorithm [29].

According to formula (25), the center of sphere $\mathbf{c}$ can be calculated as $\mathbf{c} = \sum_{i=1}^{2*n} \tilde{\alpha}_i \tilde{\varphi}(\tilde{\mathbf{x}}_i)$. In the formula, when $i = 1, \cdots, n$, then $\tilde{\varphi}(\tilde{\mathbf{x}}_i) = \varphi(\tilde{\mathbf{x}}_i)$; when $i = n + 1, \cdots, 2n$, then $\tilde{\varphi}(\tilde{\mathbf{x}}_i) = -\varphi(\tilde{\mathbf{x}}_i)$, and the derivation is available:

$$\mathbf{c} = \sum_{i=1}^{2*n} \tilde{\alpha}_i \tilde{\varphi}(\tilde{\mathbf{x}}_i) = \sum_{i=1}^n \alpha_i \varphi(\tilde{\mathbf{x}}_i) + \sum_{i=1}^n \alpha_i^* (-\varphi(\tilde{\mathbf{x}}_i)) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) \varphi(\tilde{\mathbf{x}}_i). \tag{26}$$

Therefore, the value of $\mathbf{p}$ in equation (15) is $\mathbf{p} = \lambda \mathbf{c}$.

3.4. The Implementation of Fast-RBF. Algorithm 1 describes the steps of the Fast-RBF algorithm, and the flow chart is shown in Figure 2.

4. Experimental Results and Analysis

In this paper, the effectiveness of the proposed method is verified by comparing it with the traditional RBF algorithm on MR images. The experiment is divided into two stages: the parameter optimization stage and the modeling stage. In the parameter optimization stage, the grid search method is used to obtain the optimal parameters of each algorithm based on the training set. In the modeling stage, the training set is modeled using optimal parameters, and the test set is used to obtain the performance of each algorithm.

The experiment is verified from the following four aspects:

(1) Verify that the size of the core set of the Fast-RBF algorithm is much smaller than the training set's scale, which can speed up the modeling time of the algorithm

(2) Verify that the prediction capability of the Fast-RBF algorithm is comparable to the prediction capability of the RBF algorithm

(3) Verify that the modeling time of the Fast-RBF algorithm on large datasets is much smaller than that of the RBF algorithm

For the experimental environment, the operating system is Windows 10; the processor is an Intel i5 2.71 GHz CPU; the memory is 8 GB; and the main application software is MATLAB R2015a.

4.1. Experimental Preparation. The use case in this section is from MRI scans of five subjects recruited by the University Hospitals Cleveland Medical Center Institutional Review Board. Before the experiment, a block diagram is first used
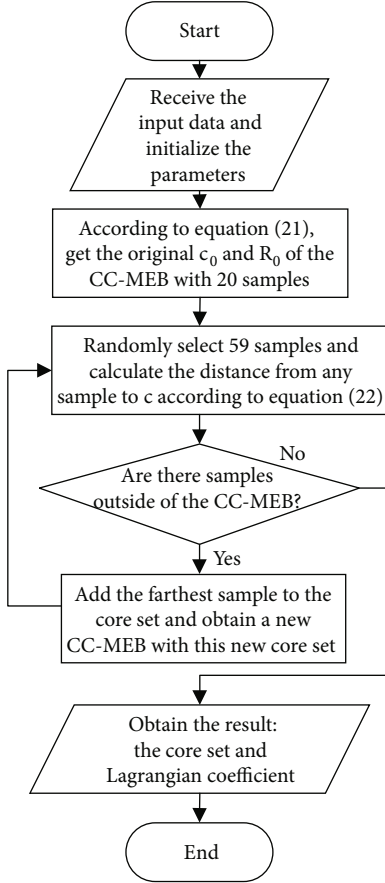
Figure 2: Flow chart of the algorithm.

Figure 3: Areas to be identified.

Table 1: Convolutional kernel $CK_{3\times3}$.

0.1	0.1	0.1
0.1	0.2	0.1
0.1	0.1	0.1

to frame the area to be identified, as shown in Figure 3. Next, we train and test the data of region of interest in abdominal organ map. The experiment is to classify the liver and kidneys of the region of interest in the abdominal organ map.

For each case, we extract the local textural features from four different types of abdominal 3D MR images, namely, fat, water, in-phase (IP), and opposed-phase (OP), as the input of the algorithm. Noise cannot be avoided in these actual data, and this noise will affect the final image recognition effect. Therefore, this paper adopts the method proposed in [30, 31] to design a convolution kernel as shown in Table 1, preprocesses the experimental data, and implements feature extraction.

In addition, we also consider the pixel spacing of the MR images and adopt the grid division strategy. Let (x, y) represent the position information of the pixel. That is, we combine the features that we extracted and obtain a six-dimensional feature.

4.2. Experimental Method. We define the liver of the region of interest in abdominal MR images as class A, the kidneys as class B, and other tissues as class C. Therefore, this is a multi-classification problem. We train "liver (class A)-kidney (class B)," "liver (class A)-other tissue (class C)," and "kidney (class B)-other tissue (class C)" to obtain three classification results; the final result is then determined by voting. The voting method is as follows:
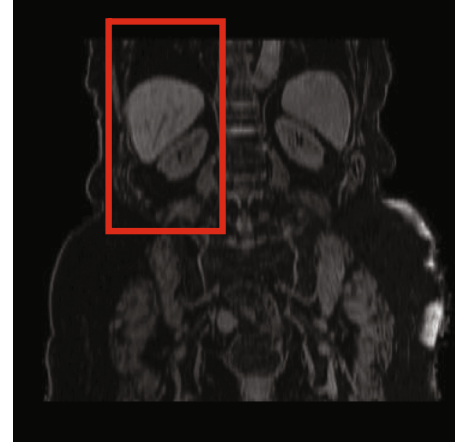
Let $A = B = C = 0$.

The classification is (A, B) if it belongs to A, and $A = A + 1$; otherwise, $B = B + 1$.

The classification is (A, C) if it belongs to A, and $A = A + 1$; otherwise, $C = C + 1$.

The classification is (B, C) if it belongs to B, and $B = B + 1$; otherwise, $C = C + 1$.

The final sample belongs to the class with the largest values of A, B, and C.

The classification accuracy is used to measure the performance of the algorithm.

$$\text{prediction accuracy} = \frac{\text{the number of correctly classified test samples}}{\text{total number of test samples}}.$$

$$(27)$$

4.3. Experimental Results. Cases 1-4 contain a total of 59,904 data points. We randomly selected 10,000 data points, 20,000 data points, 30,000 data points, 40,000 data points, 50,000 data points, and 59,904 data points for training. Case 5, which contains 16,896 data points, was used as the test set. The experiment was repeated 10 times for each training set size to verify the advantages of the proposed method.

4.3.1. Core Set of the Fast-RBF Algorithm. Figure 4 shows the average values of the total number of core-set samples for the three classifiers at different training set sizes. Figure 4 shows that the total number of core sets is between 240 and 300, which is much smaller than the sample size. Replacing all the samples with the core sets in the model construction step will greatly improve the operational efficiency.

4.3.2. Prediction Ability of the Fast-RBF Algorithm. It can be seen from Table 2 that both algorithms can achieve a good generalization performance. However, with the increase in
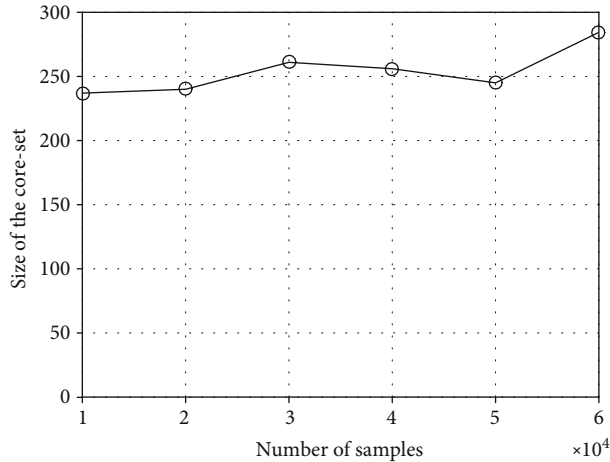
FIGURE 4: The size of the core set at different sample sizes.

TABLE 2: Prediction accuracy and standard deviation of the two algorithms at different dataset sizes.

Size of the dataset	Prediction accuracy and standard deviation	
	RBF	Fast-BRF
10,000	0.9345 ± 0.0083	0.9458 ± 0.0153
20,000	0.9389 ± 0.0063	0.9551 ± 0.0132
30,000	0.9381 ± 0.0104	0.9496 ± 0.0098
40,000	—	0.9467 ± 0.0111
50,000	—	0.9432 ± 0.0112
59,904	—	0.9552 ± 0.0066

the amount of training data, the RBF algorithm requires more samples to participate in the modeling step, so it is more constrained. When the data size exceeds 30,000 data points, it can no longer be solved. The Fast-RBF algorithm uses core-set technology to solve the problem. Key samples are added to the core set one by one, and the average size of the core set does not exceed 300, so it can process a larger dataset and can achieve a generalization ability comparable to that of the RBF algorithm. The organ classification results are shown in Figure 5.

4.3.3. Time Performance of the Fast-RBF Method. Table 3 shows that the modeling time required by the two algorithms has a stable growth with increasing sample size. When the size of the dataset is 30,000 data points, the average modeling time of the RBF algorithm is 7,580 seconds, while the average time of the Fast-RBF algorithm is 15.2609 seconds. The modeling time of the Fast-RBF algorithm is much smaller than that of the RBF algorithm. In addition, when the size of the dataset is more than 30,000 data points, the RBF algorithm will not run.

4.4. Experimental Conclusion. It is known from experiments that the Fast-RBF algorithm can be used for organ recognition in MR images. The advantage of the proposed algorithm is that it requires far less modeling time than the RBF algo-
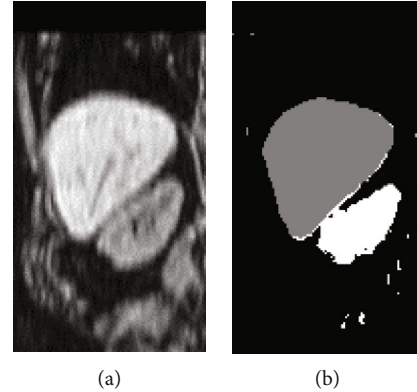

(a)   (b)

FIGURE 5: Test results. (a) Original picture. (b) Organ classification results.

TABLE 3: Average modeling time and standard deviation of the two algorithms under different data sizes.

Size of the dataset	Modeling time and standard deviation of each method (s)	
	RBF	Fast-BRF
10,000	210.5813 ± 3.5134	10.1719 ± 7.0177
20,000	737.1344 ± 7.1357	12.8016 ± 4.0126
30,000	$7.58E + 03 \pm 164.0596$	15.2609 ± 8.2559
40,000	—	16.5953 ± 9.1518
50,000	—	17.5953 ± 9.1983
59,904	—	22.9781 ± 7.0587

rithm in large datasets under the premise of ensuring the prediction accuracy. The algorithm has strong practicability.

5. Conclusion

Our studies are based on MRI of challenging body sections of the abdomen. We proposed the Fast-RBF algorithm, which is suitable for the rapid training of a large dataset. By introducing the ε-insensitive loss function, learning the structural risk term and kernel method of the support vector machine, and using the core-set principle, the proposed algorithm can meet the needs of large sample sizes. This method can quickly process large datasets and is suitable for medical image processing.

The method proposed in this paper is a supervised learning method. The training samples need to be labeled, and the workload of data preparation is large. In the future, we will further study the semisupervised abdominal image recognition method in which only a small number of class labels are needed to achieve image processing.

Data Availability

Data sharing is not available for our study, as the experimental data were afforded by our collaboration partners at the University Hospitals Cleveland Medical Center, OH, USA. Without permission, we cannot share any of our data with others.

Additional Points

Consent

Informed consent was obtained from all the individual participants included in this study.

Disclosure

Conflicts of Interest

We declare that there are no conflicts of interest.

Acknowledgments

References

[1] N. Upadhyay and A. D. Waldman, "Conventional MRI evaluation of gliomas," *The British Journal of Radiology*, vol. 84, pp. S107–S111, 2011.

[2] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 12, pp. 7405–7415, 2016.

[3] H. Zhang, Z. Gao, L. Xu et al., "A meshfree representation for cardiac medical image computing," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 6, pp. 1–12, 2018.

[4] D. Zhang, D. Meng, and J. Han, "Co-saliency detection via a self-paced multiple-instance learning framework," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 5, pp. 865–878, 2017.

[5] M. Li, S. Dong, Z. Gao et al., "Unified model for interpreting multi-view echocardiographic sequences without temporal information," *Applied Soft Computing*, vol. 88, article 106049, 2020.

[6] N. Gordillo, E. Montseny, and P. Sobrevilla, "State of the art survey on MRI brain tumor segmentation," *Magnetic Resonance Imaging*, vol. 31, no. 8, pp. 1426–1438, 2013.

[7] Y. J. Zhang and J. J. Gerbrands, "Transition region determination based thresholding," *Pattern Recognition Letters*, vol. 12, no. 1, pp. 13–23, 1991.

[8] C. Yan, N. Sang, and T. Zhang, "Local entropy-based transition region extraction and thresholding," *Pattern Recognition Letters*, vol. 24, no. 16, pp. 2935–2941, 2003.

[9] P. Gibbs, D. L. Buckley, S. J. Blackband, and A. Horsman, "Tumour volume determination from MR images by morphological segmentation," *Physics in Medicine and Biology*, vol. 41, no. 11, pp. 2437–2446, 1996.

[10] R. Adams and L. Bishof, "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 641–647, 1994.

[11] Y. Yang, D. Tian, and B. Wu, "A fast and reliable noise-resistant medical image segmentation and bias field correction model," *Magnetic Resonance Imaging*, vol. 54, pp. 15–31, 2018.

[12] J. Nie, Z. Xue, T. Liu et al., "Automated brain tumor segmentation using spatial accuracy-weighted hidden Markov Random Field," *Computerized Medical Imaging and Graphics*, vol. 33, no. 6, pp. 431–441, 2009.

[13] M. Havaei, A. Davy, D. Warde-Farley et al., "Brain tumor segmentation with deep neural networks," *Medical Image Analysis*, vol. 35, pp. 18–31, 2017.

[14] S. Pereira, A. Pinto, V. Alves, and C. A. Silva, "Brain tumor segmentation using convolutional neural networks in MRI images," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1240–1251, 2016.

[15] A. S. Miller, B. Blott, and T. K. Hames, "Review of neural network applications in medical imaging and signal processing," *Medical & Biological Engineering & Computing*, vol. 30, no. 5, pp. 449–464, 1992.

[16] R. Lucht, S. Delorme, and G. Brix, "Neural network-based segmentation of dynamic MR mammographic images," *Magnetic Resonance Imaging*, vol. 20, no. 2, pp. 147–154, 2002.

[17] M. Egmont-Petersen, A. F. Frangi, W. J. Niessen et al., "Segmentation of bone tumor in MR perfusion images using neural networks and multiscale pharmacokinetic features," in *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, pp. 80–83, Barcelona, Spain, September 2000.

[18] C. Zhang, K. Qiao, L. Wang et al., "A visual encoding model based on deep neural networks and transfer learning for brain activity measured by functional magnetic resonance imaging," *Journal of Neuroscience Methods*, vol. 325, pp. 108318-108319, 2019.

[19] N. J. Tustison, B. B. Avants, and J. C. Gee, "Learning image-based spatial transformations via convolutional neural networks: a review," *Magnetic Resonance Imaging*, vol. 64, pp. 142–153, 2019.

[20] N. Khalili, N. Lessmann, E. Turk et al., "Automatic brain tissue segmentation in fetal MRI using convolutional neural networks," *Magnetic Resonance Imaging*, vol. 64, pp. 77–89, 2019.

[21] Z. Wang, L. Xie, and J. Qi, "Dynamic pixel-wise weighting-based fully convolutional neural networks for left ventricle segmentation in short-axis MRI," *Magnetic Resonance Imaging*, vol. 66, pp. 131–140, 2020.

[22] A. K. Anaraki, M. Ayati, and F. Kazemi, "Magnetic resonance imaging-based brain tumor grades classification and grading via convolutional neural networks and genetic algorithms," *Biocybernetics and Biomedical Engineering*, vol. 39, no. 1, pp. 63–74, 2019.

[23] J. O. Glass, W. E. Reddick, O. Goloubeva, V. Yo, and R. G. Steen, "Hybrid artificial neural network segmentation of precise and accurate inversion recovery (PAIR) images from normal human brain," *Magnetic Resonance Imaging*, vol. 18, no. 10, pp. 1245–1253, 2000.

[24] R. Pérez de Alejo, J. Ruiz-Cabello, M. Cortijo et al., "Computer-assisted enhanced volumetric segmentation magnetic resonance imaging data using a mixture of artificial neural networks," *Magnetic Resonance Imaging*, vol. 21, no. 8, pp. 901–912, 2003.

[25] W. E. Reddick, J. O. Glass, E. N. Cook, T. D. Elkin, and R. J. Deaton, "Automated segmentation and classification of multispectral magnetic resonance images of brain using artificial neural networks," *IEEE Transactions on Medical Imaging*, vol. 16, no. 6, pp. 911–918, 1997.

[26] M. Bădoiu and K. L. Clarkson, "Optimal core-sets for balls," *Computational Geometry: Theory and Applications*, vol. 40, no. 1, pp. 14–22, 2008.

[27] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 3, pp. 370–379, 1995.

[28] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, New York, NY, USA, 2013.

[29] I. W. Tsang, J. T. Kwok, and J. M. Zurada, "Generalized core vector machines," *IEEE Transactions on Neural Networks*, vol. 17, no. 5, pp. 1126–1140, 2006.

[30] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 6232–6251, 2016.

[31] J. Zheng, J. Cao, Z. Wang et al., "Semi-automatic synthetic computed tomography generation for abdomens using transfer learning and semi-supervised classification," *Journal of Medical Imaging and Health Informatics*, vol. 9, no. 9, pp. 1878–1886, 2019.