Received: 24 January 2020 | Revised: 17 June 2020 | Accepted: 27 July 2020

IET Image Processing

DOI: 10.1049/ipr2.12019

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

Atom search-Jaya-based deep recurrent neural network for liver cancer detection

Mariappan Navaneethakrishnan[1] | **Subbiah Vairamuthu**[2] | **Govindaswamy Parthasarathy**[3] | **Rajan Cristin**[4]

[1] Department of Computer Science and Engineering, St. Joseph College of Engineering, Chennai, Tamil Nadu, India

[2] Department of Software Systems, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India

[3] School of Computing and Information Technology, REVA University, Bengaluru, Karnataka, India

[4] Department of Computer Science and Engineering, GMR Institute of Technology, Rajam, Andhra Pradesh, India

Correspondence
M. Navaneethakrishnan, St. Joseph College of Engineering, Chennai, Tamil Nadu 600119, India.
Email: navaneethakrishnanm9@gmail.com

Abstract

Automatic detection of liver cancer is the fundamental requirement of computer-aided diagnosis in the clinical sector. The traditional methods used in the liver detection process are not effective in accurately detecting the tumour region using a large-sized dataset. Moreover, segmenting the large intensity of the tumour region is a complex issue with the existing methods. To overcome these issues, an accurate and efficient liver cancer detection method named atom search-Jaya-based deep recurrent neural network is proposed in this research. The proposed method mimics the atomic motion based on the interaction forces and the constraint forces of the hybrid molecules. The optimal solution is revealed through the fitness measure, which in turn accepts the minimal error value as the optimal solution. The weights of the classifier are optimally updated based on the position of the atom with respect to the iterations. The proposed atom search-Jaya-based deep recurrent neural network attained significantly better performance in accurately detecting the tumor region using the exploration ability of atoms in the search space. The results obtained by the proposed model in terms of accuracy, specificity, sensitivity, and precision are 93.64%, 96%, 95%, and 94.88%, respectively, while considering the three features using 80% of training data.

1 | INTRODUCTION

The liver has major functions in animals and vertebrates. In the human body, the liver diseases occur without causing any sign or warning [1]. Therefore, detecting the liver diseases at an early stage is a significant keystone in the medical sector [2]. In recent years, liver cancer becomes a dangerous and harmful disease, as it escorts to human death. The most significant type of liver cancer among the worldwide level is cholangiocarcinoma and hepatocellular carcinoma (HCC), respectively [3, 4]. Based on the geographical region and gender, the cause of liver cancer may vary [4, 5]. In the worldwide level, the rate of liver cancer increases, and most of the patients with cancer died even after taking the diagnosis for six months [4]. However, the liver cancer specifies a high changeability in their localization, appearance, and shape [6]. The liver cancer may be either hyperdense or hypodense. Here, the hyperdense appears brighter, whereas the hypodense looks darker when compared to the healthy parenchyma liver [7]. However, the appearance of liver cancer is based on the state, lesion type, and imaging, such as timing, contrast method, and settings, and also the type of cancer may vary from patient to patient [8, 9]. The liver cancers are of two types, namely primary and secondary liver cancer. The primary cancer originates from the liver and is also termed as hepatoma or HCC. In the secondary liver cancer, the cancer cell originates from various human organs and passes into the liver [10–12]. Moreover, the diagnosis of cancers is carried out in three different ways: biopsy, imaging test, and blood test [13, 14]. Among these testing models, the imaging test is highly focused on in order to receive the computer-assisted approach. The imaging tests used to diagnose the liver cancer are computed tomography (CT).

CT scan is the most robust and common imaging method used to detect the lesions in the liver. CT is a non-invasive procedure or medical examination that uses an X-ray equipment for producing the cross-sectional images in the parts of a human body [15]. This imposes the requirement to model the automated system for classifying and detecting the liver anomalies based on the images of the CT scan [16]. In the CT scan image, each cross-sectional image specifies the slice of a liver, which is similar to the slice of a bread. In recent years, the quality of CT scan images has improved, which limits the role of an image interpretation in human beings [8]. However, the computer-assisted detection mechanism is based on the principle of image analysis, which in turn offers more information regarding the detection of liver cancer. Moreover, the conventional liver cancer detection method consists of three different steps: first, the tumor is segmented from the CT scan abdominal images; second, the features are extracted; and third, the classification mechanism is carried out based on the selected classifiers. Based on the texture features, characteristics of the liver images are modelled in the past years [17]. The major issue in detecting the liver tumor is the poor contrast between the values of liver intensity and tumor. Initially, the tumor is present in a small region, hence it is very difficult to detect the tumor. Therefore, this research focusses on increasing the contrast of the image that contains a tumor in the liver. The images are pre-processed to enable the user identify the presence of a tumor through CT images. To detect the liver cancer at an early stage is a major advantage in preventing the liver cancer [18, 19].

Classification of liver cancer poses a major role in diagnosing the tumor at an early stage. The deep learning methods based on deep neural network (DNN) is used to solve a wide range of issues in the recent works [20–22]. To detect the liver cancer automatically, the convolutional neural network (CNN) is used to segment the lesions effectively from the CT images and a coefficient of dice similarity value of 80.06% is obtained [22]. DNN is learned by identifying the top-level features from the bottom-level features in the hierarchy level, and the machine classification is performed by Fukushima. Optimization algorithms are used to train DNN [23]. However, the deep learning methods are categorized into sparse autoencoder, deep belief network (DBN), restricted Boltzmann machine (RBM), and CNN, respectively [19]. The deep learning model specifies the advanced structure in CNN. Based on the convolutional procedure, the features extracted from the input data are passed to the successive layers based on their respective features. Here, the convolutional procedure is used to retrieve the suitable properties from the CT image. Lu et al. [24] introduced the deep learning model based on the graph cut refinement for efficiently and automatically segmenting the CT images. In [25], the deep learning methods, such as fully convolutional networks, DBN, autoencoder, and CNN, are introduced for detecting and diagnosing liver cancer. An automated detection model is developed in [22] by combining the deep learning classifier and Gaussian mixture model (GMM) for detecting the region of the liver lesion [22].

The detection of liver and its lesion poses a challenging issue due to the variability of high anatomical shape and the facts regarding the contact of neighbouring pixel intensity [26]. There exist a number of difficulties in the analysis of automatic cancer detection that includes less image contrast among the tissue samples due to the differences in scan time and perfusion, parenchyma, and the behaviour of lesions in the contrast enhancement [19]. To extract the boundary region of liver based on the abdominal CT image poses a major challenge in the computer-aided diagnosis, as the boundary region is weak in liver cancer [21]. The detection of liver cancer from the CT images poses a challenging issue due to less contrast among the liver and tumor. However, the presence of other human parts with various dimensions and equivalent intensity level makes the detection process more difficult [19]. Due to the complexity of liver anatomy issues and the insufficiency of organ shape, the accurate segmentation remains a difficult process. These problems are considered a motivation, a new method is proposed for the liver cancer detection.

In this research, the liver cancer detection mechanism is performed using the proposed atom search-Jaya-based deep recurrent neural network (AS-Jaya-based Deep RNN). The proposed work highly concentrates on detecting the liver cancer based on the parametric features of atomic motion and control parameters. Initially, the input images are pre-processed and are segmented using black hole entropic fuzzy clustering (BHEFC). The segmented result obtained from the segmentation module is allowed to perform the feature extraction phase, where the features are effectively extracted. The feature extraction module ensures the effectiveness of detection accuracy. Deep RNN effectively detects the liver lesion based on weights, as the weights are optimally trained using the optimization algorithm. The proposed work is modelled based on the inspiration of basic molecular dynamics and is functioned with the population-based heuristic model. The proposed optimization achieved effective performance in cancer detection through the mimicking behaviour of atomic motion, which is controlled using the constraint and the interaction (total) forces. The training procedure of the Deep RNN is carried out with the proposed AS-Jaya optimization in order to update the weights of the classifier using the position of the atom.

The major contributions of this research are elaborated as follows. The liver cancer detection model is achieved using the proposed AS-Jaya-based Deep RNN classifier. The characteristic features of AS-Jaya tune the classifier optimally to achieve the best detection result through the optimal fitness value. The fitness measure is evaluated based on the position of an atom, which in turn reflects the accuracy and the robustness of detection measure.

The paper is organized as follows. Section 2 elaborates the literature review of the existing liver cancer detection methods. Section 3 discusses the proposed AS-based Deep RNN for detecting the liver cancer. Section 4 elaborates the results and discussion of the proposed method, and finally, Section 5 concludes the paper.

2 | LITERATURE SURVEY

Various existing liver cancer detection methods are reviewed in this section. Das et al. [22] introduced a watershed Gaussian-based deep learning (WGDL) approach for effectively delineating the liver lesion through CT images. The presence of the cancer region was segmented with the Gaussian model. After segmenting the tumour, the extraction of texture features was carried out for classifying the tumor region. It increased the classification accuracy but failed to consider the volumetric image. Priya et al. [27] developed an efficient fusion model for increasing the information of edge location in the CT images. It utilized the Laplacian operators and phase congruency to generate the fusion of high- and low-frequency coefficients. The computational complexity of this model was high. Das et al. [28] developed an approach to detect the liver cancer regions in CT scan images, by integrating the spatial fuzzy clustering approach and adaptive thresholding. This method was used to effectively identify the cancer region, without any manual process. Anyhow, this method did not apply to the large datasets. Wang et al. [29] developed an electrochemical detection model to increase the efficiency in detecting the presence of the liver tumour. It reached the detection limit of 125 fM with less detection time. However, it failed to consider the amplification system.

Bi et al. [30] developed a deep residual network (Deep ResNet) for segmenting the liver tumour. It consisted of skip connections between the convolutional layers and was used to solve the issues in the training accuracy of deep networks. Here, the multi-scale fusion was used to obtain the precise boundary in the liver lesion. However, the performance achieved by this model was poor. Wang et al. [31] developed a cantilevel-based biosensor model to diagnose the liver cancer by achieving high throughput and sensitivity. It effectively detected the multi-biomarkers in the cantilever array. It attained high detection accuracy with low cost and volume. However, the performance in the resolution and accuracy still needs to be improved. Gruber et al. [32] developed a deep learning approach for segmenting the liver cancer. It used two different types of network segments to accurately segment the outer and inner tumour regions. This approach performed the segmenting procedure automatically and achieved better performance. However, it required more number of iterations to segment the tumor region. Kumar and Bharathi [33] introduced an edge prediction based segmentation model for segmenting the tumour region from the liver. It effectively classified abnormal and normal lesions using a machine learning classifier. The computational cost required in this model was high.

3 | PROPOSED ATOM SEARCH-JAYA BASED DEEP RECURRENT NEURAL NETWORK FOR LIVER CANCER DETECTION

Liver cancer is one of the leading diseases that causes death in males [34]. To accurately detect the liver cancer, there are various techniques, but they face challenge regarding the pixel intensities. Hence, an effective method is developed to perform the liver cancer detection framework based on BHEFC. Initially, the input image is passed on to the pre-processing stage, which is subjected to segmentation using BHEFC [35, 36], which generated multi-segments. The segmented result is fed to the feature extraction stage, where the features such as CNN features, statistical features, and the pixel pattern based texture features (PPBTFs) are effectively extracted. The statistical features include mean, variance, skewness, energy, and kurtosis, respectively. The features that are extracted from the segmented result are passed into the Deep RNN classifier, which is trained using the proposed AS-Jaya-based Deep RNN that is the integration of atomic search optimization (ASO) [37] with the Jaya optimization [38]. Figure 1 represents the schematic diagram of liver cancer detection modules.

3.1 | Read the input image

The input image is collected from the liver cancer dataset [11] and is used to perform the cancer detection process. Let us consider the database $M$with $m$number of cancer images, which is represented as

$$M = \{R_1, R_2, \dots R_k, \dots R_\varpi\}, \tag{1}$$

where M represents the database, ϖ denotes the total number of input images, and R_K represents the image located at the kth index. The total images are considered for experimentation, but the explanation is provided based on the kth image.

3.2 | Pre-processing the input image

The input image R_k is selected and is pre-processed in the pre-processing module in order to enhance the contrast of the image. The aim of pre-processing is to increase the quality of the image by removing unwanted falsification or distortions. It is required to pre-process the image in order to make the image highly contrasting and effective for further processing. The result of the pre-processed image is denoted by R_p.

3.3 | Segmentation using BHEFC

Segmentation is the process of partitioning the pre-processed image into various segments, such as pixels or image objects. The pre-processed image R_p is passed into the segmentation phase, where the segmentation process is carried out using BHEFC [35]. The BHEFC generates samples based on the assumption of the Dirichlet distribution of the fuzzy membership function. BHEFC performs the segmentation process by absorbing the merits of three aspects, such as black hole entropy (BHE) based information, fuzzy clustering, and Bayesian inference model. BHEFC is developed by integrating the BHE with the fuzzy clustering. BHEFC accurately and effectively characterizes the behaviour of clustering using the principle called maximum-a-posteriori. BHEFC uses the Gibbs sampler for generating the samples from the posteriori distribution. According to the fuzzy c-means clustering and principle of Lagrangian

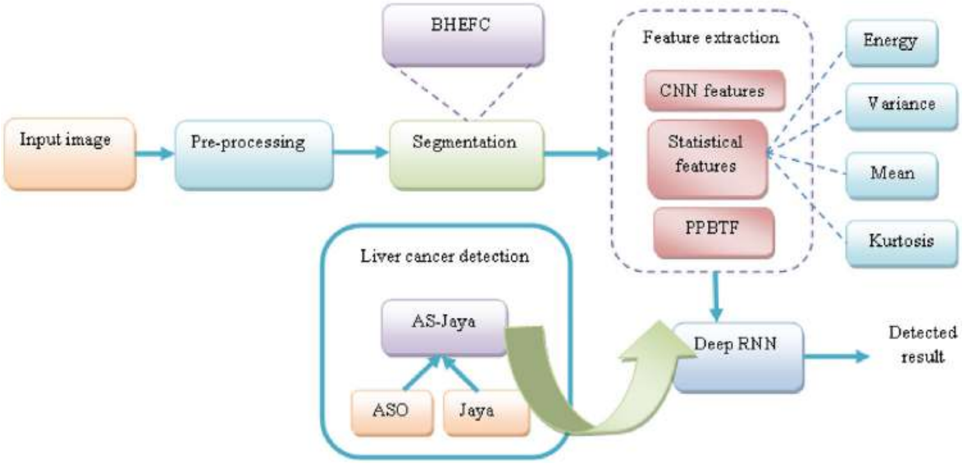

FIGURE 1 Schematic diagram of liver cancer detection modules

optimization, the BHEFC is modelled as

$$P\left(N, T, S\right) = P\left(N \middle| T, S\right) \tilde{P}\left(T \middle| S\right) \cdot P\left(S\right)$$
$$\propto \ \exp\left\{ -\frac{1}{2} \sum_{p=1}^{\tau} \sum_{l=1}^{Q} q_p^a \left\| R_p - Z_l \right\|^2 \right\} \times \left(\prod_{p=1}^{\tau} \prod_{l=1}^{Q} q_p^{-a} \right) \times$$
$$\exp\left\{ -\frac{1}{2} \left(\mu + \alpha \sum_{p=1}^{\tau} \sum_{l=1}^{Q} \ln \left\| R_p - Z_p \right\|^2 \right) \right\}, \tag{2}$$

where λ represents the vector, α denotes the factor, which is set to '2' for simplifying the segmentation process. R_p denotes the pre-processed image, q_p indicates the fuzzy membership function, S denotes the clustering centre set, T specifies the membership matrix, Z_l indicates the cluster centre, and N denotes the cluster dataset, respectively. Finally, the segmented result obtained using the BHEFC is represented as R_s such that $R_s = \tilde{P}(R_p, q_p|S)$.

3.4 | Feature extraction using the segments

The segmented result R_s is passed into the feature extraction phase, where the features, such as CNN features, statistical features, and the PPBTFs, are effectively extracted. The feature extraction is the process of transforming the segmented image into the confined matrix such that the processing complexity is relieved. The features that are extracted from the image R_s are discussed as follows:

(i) *CNN features*: CNN is the multi-layered network with the special architecture, which is used to detect the complex features from the result of the segmented image. The architecture of CNN consists of three different layers, namely convolution layer, pooling layer, and fully connected layer. Convolution is the first layer, which helps to extract the essential features from the segmented image. It preserves the relations between the image features and the pixel values. It takes the segmented result R_s as input and extracts the CNN features through the convolution layer. The features that are extracted at the first convolution layer are termed as CNN features and are represented as f_ω with the dimension of [1 × 256], respectively. The CNN feature extracted using the convolution layer is depicted in Figure 2.

(ii) *Statistical features*: The extraction of statistical features assures the effectiveness of cancer detection. The features that are extracted through the statistical analysis are termed as statistical features. Here are some of the statistical features extracted from the segmented result that are listed as follows:

- *Mean* The pixels present in the segmented results are averaged to compute the mean value.
- *Variance* The variance feature f_1 is calculated based on the value of the mean.
- *Kurtosis and skewness* Kurtosis refers to the symmetry and is denoted as f_3. Skewness specifies the shape of the object with respect to the numerical value and is indicated as f_4, respectively.
- *Energy* The energy of the individual segments is extracted by summing the energy of all the pixels in the segment.

Finally, the statistical features extracted from the segmented result are represented as $f_\upsilon = \{f_1, f_2, f_3, f_4, f_5\}$ with the dimension of [1 × 5], respectively.

(iii) *PPBTFs*: The grayscale image is transformed into the pattern map, where the background and the edge pixels are classified using the pattern matching that returns the spatial features. The pixel values present in the pattern map specifies the pattern classes of the image pixels. However, the PPBTF [39] extracted from the segmented result is represented as f_ρ, with the dimension of [1 × 32]. Finally, the features extracted from the segmented result R_s are denoted as F such that F includes $\{f_\omega, f_\upsilon, f_\rho\}$, respectively.

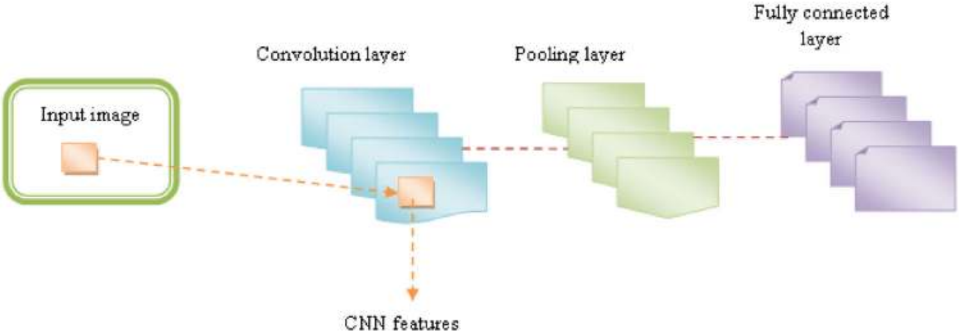

FIGURE 2 Extraction of CNN features

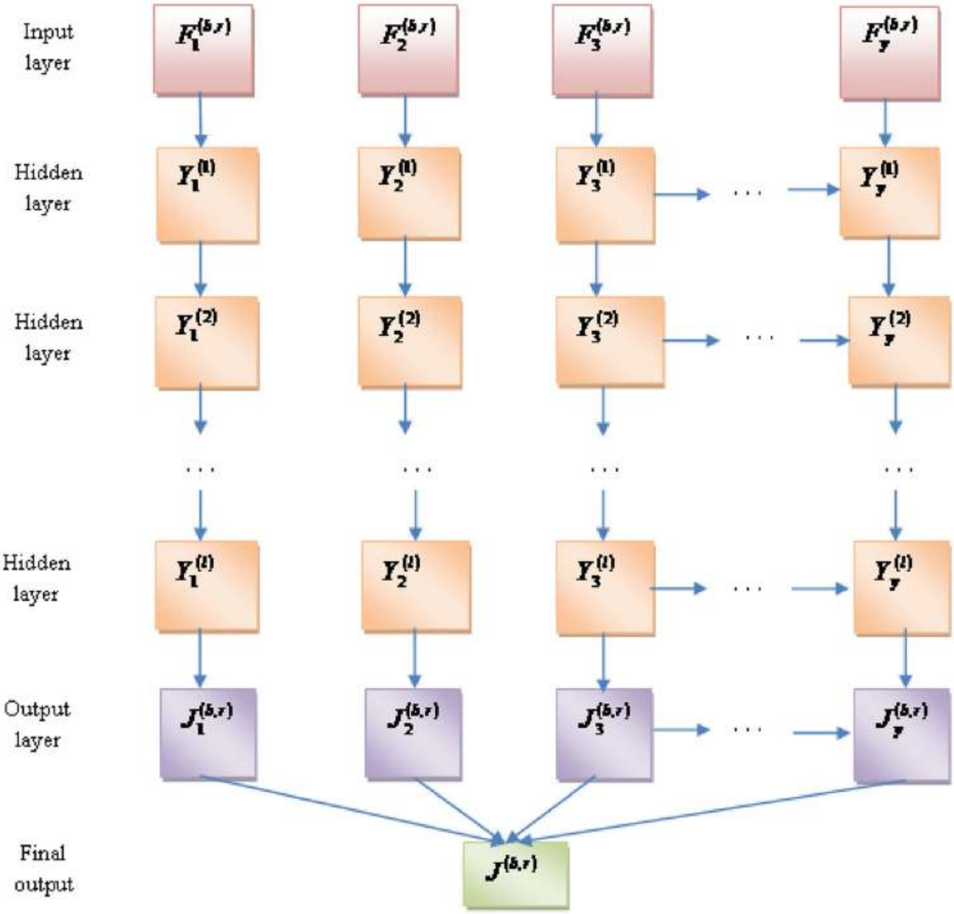

FIGURE 3 Architecture of Deep RNN classifier

3.5 | Deep recurrent neural network for liver cancer detection

The features F that are extracted from the segmented images are given as the input to the Deep RNN classifier. Deep RNN [40] is the network architecture that contains multiple recurrent hidden layers in the layer of network hierarchy. The architecture of Deep RNN is shown in Figure 3.

The structure of Deep RNN is made by considering the input vector of the bth layer at the rth time as $F^{(b,r)} = \{F_1^{(b,r)}, F_2^{(b,r)}, \dots F_i^{(b,r)}, \dots F_y^{(b,r)}\}$ and the output vector of the bth layer at the rth time as $J^{(b,r)} = \{J_1^{(b,r)}, J_2^{(b,r)}, \dots J_i^{(b,r)}, \dots J_y^{(b,r)}\}$, respectively. The pair of each elements of input and the output vectors is termed as the unit. Here, i denotes the arbitrary unit number of the bth layer and y represents the total number of units of the bth layer. In addition, the arbitrary unit number and the total number of units of $(b-1)$th layer is denoted as j and E, respectively. At this time, the input propagation weight from the $(b-1)$th layer to the bth layer is represented as $W^{(b)} \in H^{y \times E}$, and the recurrent weight of the bth layer is represented as $w^{(b)} \in H^{y \times y}$. Here, H denotes the set of weights. However, the components of the input vector is expressed as

$$F_i^{(b,r)} = \sum_{z=1}^{E} p_{iz}^{(b)} J_z^{(b-1,r)} + \sum_{i'}^{y} x_{ii'}^{(b)} J_{i'}^{(b,r-1)}, \tag{3}$$

where $p_{iz}^{(b)}$ and $x_{ii'}^{(b)}$ are the elements of $W^{(b)}$ and $w^{(b)}$. i' denotes the arbitrary unit number of the bth layer. The elements of the output vector of the bth layer is represented as

$$J_i^{(b,r)} = \beta^{(b)} \left(F_i^{(b,r)}\right), \tag{4}$$

where $\beta^{(b)}$ denotes the sigmoid activation function. The rectified linear unit function is denoted as $\beta(F) = \max(F, \ell)$, and the logistic sigmoid function is denoted as $\beta(F) = \frac{1}{(1+e^{-F})}$, which are the frequently used activation functions. To simplify the process, let us introduce the ℓth weight as $p_{i\ell}^{(b)}$ and the ℓth unit as $J_\ell^{(b-1,r)}$ and hence, the bias is represented as

$$J^{(b,r)} = \beta^{(b)} \cdot \left(W^{(b)} J^{(b-1,r)} + w^{(b)} \cdot J^{(b,r-1)}\right). \tag{5}$$

Here,$J^{(b,r)}$ denotes the output of the classifier.

3.5.1 | Training of deep recurrent neural network

The training of the Deep RNN classifier is done using the proposed optimization algorithm named AS-Jaya optimization in order to obtain the optimal weights to achieve the update process.

3.6 | Proposed AS-Jaya optimization for training the deep recurrent neural network classifier

In the clinical diagnosis sector, the pathologists rely on the examination of the set of nuclei in the tissue sample. In most of the situations, the diagnostic label is used for the tissue sample, and hence, the optimization algorithm is required to find the sets of nuclei among the patients with cancer in order to detect liver cancer. To achieve the liver cancer detection mechanism, the optimization named AS-Jaya is proposed in this research. The proposed AS-Jaya optimization is used to train the weight of the Deep RNN classifier in order to generate the optimal solution with the optimal detection rate. The proposed AS-Jaya optimization is the integration of ASO [37] with the Jaya optimization [38], which extracts the parametric features from both the optimization, towards boosting the performance of detection accuracy. Here, the location of each atom lying in the search space specifies the solution that is measured by the mass. All the atoms present in the population either attract or repel each other based on the distance between them. The heavier atoms have lower acceleration, which makes them obtain a better solution in the search space. The lighter atoms have higher acceleration, which intensively finds a new region in the search space.

Solution encoding

It is the representation of the solution vector, which determines the optimal solution for detecting liver cancer. The optimal solution is computed based on the fitness function. The fitness function with the minimal error value is accepted as the best solution. The algorithmic steps of the proposed AS-Jaya optimization based Deep RNN are elaborated as follows:

(i) *Initialize the population*: The unconstrained optimization problem is solved by initializing the population of atoms. Here, the term atom represents the image. Let us consider the population with ε number of atoms and the location of n^{th} atom is represented as

$$h_n = \left[h_n^1, \dots h_n^K\right]; \quad n = \{1, \dots \ell\}, \tag{6}$$

where h_n^t $(t = 1, \dots K)$ represents the tth position component of the nth atom in the Kth dimensional space.

(ii) *Fitness function*: The fitness function is calculated based on the difference between the actual output value and the estimated output value. The function with the minimal fitness value is accepted as the optimal solution. However, the fitness function is computed using the following expression:

$$L_n = \frac{1}{\sigma} \sum_{\nu=1}^{\sigma} J_\nu^{(b,r)} - \kappa_\nu, \tag{7}$$

where L_n denotes the fitness value of the nth atom,$J_\nu^{(b,r)}$ denotes the output of the classifier, and κ_ν denotes the estimated output.

(iii) *Compute the mass*: The mass of the atom is measured at the lowest level based on the fitness value. However, the mass of the nth atom at the gth iteration is computed as

$$G_n(g) = \frac{\chi_n(g)}{\sum_{n=1}^{\ell} \chi_n(g)}, \tag{8}$$

where $G_n(g)$ represents the mass, and the term $\chi_n(g)$ is computed as

$$\chi_n(g) = \frac{L_n - L_{bst}}{e^{L_{wst} - L_{bst}}}. \tag{9}$$

Here, L_{bst} and L_{wst} are represented as $L_{bst} = \min_{n=1,\ldots\ell} L_n$, and $L_{wst} = \max_{n=1,\ldots\ell} L_n$, respectively.

(iv) *Determine theD neighbours*: Each atom interacts with other atoms based on the best fitness value with D neighbours. Therefore, with the function of time, D gradually decreases with respect to the lapse of iterations. Hence, D is computed as

$$D(g) = \ell - (\ell - 2) \times \sqrt{\frac{g}{\gamma}}, \tag{10}$$

where γ is the maximum number of iterations.

(vi) *Compute the total force and constraint force*: The sum of all the components being used with the random weight that is acted on the nth atom from other atoms is considered as the total force and is expressed as

$$W_n^t(g) = \sum_{u \in Dbst} rand_u W_{nu}^t(g), \tag{11}$$

where $W_n^t(g)$ denotes the total force, and $rand_u$ denotes the random number that lies in the range of 0–1, respectively. Each atom in the population space is acted based on the constraint force from the best atom, and so the constraint force of the nth atom is represented as

$$O_n^t(g) = \eta(g)\left(h_{bst}^t(g) - h_n^t(g)\right), \tag{12}$$

where $\eta(g)$ denotes the Lagrangian multiplier, $h_{bst}^t(g)$ represents the position of best atom at the gth iteration, and $O_n^t(g)$ denotes the constraint force. Here, the Lagrangian multiplier is represented as

$$\eta(g) = \partial e^{\frac{20g}{\gamma}}, \tag{13}$$

where ∂ denotes the multiplier weight.

(vii) *Compute the acceleration*: Based on the geometric constraint and the total force, the acceleration of the nth atom at the gth time is expressed as

$$z_n^t(g) = \frac{W_n^t(g)}{G_n(g)} + \frac{O_n^t(g)}{G_n(g)}, \tag{14}$$

where $W_n^t(g)$ is the total force, $O_n^t(g)$ is the constraint force, and $G_n(g)$ denotes the mass.

(viii) *Update the velocity*: Based on the $(g+1)$ iteration, the velocity of the nth atom is updated as

$$c_n^t(g+1) = rand_n^t\, c_n^t(g) + z_n^t(g), \tag{15}$$

where $rand_n^t$ denotes the random number and $z_n^t(g)$ indicates the acceleration.

(ix) *Update the position of atom*: The update process of the proposed AS-Jaya optimization is carried out by modifying the position of the nth atom of ASO with the nth candidate solution of Jaya optimization. The position of the nth atom at the $(g+1)$th iteration is represented as

$$h_n^t(g+1) = h_n^t(g) + c_n^t(g+1) \tag{16}$$

$$h_n^t(g+1) = h_n^t(g) + rand_n^t\, c_n^t(g) + z_n^t(g). \tag{17}$$

Equation 17 specifies the position of the nth atom, which is modified with the candidate solution of Jaya optimization. The updated equation of the candidate solution of Jaya optimization is represented as

$$h_n^t(g+1) = h_n^t(g) + m_{1n}^t\left(h_{bst,n}^t - \left|h_n^t(g)\right|\right) \\ -m_{2n}^t\left(h_{wst,n}^t - \left|h_n^t(g)\right|\right). \tag{18}$$

Let us assume $h_n^t(g)$ as positive, and hence, the above equation can be rewritten as Equations (A1)–(A3) provided in the Appendix.

Equation (19) specifies the updated equation of Jaya optimization. By substituting Equation (A3) in Equation (17), the resultant equation is expressed as Equations (A4)–(A11) (given in the Appendix).

Here $h_n^t(g+1)$ represents the position of the nth atom at the $(g+1)$th iteration, $rand_n^t$ denotes the random number, $z_n^t(g)$ indicates the acceleration, $h_{wst,n}^t$ represents the value for worst candidate, and m_{2n}^t and m_{1n}^t are the random number that lie in the interval [0, 1]. The equation (19) is the standard equation of the proposed AS algorithm, which is obtained by modifying the ASO with the Jaya optimization. By integrating the ASO with the Jaya provides more beneficial result in liver cancer detection with less computation cost and time. The integration of the parametric features of ASO with the Jaya optimization ensures the effectiveness of performance in the liver cancer detection model. The mimicking behaviour of atomic motion results in the potential and constraint forces of atoms from the best atom. Algorithm 1 specifies the pseudo-code of the proposed AS-Jaya-based Deep RNN.

4 | RESULTS AND DISCUSSION

The proposed AS-Jaya-based Deep RNN with respect to the evaluation metrics is discussed in this section.

4.1 | Evaluation metrics

The performance attained by the proposed method is evaluated based on the performance metrics such as sensitivity, specificity, accuracy, and precision [4].

ALGORITHM 1 Pseudo-code of the proposed AS-Jaya-based Deep RNN

```
Pseudo code of the proposed AS-Jaya-based Deep RNN
Input: $h_n$
Output: $h_n^t(g+1)$
Initialize the set of atoms $A$ and the velocity $c$
While stopping criteria is not satisfied
Do
    Compute $L$
    if $(L_n < L_{bst})$ then
        $L_{bst} = L_n$
        $A_{bst} = A_n$
    End if
    Compute $G_n(g)$
    Determine the $D$ neighbours
    Compute $W_n^t(g)$ and $O_n^t(g)$
    Calculate $z_n^t(g)$
    Update $c_n^t(g+1)$
    Update $h_n^t(g+1)$ using Equation (A10)
End while
Return $A_{bst}$
```

4.1.1 | Accuracy

Accuracy is the measure used to differentiate the proportion of true-positive rate and the true-negative rate, which is represented as

$$Acc = \frac{TP + TN}{TP + FN + FP + FN}, \tag{20}$$

where TP represents the true-positive rate, TN denotes the true-negative, FP indicates the false-positive, and FN is the false-negative.

4.1.2 | Sensitivity

It is the measure used to determine the accurate result of true-positive rate, which is represented as

$$Sen = \frac{TP}{TP + FN}. \tag{21}$$

4.1.3 | Specificity

Specificity is the measure used to determine the true-negative result, which is represented as

$$Spe = \frac{TN}{TN + FP}. \tag{22}$$

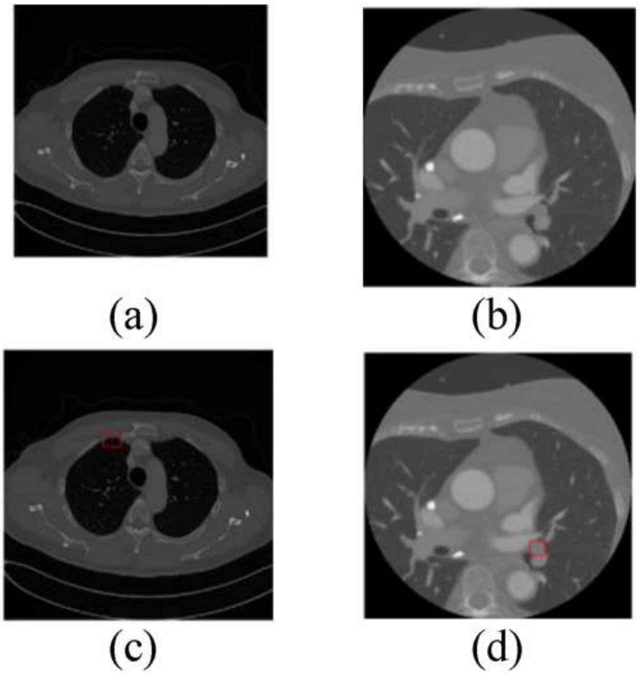

FIGURE 4 Experimental result: (a) input image-1, (b) input image-2, (c) detected result-1, and (d) detected result-2

4.1.4 | Precision

Precision is the measure used to determine the positive predictive value, which is represented as

$$Pre = \frac{TP}{TP + FP}. \tag{23}$$

4.2 | Experimental set-up

The implementation of the proposed method is carried out in the MATLAB tool [41] using the dataset specified in [11]. It contains 416 liver patient records among which 167 patient records are collected from the northeast of Andhra Pradesh.

4.3 | Experimental result

Figure 4 represents the experimental result of the proposed AS-Jaya-based Deep RNN. Figure 4(a) portrays the input image 1, Figure 4(b) shows the input image 2. Figure 4(c) represents the detected result 1, and Figure 4(d) represents the detected result 2.

4.4 | Performance analysis

The performance analysis conducted using the proposed AS-Jaya-based Deep RNN by varying the training percentage and hidden layers is discussed as follows.

TABLE 1 Performance analysis of the proposed AS-Jaya-based Deep RNN based on training percentage

Training percentage Population size	40%	50%	60%	70%	80%
	Specificity				
5	94.41	94.53	94.65	94.85	94.89
10	95.51	95.63	95.75	95.96	96.00
15	95.51	95.63	95.75	95.96	96.00
20	95.51	95.63	95.75	95.96	96.00
25	95.51	95.63	95.75	95.96	96.00
	Sensitivity				
5	70.48	70.60	70.70	70.83	70.93
10	79.84	79.95	80.05	80.18	80.28
15	84.67	84.78	84.88	85.01	85.11
20	87.75	87.86	87.96	88.09	88.19
25	89.96	90.07	90.17	90.30	90.40
	Accuracy				
5	92.39	92.52	92.65	92.85	92.92
10	92.57	92.70	92.83	93.03	93.10
15	92.75	92.88	93.01	93.21	93.28
20	92.94	93.07	93.19	93.39	93.46
25	93.12	93.25	93.37	93.57	93.65
	Precision				
5	85.76	85.91	86.02	86.18	86.25
10	89.30	89.46	89.57	89.73	89.79
15	90.97	91.13	91.24	91.40	91.46
20	92.06	92.22	92.32	92.49	92.55
25	92.86	93.02	93.12	93.29	93.35

TABLE 2 Performance analysis of the proposed AS-Jaya-based Deep RNN based on hidden layers

Hidden layers Population size	2	4	6	8	10
	Specificity				
5	95.47	95.64	95.79	95.94	96.00
10	95.47	95.64	95.79	95.94	96.00
15	95.47	95.64	95.79	95.94	96.00
20	95.47	95.64	95.79	95.94	96.00
25	95.47	95.64	95.79	95.94	96.00
	Sensitivity				
5	84.67	84.71	84.87	84.99	85.08
10	89.76	89.80	89.96	90.07	90.17
15	92.50	92.54	92.70	92.81	92.91
20	94.27	94.31	94.47	94.59	94.68
25	94.59	94.63	94.79	94.90	95.00
	Accuracy				
5	92.38	92.47	92.69	92.86	92.92
10	92.57	92.65	92.87	93.04	93.10
15	92.75	92.83	93.05	93.22	93.28
20	92.93	93.01	93.23	93.40	93.46
25	93.11	93.19	93.41	93.58	93.65
	Precision				
5	90.81	90.95	91.10	91.27	91.33
10	92.57	92.71	92.86	93.03	93.09
15	93.54	93.68	93.83	94.00	94.06
20	94.20	94.34	94.48	94.65	94.72
25	94.36	94.50	94.65	94.82	94.88

4.4.1 | Performance analysis with training percentage

Table 1 portrays the performance analysis of the proposed detection mechanism by varying the training percentage. When the training percentage is 40%, the specificity obtained by the proposed AS-Jaya-based Deep RNN with population size 5 is 94.41%, population size 10 is 95.51%, population size 15 is 95.51%, population size 20 is 95.51%, and population size 25 is 95.51%, respectively. When training percentage is 50%, the sensitivity obtained by the proposed AS-Jaya-based Deep RNN with population size 5 is 70.60%, population size 10 is 79.95%, population size 15 is 84.78%, population size 20 is 87.86%, and population size 25 is 90.07%, respectively.

When training percentage is 60%, the accuracy obtained by the proposed AS-Jaya-based Deep RNN with population size 5 is 92.65%, population size 10 is 92.83%, population size 15 is 93.01%, population size 20 is 93.19%, and population size 25 is 93.37%, respectively. For the training percentage is 50%, the AS-Jaya-based Deep RNN with population size 5 is 85.91%, population size 10 is 89.46%, population size 15 is 91.13%, population size 20 is 92.22%, and population size 25 is 93.02%, respectively.

4.4.2 | Performance analysis based on hidden layers

Table 2 portrays the performance analysis of the proposed detection mechanism by varying the hidden layers. When the number of hidden layers is 8, the specificity obtained by the proposed AS-Jaya-based Deep RNN with population sizes 5, 10, 15, 20, and 25 is 95.94%. When the number of hidden layers is 10, the sensitivity obtained by the proposed AS-Jaya-based Deep RNN with population size 5 is 85.08%, population size 10 is 90.17%, population size 15 is 92.91%, population size 20 is 94.68%, and population size 25 is 95.00%, respectively.

When the number of hidden layers is 8, the accuracy obtained by the proposed AS-Jaya-based Deep RNN with population size 5 is 92.86%, population size 10 is 93.04%, population size 15 is 93.22%, population size 20 is 93.40%, and population size 25 is 93.58%, respectively. When the training percentage is 50%, the AS-Jaya-based Deep RNN with population size 5 is 90.95%,

TABLE 3 Comparative analysis using two features based on the training percentage

Method Metrics	GMM	SetSVM	CNN	Deep ResNet	Proposed AS-Jaya-based Deep RNN
Specificity (%)	73.27	90.33	94.76	93.33	94.90
	73.41	93.35	96.00	95.73	96.00
	73.54	94.20	96.00	96.00	96.00
	73.68	94.73	96.00	96.00	96.00
	73.81	95.38	96.00	96.00	96.00
Sensitivity (%)	70.37	70.56	70.74	0.74	70.93
	70.56	70.74	78.94	76.65	80.28
	70.74	70.93	80.28	78.77	85.11
	70.93	71.11	83.73	81.01	88.19
	71.11	71.30	85.94	82.55	90.40
Accuracy (%)	70.37	84.54	92.73	90.07	92.92
	70.55	87.73	92.92	91.25	93.10
	70.73	88.16	93.10	91.51	93.28
	70.91	89.04	93.28	91.93	93.46
	71.09	89.63	93.46	92.25	93.65
Precision (%)	71.34	81.81	86.08	84.71	86.25
	71.50	83.94	89.28	87.88	89.79
	71.67	84.43	89.79	88.76	91.46
	71.84	84.96	91.00	89.65	92.55
	72.01	85.44	91.80	90.26	93.34

TABLE 4 Comparative analysis using three features based on the training percentage

Method Metrics	GMM	SetSVM	CNN	Deep ResNet	Proposed AS-Jaya-based Deep RNN
Specificity (%)	73.27	95.45	96.00	96.00	96.00
	73.41	96.00	96.00	96.00	96.00
	73.54	96.00	96.00	96.00	96.00
	73.67	96.00	96.00	96.00	96.00
	73.81	96.00	96.00	96.00	96.00
Sensitivity (%)	70.37	70.56	84.90	80.18	85.08
	70.56	70.74	89.99	83.63	90.17
	70.74	70.93	92.72	85.52	92.91
	70.93	71.11	94.50	86.76	94.68
	71.11	71.30	95.00	87.68	95.00
Accuracy (%)	70.37	89.44	92.73	91.70	92.92
	70.55	89.94	92.92	91.98	93.10
	70.73	90.37	93.10	92.25	93.28
	70.91	91.01	93.28	92.59	93.46
	71.09	91.59	93.46	92.90	93.65
Precision (%)	71.34	85.15	91.21	89.29	91.33
	71.50	85.54	92.97	90.54	93.09
	71.67	85.77	93.94	91.26	94.06
	71.84	86.04	94.59	91.78	94.72
	72.01	86.29	94.82	92.19	94.88

population size 10 is 92.71%, population size 15 is 93.68%, population size 20 is 94.34%, and population size 25 is 94.50%, respectively.

4.5 | Comparative methods

The performance of the proposed method is revealed by comparing the proposed method with the existing methods, such as GMM [22], set support vector machine (SetSVM) [7], CNN [21], and Deep ResNet [30], respectively.

4.6 | Comparative analysis

The comparative analysis of the proposed algorithm is discussed by considering the extracted features with respect to the training percentage.

4.6.1 | Comparative analysis by considering two features

Table 3 shows the comparative analysis of the proposed algorithm by considering two features, namely statistical features and PPBTFs. When training percentage is 40%, the specificities obtained by the existing GMM, SetSVM, CNN, and Deep ResNet are 73.27%, 90.33%, 94.76%, and 93.33%, respectively, while the specificity of the proposed AS-Jaya-based Deep RNN is 94.90%. When training percentage is 70%, the sensitivity obtained by the proposed AS-Jaya-based Deep RNN is 88.19%, while the percentage of improvement reported when comparing the proposed AS-Jaya-based Deep RNN with the existing GMM, SetSVM, CNN, and Deep ResNet is 24%, 24%, 5%, and 8%, respectively.

When training percentage is 50%, the accuracy obtained by the existing GMM, SetSVM, CNN, and Deep ResNet is 70.55%, 87.73%, 92.92%, and 91.25%, respectively, while the accuracy of the proposed AS-Jaya-based Deep RNN is 93.10%. For training percentage is 60%, the precision of the methods such as GMM, SetSVM, CNN, and Deep ResNet, and the proposed AS-Jaya-based Deep RNN are 71.67%, 84.43%, 89.79%, 88.76%, and 91.46%, respectively.

4.6.2 | Comparative analysis by considering three features

Table 4 shows the comparative analysis of the proposed algorithm by considering three features, namely statistical features, CNN features, and PPBTFs. When training percentage is 50%, the specificity obtained by the existing GMM, SetSVM, CNN,

TABLE 5 Comparative discussion

Metrics/Methods	Specificity (%)	Sensitivity (%)	Accuracy (%)	Precision (%)
	By considering two features			
GMM	73.81	71.11	71.09	72.01
SetSVM	95.38	71.30	89.63	85.44
CNN	96.00	85.94	93.46	91.80
Deep ResNet	96.00	82.55	92.25	90.26
Proposed AS-Jaya-based Deep RNN	**96.00**	**90.40**	**93.65**	**93.34**
	By considering three features			
GMM	73.81	71.11	71.09	72.01
SetSVM	96.00	71.30	91.59	86.29
CNN	96.00	95.00	93.46	94.82
Deep ResNet	96.00	87.68	92.90	92.19
Proposed AS-Jaya-based Deep RNN	**96.00**	**95.00**	**93.65**	**94.88**

and Deep ResNet are 73.41%, 96%, 96%, and 96%, respectively, while the specificity of the proposed AS-Jaya-based Deep RNN is 96%. When training percentage is 40%, the sensitivity obtained by the existing GMM, SetSVM, CNN, and Deep ResNet is 70.37%, 70.56%, 84.90%, and 80.18%, respectively, while the sensitivity of the proposed AS-Jaya-based Deep RNN is 85.08%.

When training percentage is 70%, the accuracy obtained by the existing GMM, SetSVM, CNN, and Deep ResNet is 70.91%, 91.01%, 93.28%, and 92.59%, respectively, while the sensitivity of the proposed AS-Jaya-based Deep RNN is 93.46%. When training percentage is 60%, the precision of the methods such as GMM, SetSVM, CNN, and Deep ResNet, and the proposed AS-Jaya-based Deep RNN are 71.67%, 85.77%, 93.94%, 91.26%, and 94.06%, respectively.

4.7 | Comparative discussion

Table 5 portrays the comparative discussion. The proposed AS-Jaya-based Deep RNN obtained the best results in 80% of training data. The proposed AS-Jaya-based Deep RNN obtained better performance for accuracy as 93.65%, specificity as 96%, and sensitivity as 95%, respectively, by considering three features. By considering two features, the proposed AS-Jaya-based Deep RNN obtained 96% for specificity, 90.40% for sensitivity, and 93.65% for accuracy using 80% training data, respectively, which is better than the existing methods, such as GMM, SetSVM, CNN, and Deep ResNet. The accuracy of the proposed AS-Jaya-based Deep RNN shows 24.19%, 4.29%, 0.20%, and 1.49%, better performance than the existing methods such as GMM, SetSVM, CNN, and Deep ResNet, respectively, by considering the two features. While considering the three features, the proposed AS-Jaya-based Deep RNN is 24.19%, 2.20%, 0.20%, and 0.8%, better accuracy than the existing methods such as GMM, SetSVM, CNN, and Deep ResNet, respectively. The precision of the proposed AS-Jaya-based Deep RNN is 22.85%, 8.46%, 1.65%, and 3.30%, better than the existing methods, by considering the two features. By considering the three features, the proposed AS-Jaya-based Deep RNN is 24.10%, 9.05%, 0.06%, and 2.83%, better than the existing methods such as GMM, SetSVM, CNN, and Deep ResNet, respectively.

TABLE 6 Computational complexity

Method	Time (s)
GMM	11.87
SetSVM	9.98
CNN	8.21
Deep ResNet	7.36
Proposed AS-Jaya-based Deep RNN	**6.08**

Table 6 depicts the computational complexity of the proposed AS-Jaya-based Deep RNN, with the existing methods such as GMM, SetSVM, CNN, and Deep ResNet, in which the proposed AS-Jaya-based Deep RNN has the minimum computation time of 6.08 s.

5 | CONCLUSION

A new optimization named AS-Jaya-based Deep RNN is modelled in this research to achieve the liver cancer detection process. The pre-processed result enhances the contrast of the image by removing the redundancies and noise in the original image. The segmentation module processes the image and transformed into segmented using BHEFC. The features are effectively extracted based on the pixel values and increase the efficiency of the detection rate. The proposed AS-Jaya optimization inherits the characteristic features from ASO and Jaya and effectively updates the weight of the classifier. The fitness measure is evaluated based on the position of an atom in order to obtain the best optimal solution such that the function with the minimal error value is accepted as the optimal solution. The proposed detection model outperforms the existing liver cancer detection techniques with respect to the performance of the segmentation result. However, the proposed AS-Jaya-based Deep RNN attained better performance with the values of accuracy as 93.64%, and specificity as 96%, sensitivity as 95%, and precision as 94.88%, respectively, by considering three features at 80% training data. Some of the real-time applications of this method are Cancer Epidemiology, Cancer Immunology Research, Cancer Discovery, Molecular Cancer Research, and Clinical Cancer Research. However, this method has the difficulty of operating in artificial computing and fictional computing platforms. In future, the performance of cancer detection will be enhanced using any other optimization algorithm that operates in artificial computing and fictional computing platforms.

REFERENCES

1. Vito Racanelli, Barbara Rehermann: The liver as an immunological organ, Hepatology 43(1), S54–S62 (2006)
2. Guogang, C., et al.: Primary liver cancer early screening based on gradient boosting decision tree and support vector machine. In: 2019 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), Shanghai, China (2019)
3. Petrick, J.L., et al.: International trends in liver cancer incidence, overall and by histologic subtype, 1978–2007. Int. J. Cancer 139(7), 1534–1545 (2016)
4. Doğantekin, A., et al.: A novel approach for liver image classification: PH-C-ELM. Measurement 137, 332–338 (2019)
5. Chuang, S.-C., et al.: Liver cancer: descriptive epidemiology and risk factors other than HBV and HCV infection. Cancer Lett. 286(1), 9–14 (2009)
6. Priya, B.L., et al.: A modified framework for multislice image fusion for high contrast liver cancer detection. IETE J. Res. 66(2), 139–149 (2018)
7. Liu, C., et al.: SetSVM: an approach to set classification in nuclei-based cancer detection. IEEE J. Biomed. Health Inform. 23(1), 351–361 (2018)
8. Chlebus, G., et al.: Deep learning based automatic liver tumor segmentation in CT with shape-based post-processing. In: MIDL 2018 Conference, Amsterdam (2018)
9. Bai, X.R., et al.: Accurate clinical diagnosis of liver cancer based on simultaneous detection of ternary specific antigens by magnetic induced mixing surface-enhanced Raman scattering emissions. Anal. Chem. 91(4), 2955–2963 (2019)
10. Dutta, A., Dubey, A.: Detection of liver cancer using image processing techniques. In: International Conference on Communication and Signal Processing, Melmaruvathur, Tamil Nadu, (2019)
11. https://nihcc.app.box.com/v/DeepLesion/folder/50715173939. Accessed October 2019
12. Narayan, R.R., et al.: Peripheral circulating tumor DNA detection predicts poor outcomes after liver resection for metastatic colorectal cancer. Ann. Surg. Oncol., 26(6), 1824–1832 (2019)
13. Salwa Shamis Sulaiyam Al-Mazidi et al.: Implementation of fuzzy c means and snake model for brain tumor detection. In: 4th National Symposium on Engineering Final Year Projects (2014)
14. Vinolin, V.: Breast cancer detection by optimal classification using GWO algorithm. Multimed. Res. (MR) 2(2), 10–18 (2019)
15. Sadeque, Z.A., et al.: Automated detection and classification of liver cancer from CT images using HOG-SVM model. In: Proceeding of 5th International Conference on Advances in Electrical Engineering (ICAEE), Dhaka, Bangladesh (2019)
16. Benny, R, Thomas, T.: Automatic detection and classification of liver lesions from CT-scan images. In: Proceedings of Fifth International Conference on Advances in Computing and Communications, Kochi, India (2015)
17. Anisha, P.R., et al.: A pragmatic approach for detecting liver cancer using image processing and data mining techniques. In: 2015 International Conference on Signal Processing and Communication Engineering Systems, Vijayawada, India, pp. 352–357 (2015)
18. Özyurt, F., et al.: A novel liver image classification method using perceptual hash-based convolutional neural network. Arabian J. Sci. Eng. 44(4), 3173–3182 (2019)
19. Nugroho, H.A., et al.: Contrast enhancement for liver tumor identification. In MICCAI Workshop 41(43), 201 (2008)
20. Sun, C., Guo, S., et al.: Automatic segmentation of liver tumors from multiphase contrast-enhanced CT images based on FCNs. Artif. Intell. Med. 83, 58–66 (2017)
21. Cohen, A.B., et al.: Fully convolutional network and sparsity based dictionary learning for liver lesion detection in CT examinations. Neurocomputing 275, 1585–1594 (2018)
22. Das, A., et al.: Deep learning based liver cancer detection using watershed transform and Gaussian mixture model techniques. Cogn. Sys. Res. 54, 165–175 (2019)
23. George, A., Rajakumar, B.R.: On hybridizing fuzzy min max neural network and firefly algorithm for automated heart disease diagnosis. In: Fourth International Conference on Computing, Communications and Networking Technologies, Tiruchengode, India (2013)
24. Lu, F., et al.: Automatic 3D liver location and segmentation via convolutional neural network and graph cut. Int. J. Comp. Assist. Radiol. Surg. 12(2), 171–182 (2017)
25. Hu, Z., et al.: Deep learning for image based cancer detection and diagnosis – a survey. Pattern Recognit. 83, 134–149 (2018)
26. Ouhmich, F., et al.: Liver tissue segmentation inmultiphase CT scans using cascaded convolutional neural networks. Int. J. Comput. Assist. Radiol. Surg. 14, 1275–1284 (2019)
27. Priya, B.L., et al.: A modified framework for multislice image fusion for high contrast liver cancer detection. IETE J. Res. 1–11 (2018)
28. Das, A., et al.: Detection of liver cancer using modified fuzzy clustering and decision tree classifier in CT images. Math. Method Pattern Recognit. 29, 201–211 (2019)
29. Wang, M., et al.: Catalytic hairpin assembly-programmed formation of clickable nucleic acids for electrochemical detection of liver cancer related short gene. Anal. Chim. Acta 1045, 1–8 (2018)
30. Bi, L., et al.: Automatic liver lesion detection using cascaded deep residual networks. Comput. Vision Pattern Recognit. (2017). https://deepai.org/publication/automatic-liver-lesion-detection-using-cascaded-deep-residual-networks
31. Wang, J.,et al.: A High Throughput Cantilever Array Sensor for Multiple Liver Cancer Biomarkers Detection. IEEE Sens. J. 16(12), 4675–4682 (2016)
32. Gruber, N., et al.: A Joint Deep Learning Approach for Automated Liver and Tumor Segmentation. In: 2019 13th International Conference on Sampling Theory and Applications (SampTA), Bordeaux, France, pp. 1–5 (2019)
33. Kumar, P.K., Bharathi, R.: Edge Prediction Based Segmentation and Vector Distance Based Classification Techniques for Liver Tumor Detection. Sens. Lett. 16, 614–623 (2018)
34. Nagai, H., Hak Kim, Y.: Cancer prevention from the perspective of global cancer burden patterns. J. Thorac. Dis. 9(3), 448–451 (2017)
35. Liu, J., et al.. Black hole entropic fuzzy clustering. IEEE Trans. Syst. Man Cybern. Syst. Humans 48(9), 1622–1636 (2017)
36. Mou, L., et al.: Deep recurrent neural networks for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 55(7), 3639–3655. (2017)
37. Zhao, W., et al.: Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. Knowl. Based Syst. 163, 283–304 (2019)
38. Rao, R.V.: Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int. J. Ind. Eng. Comp. 7(1), 19–34 (2016)
39. Lu, H., et al.: Automatic gender recognition based on pixel-pattern-based texture feature. J. Real-Time Image Process. 3(1-2), 109–116 (2008)
40. Inoue, M., et al.: Deep recurrent neural network for mobile human activity recognition with high throughput. Artif. Life Robot. 23(2), 173–185 (2018)
41. MATLAB. 9.7.0.1190202 (R2019b). The MathWorks Inc., Natick, MA (2018)

How to cite this article: Navaneethakrishnan M, Vairamuthu S, Parthasarathy G, Cristin R. Atom search-Jaya-based deep recurrent neural network for liver cancer detection. *IET Image Process*. 2021;15:337–349. https://doi.org/10.1049/ipr2.12019

APPENDIX

$$b_n^t(g+1) = b_n^t(g) + m_{1n}^t\left(b_{bst,n}^t - b_n^t(g)\right) - m_{2n}^t\left(b_{wst,n}^t - b_n^t(g)\right) \tag{A.1}$$

$$b_n^t(g+1) = b_n^t(g) + m_{1n}^t b_{bst,n}^t - m_{1n}^t b_n^t(g) - m_{2n}^t b_{wst,n}^t + m_{2n}^t b_n^t(g) \tag{A.2}$$

$$b_n^t(g+1) = b_n^t(g)\left[1 - m_{1n}^t + m_{2n}^t\right] + m_{1n}^t b_{bst,n}^t - m_{2n}^t b_{wst,n}^t \tag{A.3}$$

$$b_n^t(g) = \frac{b_n^t(g+1) - m_{1n}^t b_{bst,n}^t + m_{2n}^t b_{wst,n}^t}{\left[1 - m_{1n}^t + m_{2n}^t\right]} \tag{A.4}$$

$$b_n^t(g+1) = \frac{b_n^t(g+1) - m_{1n}^t b_{bst,n}^t + m_{2n}^t b_{wst,n}^t}{\left[1 - m_{1n}^t + m_{2n}^t\right]} + rand_n^t\, c_n^t(g) + z_n^t(g) \tag{A.5}$$

$$b_n^t(g+1) = \frac{b_n^t(g+1)}{\left[1 - m_{1n}^t + m_{2n}^t\right]} + \frac{m_{2n}^t b_{wst,n}^t - m_{1n}^t b_{bst,n}^t}{\left[1 - m_{1n}^t + m_{2n}^t\right]} + rand_n^t\, c_n^t(g) + z_n^t(g) \tag{A.6}$$

$$b_n^t(g+1) - \frac{b_n^t(g+1)}{\left[1 - m_{1n}^t + m_{2n}^t\right]} = \frac{m_{2n}^t b_{wst,n}^t - m_{1n}^t b_{bst,n}^t}{\left[1 - m_{1n}^t + m_{2n}^t\right]} + rand_n^t\, c_n^t(g) + z_n^t(g) \tag{A.7}$$

$$b_n^t(g+1)\left[1 - \frac{1}{\left[1 - m_{1n}^t + m_{2n}^t\right]}\right] = \frac{m_{2n}^t b_{wst,n}^t - m_{1n}^t b_{bst,n}^t}{\left[1 - m_{1n}^t + m_{2n}^t\right]} + rand_n^t\, c_n^t(g) + z_n^t(g) \tag{A.8}$$

$$b_n^t(g+1)\left[\frac{1 - m_{1n}^t + m_{2n}^t - 1}{1 - m_{1n}^t + m_{2n}^t}\right] = \frac{m_{2n}^t b_{wst,n}^t - m_{1n}^t b_{bst,n}^t}{\left[1 - m_{1n}^t + m_{2n}^t\right]} + rand_n^t\, c_n^t(g) + z_n^t(g) \tag{A.9}$$

$$b_n^t(g+1)\left[\frac{m_{2n}^t - m_{1n}^t}{1 - m_{1n}^t + m_{2n}^t}\right] = \frac{m_{2n}^t b_{wst,n}^t - m_{1n}^t b_{bst,n}^t}{\left[1 - m_{1n}^t + m_{2n}^t\right]} + rand_n^t\, c_n^t(g) + z_n^t(g) \tag{A.10}$$

$$b_n^t(g+1) = \frac{1 - m_{1n}^t + m_{2n}^t}{m_{2n}^t - m_{1n}^t} \times \left[\frac{m_{2n}^t b_{wst,n}^t - m_{1n}^t b_{bst,n}^t}{\left[1 - m_{1n}^t + m_{2n}^t\right]} + rand_n^t\, c_n^t(g) + z_n^t(g)\right]. \tag{A.11}$$