

# A New Frog Leaping Algorithm-oriented Fully Convolutional Neural Network for Dance Motion Object Saliency Detection

Yin Lyu<sup>1</sup> and Chen Zhang<sup>2</sup>

<sup>1</sup> Music College, Huaiyin Normal University  
Huaian City, 223001 China  
8201711037@hytc.edu.cn

<sup>2</sup> College of Sports Art, Harbin Sport University  
Harbin City, 150000 China  
sarkozyteague@foxmail.com

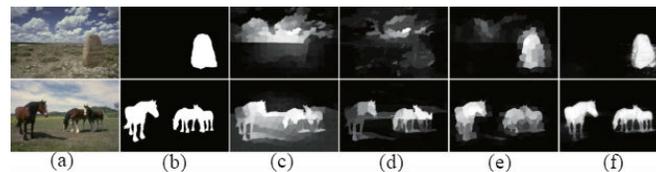
**Abstract.** Image saliency detection is an important research topic in the field of computer vision. With the traditional saliency detection models, the texture details are not obvious and the edge contour is not complete. The accuracy and recall rate of object detection are low, which are mostly based on the manual features and prior information. With the rise of deep convolutional neural networks, saliency detection has been rapidly developed. However, the existing saliency methods still have some common shortcomings, and it is difficult to uniformly highlight the clear boundary and internal region of the whole object in complex images, mainly because of the lack of sufficient and rich features. In this paper, a new frog leaping algorithm-oriented fully convolutional neural network is proposed for dance motion object saliency detection. The VGG (Visual Geometry Group) model is improved. The final full connection layer is removed, and the jump connection layer is used for the saliency prediction, which can effectively combine the multi-scale information from different convolution layers in the convolutional neural network. Meanwhile, an improved frog leaping algorithm is used to optimize the selection of initial weights during network initialization. In the process of network iteration, the forward propagation loss of convolutional neural network is calculated, and the anomaly weight is corrected by using the improved frog leaping algorithm. When the network satisfies the terminal conditions, the final weight is optimized by one frog leaping to make the network weight further optimization. In addition, the new network can combine high-level semantic information and low-level detail information in a data-driven framework. In order to preserve the unity of the object boundary and inner region effectively, the fully connected conditional random field (CRF) model is used to adjust the obtained saliency feature map. In this paper, the precision recall (PR) curve, F-measure, maximum F-measure, weighted F-measure and mean absolute error (MAE) are tested on six widely used public data sets. Compared with other most advanced and representative methods, the results show that the proposed method achieves better performance and it is superior to most representative methods. The presented method reveals that it has strong robustness for image saliency detection with various scenes, and can make the boundary and inner region of the saliency object more uniform and the detection results more accurate.

**Keywords:** Image saliency detection, dance motion, deep convolutional neural network, frog leaping algorithm, fully connected conditional random field.

## 1. Introduction

Saliency object detection is the most eye-catching object or region in the image [1]. The result is usually represented by a grayscale image, and the grayscale value of each pixel in the image indicates the probability that the pixel belongs to a saliency object. Saliency object detection has become an important preprocessing step in many computer vision applications, including image and video compression [2], image relocation [3], video tracking [4] and robot navigation [5], etc.

Although the detection performance of the saliency object detection method has been significantly improved, there are still some bottlenecks to be broken through in the computer vision task. Traditional saliency object detection methods focus on the low-level features in the manually selected images, and use a variety of prior knowledge to calculate saliency, such as contrast prior [6], center prior [7], background prior and object prior [8]. However, the detection effect of these models is not satisfactory in practical problems. For example, it is difficult to detect foreground objects when background and foreground objects share some similar visual features (see line 1 in figure 1(c)(d)). In addition, detection may fail when multiple saliency objects partially or completely overlap each other (see line 2 of figure 1(c) (d)).



**Fig. 1.** Comparison with different methods. (a) input images; (b) ground truth; (c) non-deep learning1 method; (d) non-deep learning2 method; (e) deep learning method; (f) proposed

The convolutional neural network (CNN) based methods have successfully overcome the performance bottleneck of traditional manual feature selection methods in many computer vision tasks, such as image classification [9] and semantic segmentation [10], etc.. Similarly, the saliency detection methods based on CNN greatly improve the detection performance. The CNN-based models have demonstrated their advantages in feature extraction and can better capture the high-level semantic information of objects in the complex backgrounds, which can achieve better performance than traditional methods as shown in figure 1(e)(f).

Generally, each object can be represented by three different feature levels, namely low feature, intermediate feature, and high feature. Low-level features correspond to shallow features of deep convolutional networks, such as texture, color, and edge. Intermediate features are related to object shape and contour information, while high-level features are related to object semantic information. Though only using high-level semantic information can improve detection performance, other features levels are also important for detecting saliency objects. Therefore, it is a key and challenging problem to extract and fuse effective feature information of all levels in CNN model. A standard convolu-

tional neural network usually consists of repeated cascade convolutional layers. Deeper convolution layers encode semantic information at the expense of spatial resolution. The shallow convolution layer contains more detailed information about object structure but lacks global properties.

This paper proposes a simple but effective deep convolutional neural network model for saliency detection tasks. It can effectively combine multi-level features to capture unique high-level semantic information and shallow detail information simultaneously in complex images. Meanwhile, an improved frog leaping algorithm is used to optimize the selection of initial weights during network initialization. The new deep network consists of a feature extraction module and a feature fusion module. The feature extraction module can not only generate effective high-level semantic features at different scales, but also capture subtle visual contrast features between low-level and intermediate feature maps for accurate saliency detection. The main contributions of this paper are as follows:

1. A new deep convolutional network based on fully convolutional networks (FCN) is proposed for saliency object detection, which can effectively learn rich multi-scale and multi-level features from complex background images. The model can learn the global and local features of images and avoid the interference of irrelevant background information.
2. An improved frog leaping algorithm is used to optimize the selection of initial weights during network initialization. In the process of network iteration, the forward propagation loss of convolutional neural network is calculated, and the anomaly weight is corrected by using the improved frog leaping algorithm. When the network satisfies the terminal conditions, the final weight is optimized by one frog leaping to make the network weight further optimization.
3. A multi-scale feature fusion mechanism is introduced. The model combines the shallow and deep feature maps in the deep convolutional network, which significantly improves the detection performance and does not need to be supplemented by manually selected features.
4. According to the five commonly used evaluation indexes, the new method is tested on DUT-OMRON, ECSSD, SED2, HKU, PASCAL-S and SOD data sets to prove the effectiveness of the new method by quantitative and qualitative analysis.

Section 2 introduces the related works. In section 3, we give the detailed proposed model analysis. Experiments are conducted in section 4. There is a conclusion in section 5.

## 2. Related Works

With the continuous progress of science and technology, how to quickly search and locate the information that people are interested in from the massive data resources has become an important research content in the field of computer vision [11]. Visual saliency has been regarded as an important mechanism for processing information tasks in computer vision. Saliency object detection obtains the areas of interest in images by simulating visual saliency and ignores the areas of uninterest. Saliency object detection is widely used in image matching, image retrieval, image compression, image quality assessment, object recognition and object relocation.

According to feature selection methods, saliency detection can be divided into two categories: artificial feature selection and deep convolutional network-based feature extraction.

### 2.1. Artificial Feature Selection

Traditional saliency object detection methods usually use manually selected features at the pixel level. Most of these algorithms make computation based on local or global features, such as color, orientation and texture. Global-based methods estimate the saliency of each pixel or region by using global contrast and feature statistics. Li et al. [12] proposed an automated saliency object segmentation method based on context and shape prior, which obtained saliency images by continuously iterating update of multi-scale context information and shape prior. Shen et al. [13] proposed a saliency object detection method based on low-rank matrix recovery, and obtained saliency map by fusing low-level features and high-level prior information. Yang et al. [14] proposed a graph regularization saliency detection method based on convex hull center prior, and refined the primary saliency images calculated by contrast and center prior with the graph regularization method to obtain the final saliency images. Xie et al. [15] proposed a Bayesian saliency detection method based on low and medium level cues, it used Bayesian methods to fuse saliency information on medium and low level cues to obtain final saliency maps. Zhang et al. [16] proposed a new saliency detection method using prior information such as frequency, color and position to obtain saliency maps. Li et al. [17] proposed a saliency detection method based on background prior and foreground seed selection, using the fusion of background prior and foreground prior to obtain a saliency map. Piao et al. [18] proposed a saliency detection method based on cellular automata (Single-layer cellular automata method (SCA), Single-layer Cellular Automata Optimizing Background Map (BSCA), Multi-layer Cellular automata (MCA)). According to the updating criteria, the saliency value was repaired by the cellular automata mechanism to obtain the saliency map. Zhou et al. [19] proposed a detection method based on integral fusion compactness and local contrast, using the complementary properties of compactness and local contrast cues to obtain saliency images. Tang et al. [20] proposed a saliency object detection method based on weighted low-rank matrix recovery, using position, color and boundary connectivity to generate a high-level background prior map, which was integrated into the weighted matrix to obtain a saliency map. Although they are easy to implement, they lack geometric structure cues and semantic information, so contrast-based algorithms cannot uniformly detect complete saliency objects, nor can they effectively suppress messy backgrounds in complex images.

### 2.2. Deep Convolutional Network-based Feature Extraction

The traditional saliency object detection methods mainly rely on the low-level features of the image which are manually selected and cannot describe the deep semantic feature information. Therefore, saliency objects cannot be accurately detected in complex images. At present, deep neural network technology is widely used in computer vision tasks, which greatly improves the performance of models. For the saliency object detection task, the data-driven model aims to obtain the semantic information of saliency

objects directly from a set of training data with pixel-level tags in the supervised learning model. CNN-based saliency detection methods can be divided into two categories, based on superpixel segmentation and based on FCN. The former uses the superpixel as the basic unit to train the deep neural network to predict saliency. All the pixels located in the same superpixel enjoy the same saliency value in the final prediction map. Wang et al. [21] used CNN to calculate the saliency score for each pixel in the local context, and then fine-tuned the saliency score for each object region in the global content. Li et al. [22] predicted the saliency score of each superpixel by combining local context and global context simultaneously in multi-context CNN. Chen et al. [23] used global and local context information to integrate them into the backbone network based on deep convolutional network for saliency detection. However, these superpixel-based methods tend to deal with local regions alone and cannot effectively capture global information about saliency objects. In addition, they rely on the over-segmentation method, so the network must run many times to calculate the saliency value of all the superpixels in the image, which makes the algorithm very time-consuming. Finally, they ignore the important spatial context information because they simply assign saliency values to each superpixel. In practice, the context information of an image is very useful for saliency detection.

To overcome these shortcomings, researchers tend to use the FCN model to detect saliency targets in a pixel-to-pixel manner.

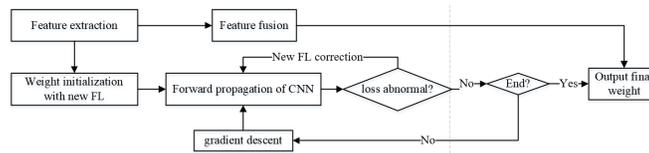
Wang et al. [24] used end-to-end convolutional neural networks to compute visual contrast information within an image. Liu et al. [25] designed a two-step deep network to obtain rough global predictions by autonomously learning globally saliency cues, and then used another network to further fine-tune the details of the prediction map by integrating local context information. On this basis, Li et al. [26] proposed saliency detection network with sharing features, and utilized a Traplus-regularized nonlinear regression model for saliency adjustment. Although these deep learning-based methods have made significant progress, the CNN-based model still can be improved, so that it can uniformly highlight the whole saliency object and retain accurate boundary information in complex images with messy backgrounds.

### 3. Proposed Saliency Object Model

The proposed saliency object detection algorithm in this paper mainly includes two steps: 1) multi-scale fully convolutional network. The weight values of the network are updated by the modified frog leaping algorithm, and the rich features from each convolution layers are extracted and fused. 2) Saliency update method. A fully connected conditional random field (CRF) model is used to update the prediction results to produce more refined saliency detection results.

#### 3.1. Multi-scale Full Convolutional Network

In order to design a FCN network that can learn the pixel-to-pixel saliency detection task, a multi-scale deep convolutional neural network is proposed to extract more multi-scale and multi-level feature information that is beneficial to saliency detection. The proposed neural network model is shown in figure 2 with three parts: weight update module, multi-scale feature extraction module and feature fusion module.

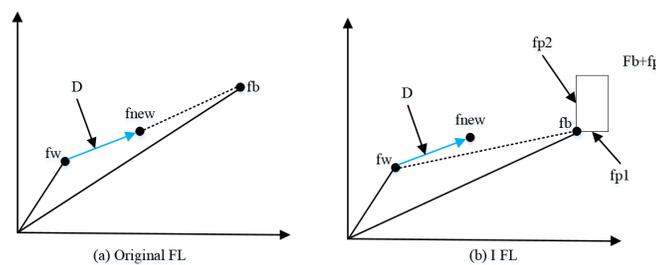


**Fig. 2.** The proposed saliency detection model

**Weight update by IFL in multi-scale deep convolutional neural network** Aiming at the complexity of saliency images, the improved frog leaping (IFL) algorithm is introduced into the weight initialization and weight update in convolutional neural networks [27]. By constantly updating the position of the worst frog, the global optimal frog is found as the initial weight of the network. In the process of network iteration, the calculated loss value each time is collected, and the network weight which produces abnormal loss value is corrected by the IFL.

When the generated weight by the network is too poor, the network needs to spend more time to perform gradient descent operation to correct the weight. So it is easy to make the network fall into local optimum and affect the final result. To solve this problem, the IFL is proposed to initialize the weight of convolutional neural network.

The traditional FL algorithm is a kind of sub-heuristic population evolutionary algorithm, which has excellent global search ability and can effectively calculate the global optimal solution. However, if we directly apply it to the weight initialization in this paper, it will generate a large number of calculations, and can increase the time cost, affect the efficiency. Therefore, an improved FL algorithm is proposed in this study. Different from the traditional FL, we no longer divide the initial samples into several populations, but regard all frogs as a population and directly conduct global optimization. And some improvements in the FL rule are made. The details of weight initialization algorithm and weight updating algorithm are shown in **Algorithm 1** and **Algorithm 2**.



**Fig. 3.** Improved frog leaping algorithm

The back propagation of traditional CNN is essentially the gradient descent method, which updates network parameters through the loss value calculated by forward propagation, so it can find the optimal solution. However, the saliency image background is complex and the targets overlap each other, so it is difficult to detect the saliency of the tar-

---

**Algorithm 1** Weight initialization algorithm

---

- 1: Step 1. Parameter initialization. Determine the frog population according to the Gaussian distribution formula:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right). \tag{1}$$

Where, we set  $\mu = 0, \sigma = 1$ .

- 2: Step 2. Ranking. All the frogs whose loss values are not calculated are brought into the CNN model.  $n$  images are randomly selected from the training image library as the reference images for forward propagation, and the loss value of each frog is calculated. Here, the loss value calculation function is the fitness function of the FL algorithm, and the loss calculation formula is:

$$loss = \sum_{k=1}^b \left(-\sum_{i=1}^n \sum_{j=1}^s t_{ij} \ln(p_{ij})\right). \tag{2}$$

Where  $p$  represents the output value of the network.  $t$  represents the real value.  $s$  represents the dimension of the saliency object label, and  $b$  represents the number of saliency targets to be detected at the same time. It ranks all frogs in ascending order according to their fitness functions.

- 3: Step 3. Searching and updating the location. The optimal frog  $f_b$  and the worst frog  $f_w$  can be obtained from Step 2. The position of the worst frog is updated by the position update function. For frog position updating, this study adds an offset to the traditional FL formula, and appropriately increases the random interval of  $rand()$  function. The formula is as follows:

$$D = (f_b + f_p - f_w) \times rand(0, 1, 2). \tag{3}$$

$$f_{new} = f_w + D. \tag{4}$$

$$f_{p_i} = \frac{rand(-1, 1) \times 0.0001}{\exp(f_b + f_a) + 1}. \tag{5}$$

Here,  $f_p$  represents the offset, whose dimension is the same as that of each frog.  $f_{p_i}$  represents the value on the  $i$ -th dimension of  $f_p$ .  $f_{new}$  represents the updated frog. By adding offset, the performance of FL can be improved effectively. Increasing the random interval makes it easy to find the optimal solution. The leaping way mentioned in this step is mapped to two-dimensional coordinate as shown in figure 3:

- 4: Step 4. Judging whether the algorithm meets the convergence condition. If YES, stop the algorithm and take the optimal frog value as the initial weight of the convolutional neural network. Otherwise, return to Step 3.
-

get. When gradient descent algorithm is used, it is easy to occur abnormal situation with a large range of loss value, which affects the efficiency and even causes the algorithm falling into the local optimum. In view of the above problems, the improved frog leaping algorithm is used to correct the poor gradient of the back propagation of convolutional neural network.

In the process of network iteration, the loss value in each forward propagation is calculated. If the absolute value of the difference between the loss value in the  $i$  time and  $i - 1$  time is greater than the threshold  $o$ , that is,  $|l_i - l_{i-1}| > o$ , it is considered that the weight in the  $i$ -th forward propagation is invalid. The IFL algorithm is used to re-find the optimal weight. After the convolutional neural network satisfies the end conditions, the trained weight FA is obtained, and the final algorithm performs the improved FL calculation again. The local optimization can be directly and effectively avoided by the last improved FL optimization.

**Feature extraction module** The multi-scale feature extraction module outputs feature maps with different resolutions from the sides of different convolution groups of the backbone network. The proposed model uses VGGNet-16 (Visual Geometry Group), which has been pre-trained for image classification in ImageNet data set as the backbone network. And it is modified to meet the requirements. It retains its 13 convolution layers and removes the fifth pooling layer and the fully connection layer. The modified VGGNet consists of five groups of convolution layers. For brevity, the third sub-layer in the fifth group of convolution layers is represented as conv5\_3, and other convolution layers in VGGNet are also represented by this method. For the input image  $256 \times 256$  pixels, the modified VGGNet-16 produces five feature maps  $f_1^a, f_2^a, \dots, f_5^a$ , the spatial resolution decreases according to stride 2. These feature maps are generated from (Conv1\_2, Conv2\_2,  $\dots$ , Conv5\_3) respectively. The feature map from Conv1\_2 has the maximum spatial resolution, while the feature map  $f_5^a$  from Conv5\_3 has the minimum spatial resolution.

**Feature fusion module** Different convolution layers usually produce different feature representations, ranging from low-level structural features to high-level semantic features. The shallow convolution layer contains rich details while the deep convolution layer contains rich semantic information but lacks spatial context information. Feature fusion module involves multi-scale convolution feature fusion.

For each feature graph  $f_i^a (i \in 1, 2, 3, 4, 5)$ , the feature graph  $f_i^b$  is obtained by a  $3 \times 3$  convolution layer and a  $5 \times 5$  convolution layer. It sets its channel number as the channel number output by the  $i$ -th side of VGGNet-16. Then, by using a single-channel  $1 \times 1$  convolution layer for dimension reduction, five feature graphs  $f_i^c (i \in 1, 2, 3, 4, 5)$  with sizes of  $256 \times 256$  pixels,  $128 \times 128$  pixels,  $64 \times 64$  pixels,  $32 \times 32$  pixels and  $16 \times 16$  pixels are obtained. In order to make these feature maps  $f_i^c$  have the same size as the input images, deconvolution operation and bilinear interpolation are used to up-sample the feature maps. The steps of the deconvolution layer in the five output layers are set to 1, 2, 4, 8 and 16 respectively. Then, these feature maps  $f_i^d$  with the same resolution are spliced together. Finally, a saliency prediction map S is generated through a  $1 \times 1$  convolution layer. In the training stage, stochastic Gradient Descent (SGD) method is used to minimize all training samples.

---

**Algorithm 2** Weight update algorithm
 

---

- 1: Step 1. Input a sample. Inputting the target image and the corresponding label, carry out one-hot coding for the label.
- 2: Step 2. Convolution, pooling. Multi-layer convolution and pooling operations are carried out on the input target image. After each pooling calculation, ReLU function is used to activate the output results.
- 3: Step 3. Fully connection calculation. The predicted value is obtained through multi-layer fully connection calculation. The predicted value is substituted into the Softmax() function for calculation. After obtaining the result, the loss value is calculated by formula (2).
- 4: Step 4. Error value comparison. Judging whether the absolute value difference between the loss value of this iteration and the previous loss value is greater than the threshold value. If YES, return to Step 5. Otherwise, go to Step 10.
- 5: Step 5. Parameter initialization. Recording the weight  $w_b$  of the  $(i - 1) - th$  forward propagation. Setting the frog number  $c$ .
- 6: Step 6. Generating frog. Different from the way of generating frogs by Gaussian distribution mentioned above, here the frogs are mainly generated based on , and the generation formula is as follows:

$$w_{ij} = w_{bj} + 0.01 \times rand(-1, 1). \quad (6)$$

Where,  $1 \leq i \leq c - 1$ ,  $w_{ij}$  represents the value in the  $j - th$  dimension of the generated  $i - th$  frog.  $n$  is the total number of frogs. In the frog colony, there is a frog  $w_g$ , which meets the conditions:  $l_g \leq l_b$ .  $l_g$  and  $l_b$  are the loss values of  $w_g$  and  $w_b$ , respectively.

- 7: Step 7. Ranking. A small number of training images are randomly selected as input values, and all frogs in  $c$  are brought into the CNN model. The loss value is calculated according to formula (2). All frogs are sorted in descending order by loss value.
  - 8: Step 8. Searching and location update. This step is the same as step 3 in Algorithm 1.
  - 9: Step 9. Checking the frog leaping stop conditions. Whether the algorithm meets the convergence condition. If so, stop the algorithm and update the network weight with the value of the optimal frog; otherwise, return to Step 8.
  - 10: Step 10. Check the network stop conditions. Whether the network meets the convergence condition. If Yes, stop the iteration; otherwise, go to Step 1.
  - 11: Step 11. Optimizing the final weights. After trained the algorithm, the weights in the network are the final weights, and the final weights are taken as the initial frog  $w_b$ . The frog swarm is generated by formula (6), and then steps 7, 8 and 9 are successively executed to finally obtain the global optimal frog  $w_{qb}$ , which is the final trained weight value of the algorithm.
-

In order to simulate the spatial correlation of the whole image and reduce the amount of computation, a network structure based on fully convolution is used. The fully convolution operation has the ability to share the convolution features through the whole image, thus reducing feature redundancy and making the full convolution network model simple and effective.

### 3.2. Optimization of Spatial Continuity

The proposed saliency detection algorithm using multi-level features can accurately locate the saliency objects and effectively suppress the complex background. However, the saliency prediction map of the proposed multi-scale fully convolutional network is relatively rough, and the contour information of saliency objects is not well retained. In order to improve the spatial continuity of the detection results, the proposed algorithm additionally uses the fully connected CRF method [28] to update the saliency map of the network pixel by pixel during the test phase. This method solves the problem of binary pixel label allocation and uses the following energy function for calculation, i.e.

$$E(L) = - \sum_i \log P(l_i) + \sum_{i,j} \theta_{ij}(l_i, l_j). \quad (7)$$

Where,  $i$  and  $j$  represent the horizontal and vertical coordinates of pixels in the image respectively.  $L$  represents the binary labels of all pixels.  $P(l_i)$  is the probability of pixel  $x_i$  with label  $l_i$ , which indicates the possibility that pixel  $x_i$  belongs to the saliency object.

Initially,  $P(1) = S_i$ ,  $P(0) = 1 - S_i$ , where  $S_i$  is the saliency score at pixel  $x_i$  of the fused saliency graph  $S$ , that is, the binary potential function  $\theta_{ij}(l_i, l_j)$  is defined as:

$$\theta_{ij} = \mu(l_i, l_j) \left[ \omega_1 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\alpha^2}\right) - \left(-\frac{\|I_i - I_j\|^2}{2\sigma_\beta^2}\right) + \omega_2 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\gamma^2}\right) \right]. \quad (8)$$

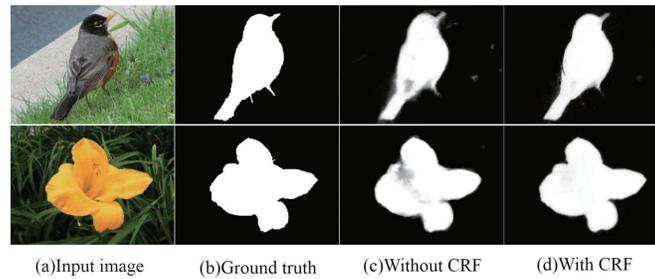
In the formula,  $p_i$  and  $I_i$  represent the position of pixel  $x_i$  and pixel value respectively.  $\omega_1$ ,  $\omega_2$ ,  $\sigma_\alpha$ ,  $\sigma_\beta$  and  $\sigma_\gamma$  are the weights. If  $l_i \neq l_j$ , then  $\mu(l_i, l_j) = 1$ .  $\theta_{ij}$  contains two convolution kernels. The first convolution kernel depends on pixel position  $p$  and pixel intensity  $I$ . This convolution kernel enables adjacent pixels with similar colors to have similar saliency fractions. The second convolution kernel is used to remove small isolated regions.

As shown in figure 4, the fused saliency graph of the multi-scale fully convolutional network without CRF is rough and cannot uniformly display the internal region of the saliency object, while the saliency graph updated by CRF well retains the contour of the saliency object and uniformly highlights the whole saliency object.

## 4. Experiments and Analysis

### 4.1. Data Set

In order to evaluate the performance of the proposed algorithm, a series of subjective and objective experiments are carried out on six benchmark data sets. These datasets have



**Fig. 4.** Comparison of saliency detection results with and without CRF

pixel-level labels, including DUT-OMRON, ECSSD (Extended Complex Scene Saliency Dataset), SED2, HKU, PASCAL-S and SOD (Saliency objects dataset) [29]. The HKU is a large data set with over 4000 challenging images, most of which have low contrast and multiple saliency objects. The DUT-OMRON includes 5168 images with one or more saliency objects and relatively complex backgrounds. ECSSD contains 1000 semantically meaningful and complex images. PASCAL-S contains 850 real-world images selected from a PASCAL-VOC data set with 20 object classes. SED2 is a multi-object data set that typically contains two saliency objects per image. SOD consists of 850 images containing one or more objects with a cluttered background. In contrast, the HKU, PASCAL-S and SOD datasets are more challenging due to the presence of multiple saliency objects in their images and the complex background.

#### 4.2. Evaluation Index

In this section, five commonly evaluation indicators are used to measure the performance of the proposed algorithm, including precision recall (PR) curve, F-measure, Max F-measure (maxF), weighted F-measure (wF) and mean absolute error (MAE) [30].

PR curve: Precision refers to the percentage of positive samples in all data predicted to be positive samples. Recall rate refers to the proportion of the data predicted as true samples to all positive samples. The saliency feature map is segmented with a fixed threshold value ranging from 0 to 255. A pair of accuracy-recall fractions are calculated to form PR curves to describe the performance of the algorithm under different conditions.

F-measure and maximum F-measure: F-measure is a comprehensive quantitative index of PR, which is calculated as,

$$F_{\beta} = \frac{(1 + \beta^2) \cdot P \cdot R}{\beta^2 \cdot P + R}. \quad (9)$$

Where,  $\beta$  is the balance parameter.  $P$  is precision, and  $R$  is the recall rate. In this paper,  $\beta^2$  is set to 0.3 to improve the proportion of important precision. The threshold is set to twice the average saliency of the entire image. The  $maxF$  is defined as the maximum F-measure calculated using the PR curve.

Weighted F measure (wF). This index is a weighted version of the F measure, which corrects the interpolation, dependence, and equal-importance defects of the F measure.

Similar to the F measure, the weighted F measure is calculated by the weighted harmonic average of the weighted precision  $P^w$  and the weighted recall rate  $R^w$ , i.e.,

$$F^w = \frac{(1 + \beta^2) \cdot P^w \cdot R^w}{\beta^2 \cdot P^w + R^w}. \quad (10)$$

Mean Absolute Error (MAE) is used to measure the mean error, which is defined as the absolute error of the average pixel between the truth graph and the predicted saliency graph.

$$M = \frac{1}{h \times w} \sum_{i=1}^h \sum_{j=1}^w |S_{ij} - G_{ij}|. \quad (11)$$

Where  $S$  represents saliency graph,  $G$  represents truth graph.  $h$  and  $w$  represent the height and width of the image.

### 4.3. Implementation Details

The experiment environment in this paper is: Windows7 64-bit operating system, Intel(R) Core(TM) i5-4210U, CPU 1.7GHz processor, Memory 4 GB, 32 RAM, MATLAB R2017a, NVIDIA 1060T GPU.

The proposed network is implemented on the an open source framework (Caffe). More specifically, the pre-trained VGGNet-16 network model is used and modified in the feature extraction module, and the parameters of the convolution layer are randomly initialized in the feature fusion module. The entire network is fine-tuned on the MSRA-B dataset to achieve the task of pixel-to-pixel saliency detection. MSRA-B is a public data set containing 5000 test images. The resolution of all test images and truth images is adjusted to  $256 \times 256$  pixels for training, and only one image is loaded each time. The learning rate is set to  $10^{-9}$ . Weight attenuation is 0.0005. The momentum is 0.9. Loss weight of each side output is 1. In addition, the weights of fusion layer are all initialized to 0.2 in the training stage. The stochastic gradient descent is adopted for network learning.

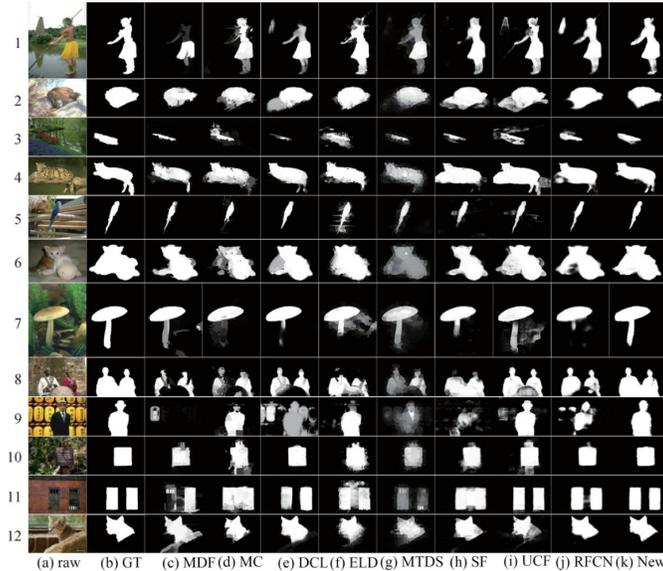
The CRF parameters in this paper are determined by using the cross validation method on the verification data set ECSSD. In the experiments,  $\omega_1 = 3.0$ ,  $\omega_2 = 1.0$ ,  $\sigma_\alpha = 8.0$ ,  $\sigma_\beta = 60.0$ ,  $\sigma_\gamma = 5.0$ .

### 4.4. Comparison of Performance

The proposed algorithm in this paper is compared with 14 saliency object detection methods, including RFCN (Recurrent Fully convolutional network) [31], PAGR (Progressive Attention Guided Recurrent Network) [32], UCF (Uncertain convolutional Features) [33], SF (Supervision by Fusion) [34], DCL (Deep Contrast Learning) [35], MC (Multi-context Deep Learning) [36], MTDS (Multi-Task Deep Neural Network) [26], ELD (Encoded Low Level Distance Map and High Level Features) [37], LEGS (Local Estimation and Global Search) [21], MDF (Multi-scale deep CNN Features) [38], KSR (Kernelized Subspace Ranking) [39], DRFI (Discriminative Regional Feature Integration) [40], SMD (Structured Matrix Decomposition) [41] and RR (Regularized Random Walks ranking) [42]. For equitable comparison, the saliency results of compared model are provided by the authors.

Here, RFCN, UCF, MTDS, ELD, DCL, SF, MC, LEGS, MDF and KSR are based on deep learning.

### A. Subjective comparative analysis



**Fig. 5.** Visual comparison results with different models on ECSSD dataset

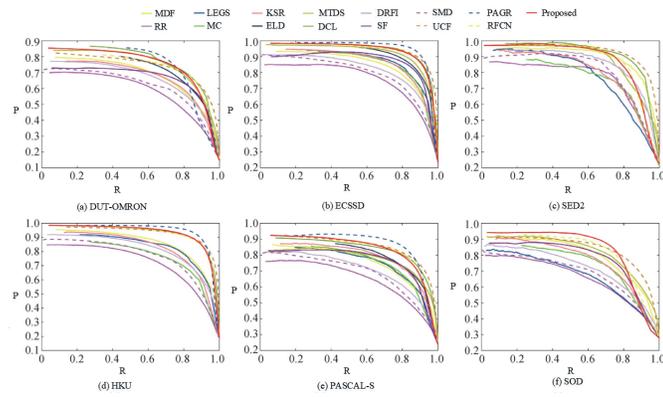
Figure 5 shows a visual comparison of saliency images generated by different algorithms on six data sets. Experimental results show that the proposed method can deal with all kinds of complex images better, it not only can display the whole saliency object evenly, but also can retain the contour of saliency object in various scenes. For example, foreground objects and background low contrast (2 row, 4 row and 12 row in figure 5), the image boundary of saliency objects (88 row and 97 row), several saliency objects (6 row and 8 row), with complex texture and structure of the saliency objects (1 row, 3 row, 8 row and 11 row), and background clutter (5 row, 6 row, 7 row, and 10 row), etc.

### B. Objective comparative analysis

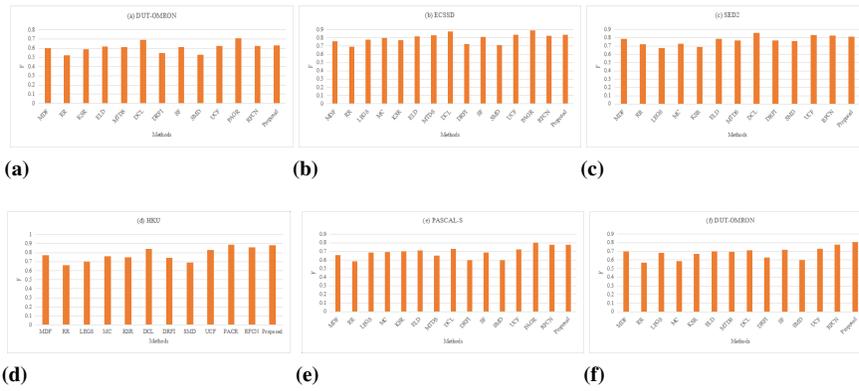
For quantitative evaluation, figure 6 shows the PR curves of the proposed method and 14 representative algorithms on six benchmark data sets. It can be seen that: 1) the saliency object detection method based on FCN is superior to other methods; 2) The new method in this paper is competitive in ECSSD, DUT-OMRON, HKU, PASCAL-S and SOD data sets, but slightly inferior to MTDS, DCL, UCF and PAGR algorithms.

In addition, the F and wF scores of the proposed model are compared with these existing methods on six benchmark data sets. The results are shown in figures 7 and 8. The results of MAE and the maxF are shown in table 1.

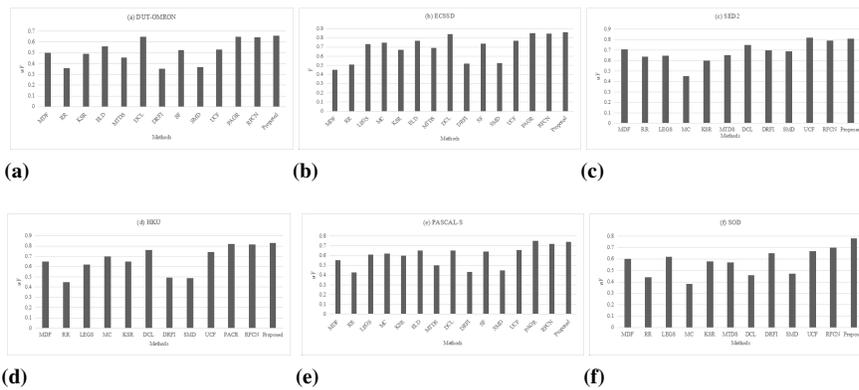
It can be seen that the performance of the proposed method is worse than that of UCF on SED2, because most of the images in SED2 contain two separate small-size objects, while the new method does not introduce corresponding modules to deal with this situa-



**Fig. 6.** PR curves of saliency maps produced by different approaches on six datasets



**Fig. 7.** F-measure scores of saliency maps with different approaches on six datasets



**Fig. 8.** Weighted F-measure scores of saliency maps with different approaches on six datasets

**Table 1.** MAE and maxF scores of saliency maps with different approaches on six datasets

Data	DUT-OMRON	DUT-OMRON	ECSSD	ECSSD	HKU	HKU
Index	MAE	maxF	MAE	maxF	MAE	maxF
RR	0.195	0.628	0.194	0.755	0.183	0.723
SMD	0.177	0.635	0.184	0.770	0.166	0.753
DRFI	0.160	0.674	0.181	0.793	0.155	0.788
LEGS	0.132	0.637	0.129	0.838	0.141	0.758
MDF	0.102	0.705	0.116	0.842	0.123	0.871
MC	0.114	0.805	0.111	0.847	0.102	0.818
DCL	0.090	0.767	0.078	0.911	0.074	0.901
ELD	0.101	0.716	0.089	0.878	0.093	0.852
MTDS	0.131	0.756	0.132	0.893	0.104	0.893
KSR	0.141	0.689	0.143	0.840	0.131	0.803
SF	0.118	0.695	0.098	0.863	0.099	0.876
UCF	0.131	0.740	0.080	0.914	0.073	0.898
PAGR	0.082	0.782	0.072	0.938	0.059	0.929
RFCN	0.088	0.744	0.076	0.793	0.065	0.894
Proposed	0.073	0.758	0.061	0.897	0.049	0.887
Data	SED2	SED2	PASCAL-S	PASCAL-S	SOD	SOD
Index	MAE	maxF	MAE	maxF	MAE	maxF
RR	0.184	0.795	0.237	0.663	0.270	0.657
SMD	0.169	0.831	0.217	0.699	0.245	0.690
DRFI	0.156	0.846	0.218	0.704	0.235	0.714
LEGS	0.152	0.763	0.170	0.756	0.206	0.747
MDF	0.134	0.670	0.153	0.769	0.235	0.713
MC	0.126	0.818	0.153	0.754	0.272	0.670
DCL	0.115	0.887	0.125	0.822	0.233	0.698
ELD	0.129	0.796	0.131	0.789	0.229	0.704
MTDS	0.145	0.884	0.186	0.773	0.201	0.795
KSR	0.163	0.792	0.165	0.778	0.208	0.755
SF	0.125	0.804	0.141	0.770	0.167	0.780
UCF	0.085	0.898	0.126	0.827	0.158	0.817
PAGR	0.096	0.813	0.101	0.870	0.156	0.801
RFCN	0.119	0.876	0.141	0.827	0.155	0.819
Proposed	0.091	0.857	0.098	0.823	0.123	0.834

tion. The new method is slightly worse than PAGR algorithm on six data sets. The possible reasons are as follows: 1) PAGR algorithm uses VGG-19 model with higher accuracy as the backbone network; 2) The PAGR algorithm introduces the recurrent network module, which can better detect the saliency from coarse to fine. Comprehensive analysis for the experimental results shows that the proposed method has certain advantages in processing complex scene images, and its performance is close to the true value.

#### 4.5. Algorithm Analysis

##### A. Feature graph analysis

In the proposed multi-scale detection network, the proposed method extracts five feature graphs  $f_i^d$  from the modified VGGNet-16 and fuses them in the connection layer. On the ECSSD benchmark data set, each feature graph  $f_i^d$  is compared with the fused feature graph, and the results are shown in table 2. It can be seen from table 2 that: 1) The performance of the feature graph  $f_5^d$  from the deepest convolution layer edge is closer to that of the fused feature graph; 2) The saliency feature map obtained by combining multi-layer features is better than the single feature map.

**Table 2.** Comparison of feature maps from different side output

Feature map	maxF	F	wF	MAE
$f_1^d$	0.345	0.097	0.223	0.491
$f_2^d$	0.317	0.174	0.172	0.319
$f_3^d$	0.501	0.469	0.298	0.318
$f_4^d$	0.427	0.302	0.235	0.243
$f_5^d$	0.871	0.835	0.724	0.105

##### B. Feature map fusion analysis

In order to verify the effectiveness of the proposed combination scheme, these features are combined in different ways and expressed as:  $S_1 = f_5^d$ ,  $S_2 = \sum_{i=4}^5 f_i^d$ ,  $S_3 = \sum_{i=3}^5 f_i^d$ ,  $S_4 = \sum_{i=2}^5 f_i^d$ . The training set and the used hyperparameters are consistent with the model in this paper. The evaluation results on the ECSSD data set are shown in table 3. It can be seen that the proposed method in this article obtains a better performance.

**Table 3.** Comparison with different fusion ways

Method	maxF	F	wF	MAE
S1	0.876	0.814	0.758	0.081
S2	0.862	0.797	0.743	0.088
S3	0.855	0.795	0.724	0.093
S4	0.879	0.812	0.758	0.079
Proposed	0.877	0.834	0.779	0.077

### C. Effect of IFL on the detection result

IFL is used for searching the optimization weight in the network. So we test the effect of IFL on the proposed method, the results are shown in table 4. The results show that the IFL can greatly improve the saliency detection.

**Table 4.** Effect of IFL on the detection result

Data set	Method	maxF	F	wF	MAE
DUT-OMPON	Without IFL	0.763	0.802	0.711	0.082
DUT-OMPON	With IFL	0.879	0.851	0.792	0.071
HKU	Without IFL	0.792	0.784	0.697	0.079
HKU	With IFL	0.881	0.845	0.785	0.068

### D. Effect of CRF on the detection result

As a post-processing step, the CRF method updates the saliency map obtained from the network to further highlight the consistency of the region inside the saliency object and retain the accurate contour information of the saliency object.

To verify its effectiveness, maxF, F, wF and MAE scores are used to evaluate the performance of the saliency method with/without CRF, and the results are shown in table 5. It can be seen from table 5 that the accuracy of the proposed model can be further improved by using the CRF method in the test phase. We extend this work by replacing VGGNet-16 with ResNet-101. The conv1, res2c, res3b3, res4b22 and res5c of ResNet-101 are used as the side output, and other settings are kept unchanged. In table 5, the VGG network in proposed method as the backbone network is denoted as V and ResNet-101 as the backbone network is denoted as R. As can be seen from table 5, with the same training sets, the saliency graph generated by ResNet-101 without CRF improves the performance of the algorithm by 2% averagely, which indicates that the overall performance of the algorithm can be further improved by using the backbone network with better performance.

Note: V/R+CRF=VGGNet/ResNet with CRF, V/R-CRF=VGGNet/ResNet without CRF

## 5. Conclusions

The existing saliency object detection methods are difficult to highlight the clear boundary of the whole object and uniformly highlight the entire internal region in complex images. Therefore, we propose a new frog leaping algorithm-oriented fully convolutional neural network for motion object saliency detection. It extracts multi-scale and multi-level features from different convolution layers in the VGG network. Meanwhile, an improved frog leaping algorithm is used to optimize the selection of initial weights during network initialization. The shallow convolution layer produces detailed information, while the deep convolution layer produces global information. Then, the connection layer is used to combine these rich image saliency features to generate a saliency map. In the test phase, in order to further obtain the saliency detection results with accurate contour and uniform internal region, it introduces the fully connected CRF for saliency update. The

**Table 5.** Comparisons with/without CRF

Data set	Method	maxF	F	wF	MAE
DUT-OMPON	V+CRF	0.758	0.707	0.667	0.085
DUT-OMPON	V-CRF	0.728	0.645	0.573	0.101
DUT-OMPON	R+CRF	0.797	0.750	0.722	0.072
DUT-OMPON	R-CRF	0.769	0.682	0.612	0.087
ECSSD	V+CRF	0.899	0.887	0.865	0.072
ECSSD	V-CRF	0.887	0.845	0.789	0.087
ECSSD	R+CRF	0.906	0.859	0.797	0.081
ECSSD	R-CRF	0.926	0.906	0.889	0.064
SED2	V+CRF	0.870	0.808	0.783	0.104
SED2	V-CRF	0.837	0.790	0.727	0.113
SED2	R+CRF	0.868	0.813	0.792	0.104
SED2	R-CRF	0.831	0.784	0.710	0.115
HKU	V+CRF	0.901	0.879	0.855	0.060
HKU	V-CRF	0.872	0.822	0.760	0.076
HKU	R+CRF	0.912	0.893	0.874	0.057
HKU	R-CRF	0.883	0.829	0.759	0.076
PASCAL-S	V+CRF	0.825	0.783	0.743	0.099
PASCAL-S	V-CRF	0.809	0.747	0.670	0.124
PASCAL-S	R+CRF	0.835	0.795	0.759	0.096
PASCAL-S	R-CRF	0.819	0.757	0.667	0.122
SOD V+CRF		0.844	0.796	0.773	0.135
SOD V-CRF		0.826	0.770	0.706	0.826
SOD R+CRF		0.849	0.798	0.784	0.132
SOD R-CRF		0.830	0.796	0.693	0.146

experimental results show that the proposed method performs better than the 14 representative methods in terms of five performance evaluation indexes on six public available benchmark data sets. For subjective vision, the saliency image obtained by the proposed method can better deal with various complex images, it not only can display the whole saliency object uniformly, but also can well retain the contour of saliency object in various scenes.

In future research, the recurrent network module and boundary detection module will be used to improve the detection performance on small object images. And we will apply them to practical engineering.

**Acknowledgments.** This work was supported by: 1) Jiangsu Art Fund: "The Stage Art Funding Project of "Cao. Ji"" (No.: 2020-14-002); 2) National Educational Information Technology Research Project Youth Project: Traditional Dance Movement Collection Based on Motion Capture and Construction of Teaching Resource Database (No. 186140095).

**Availability of data and materials.** The data used to support the findings of this study are available from the corresponding author upon request.

**Competing interests.** The authors declare that they have no conflicts of interest.

## References

1. Song H, Deng B, Pound M, et al. "A fusion spatial attention approach for few-shot learning," *Information Fusion*, vol. 81, pp. 187-202, 2022.
2. C. Guo and L. Zhang. "A Novel Multiresolution Spatiotemporal Saliency Detection Model and Its Applications in Image and Video Compression," *2013 IEEE Transactions on Image Processing*, vol. 19, no. 1, pp. 185-198, Jan. 2010, doi: 10.1109/TIP.2009.2030969.
3. Radojičić, D., Radojičić, N., Kredatus, S. "A multicriteria optimization approach for the stock market feature selection," *Computer Science and Information Systems*, Vol. 18, No. 3, pp. 749-769, 2021. <https://doi.org/doi.org/10.2298/CSIS200326044R>
4. Chen Y, Yang X, Zhong B, et al. "CNNTracker: Online discriminative object tracking via deep convolutional neural network," *Applied Soft Computing*, vol. 38, pp. 1088-1098, 2016.
5. Guo, Z., Han, D., Li, K. "Double-Layer Affective Visual Question Answering Network," *Computer Science and Information Systems*, Vol. 18, No. 1, pp. 155-168, 2021. <https://doi.org/10.2298/CSIS200515038G>
6. Li, H., Han, D. "Multimodal Encoders and Decoders with Gate Attention for Visual Question Answering," *Computer Science and Information Systems*, Vol. 18, No. 3, pp. 1023-1040, 2021. <https://doi.org/10.2298/CSIS201120032L>
7. N. Tong, H. Lu, L. Zhang and X. Ruan. "Saliency Detection with Multi-Scale Superpixels," *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1035-1039, 2014. doi: 10.1109/LSP.2014.2323407.
8. Gao S. "A Two-channel Attention Mechanism-based MobileNetV2 And Bidirectional Long Short Memory Network For Multi-modal Dimension Dance Emotion Recognition," *Journal of Applied Science and Engineering*, 26(4): 455-464, 2022.
9. Lamsiyah S, Mahdaouy A E, Ouatik S, et al. "Unsupervised extractive multi-document summarization method based on transfer learning from BERT multi-task fine-tuning," *Journal of Information Science*, 2021:016555152199061.

10. L. Jing, Y. Chen and Y. Tian, "Coarse-to-Fine Semantic Segmentation From Image-Level Labels," *IEEE Transactions on Image Processing*, vol. 29, pp. 225-236, 2020. doi: 10.1109/TIP.2019.2926748.
11. Wang G, Wang Z, Jiang K, et al. "Silicone Mask Face Anti-spoofing Detection based on Visual Saliency and Facial Motion," *Neurocomputing*, vol. 458, pp. 416-427, 2021.
12. Zheng X, Chen W. "An Attention-based Bi-LSTM Method for Visual Object Classification via EEG," *Biomedical Signal Processing and Control*, vol. 63:102174, 2021.
13. X. Shen and Y. Wu. "A unified approach to salient object detection via low rank matrix recovery," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 853-860, 2012. doi: 10.1109/CVPR.2012.6247758.
14. C. Yang, L. Zhang and H. Lu, "Graph-Regularized Saliency Detection With Convex-Hull-Based Center Prior," *IEEE Signal Processing Letters*, vol. 20, no. 7, pp. 637-640, July 2013. doi: 10.1109/LSP.2013.2260737.
15. Y. Xie, H. Lu and M. Yang, "Bayesian Saliency via Low and Mid Level Cues," *IEEE Transactions on Image Processing*, vol. 22, no. 5, pp. 1689-1698, May 2013. doi: 10.1109/TIP.2012.2216276.
16. L. Zhang, Z. Gu and H. Li, "SDSP: A novel saliency detection method by combining simple priors," *2013 IEEE International Conference on Image Processing*, pp. 171-175, 2013. doi: 10.1109/ICIP.2013.6738036.
17. Li L, Zhou F, Zheng Y, et al. "Saliency detection based on foreground appearance and background-prior," *Neurocomputing*, vol. 301(AUG.2), pp. 46-61, 2018.
18. Y. Piao, X. Li, M. Zhang, J. Yu and H. Lu, "Saliency Detection via Depth-Induced Cellular Automata on Light Field," *IEEE Transactions on Image Processing*, vol. 29, pp. 1879-1889, 2020, doi: 10.1109/TIP.2019.2942434.
19. L. Zhou, Z. Yang, Q. Yuan, Z. Zhou and D. Hu, "Salient Region Detection via Integrating Diffusion-Based Compactness and Local Contrast," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3308-3320, Nov. 2015, doi: 10.1109/TIP.2015.2438546.
20. C. Tang, P. Wang, C. Zhang and W. Li, "Salient Object Detection via Weighted Low Rank Matrix Recovery," *IEEE Signal Processing Letters*, vol. 24, no. 4, pp. 490-494, April 2017, doi: 10.1109/LSP.2016.2620162.
21. L. Wang, H. Lu, X. Ruan and M. Yang, "Deep networks for saliency detection via local estimation and global search," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3183-3192, doi: 10.1109/CVPR.2015.7298938.
22. Guanbin Li and Y. Yu, "Visual saliency based on multiscale deep features," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5455-5463, doi: 10.1109/CVPR.2015.7299184.
23. Chen H., Li Y., Su D. "RGB-D Saliency Detection by Multi-stream Late Fusion Network," *ICVS 2017. Lecture Notes in Computer Science*, vol. 10528, 2017. Springer, Cham.
24. S. Wang, R. Clark, H. Wen and N. Trigoni, "DeepVO: Towards end-to-end visual odometry with deep Recurrent Convolutional Neural Networks," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 2043-2050, doi: 10.1109/ICRA.2017.7989236.
25. N. Liu and J. Han, "DHSNet: Deep Hierarchical Saliency Network for Salient Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 678-686, doi: 10.1109/CVPR.2016.80.
26. X. Li et al., "DeepSaliency: Multi-Task Deep Neural Network Model for Salient Object Detection," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3919-3930, Aug. 2016, doi: 10.1109/TIP.2016.2579306.
27. Zou L. "An Intelligent Improvement Method Of Classroom Cognitive Efficiency Based On Multidimensional Interactive Devices," *Journal of Applied Science and Engineering*, 2022, 26(3): 445-454.

28. I. Batatia, "A Deep Learning Method with CRF for Instance Segmentation of Metal-Organic Frameworks in Scanning Electron Microscopy Images," *2020 28th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 625-629, doi: 10.23919/Eusipco47968.2020.9287366.
29. Zhang Q, Zuo B C, Shi Y J and Dai M. "A multi-scale convolutional neural network for salient object detection," *Journal of Image and Graphics*, vol. 25, no. 06, pp. 116-129, 2020. doi: 10.11834/jig.190395.
30. S. Yin and H. Li. "Hot Region Selection Based on Selective Search and Modified Fuzzy C-Means in Remote Sensing Images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 5862-5871, 2020, doi: 10.1109/JSTARS.2020.3025582.
31. L. Wang, L. Wang, H. Lu, P. Zhang and X. Ruan, "Salient Object Detection with Recurrent Fully Convolutional Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1734-1746, 1 July 2019, doi: 10.1109/TPAMI.2018.2846598.
32. X. Zhang, T. Wang, J. Qi, H. Lu and G. Wang, "Progressive Attention Guided Recurrent Network for Salient Object Detection," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 714-722, doi: 10.1109/CVPR.2018.00081.
33. P. Zhang, D. Wang, H. Lu, H. Wang and B. Yin, "Learning Uncertain Convolutional Features for Accurate Saliency Detection," *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 212-221, doi: 10.1109/ICCV.2017.32.
34. D. Zhang, J. Han and Y. Zhang, "Supervision by Fusion: Towards Unsupervised Learning of Deep Salient Object Detector," *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4068-4076, doi: 10.1109/ICCV.2017.436.
35. G. Li and Y. Yu, "Deep Contrast Learning for Salient Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 478-487, doi: 10.1109/CVPR.2016.58.
36. L. Huang, K. Song, J. Wang, M. Niu and Y. Yan, "Multi-graph Fusion and Learning for RGBT Image Saliency Detection," *IEEE Transactions on Circuits and Systems for Video Technology*, doi: 10.1109/TCSVT.2021.3069812.
37. G. Lee, Y. Tai and J. Kim, "Deep Saliency with Encoded Low Level Distance Map and High Level Features," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 660-668, doi: 10.1109/CVPR.2016.78.
38. J Su, Yi H, Ling L, et al. A surface roughness grade recognition model for milled workpieces based on deep transfer learning," *Measurement Science and Technology*, vol. 33, no. 4, 045014, 2022 (11pp).
39. L. Zhang, J. Sun, T. Wang, Y. Min and H. Lu, "Visual Saliency Detection via Kernelized Subspace Ranking With Active Learning," *IEEE Transactions on Image Processing*, vol. 29, pp. 2258-2270, 2020, doi: 10.1109/TIP.2019.2945679.
40. Wang J, Jiang H, Yuan Z, et al. Salient Object Detection: A Discriminative Regional Feature Integration Approach," *International Journal of Computer Vision*, vol. 123, pp. 251-268, 2017. <https://doi.org/10.1007/s11263-016-0977-3>
41. J. Li, Z. Wang and Z. Pan, "Double Structured Nuclear Norm-Based Matrix Decomposition for Saliency Detection," *IEEE Access*, vol. 8, pp. 159816-159827, 2020. doi: 10.1109/ACCESS.2020.3020966.
42. Y. Yuan, C. Li, J. Kim, W. Cai and D. D. Feng, "Reversion Correction and Regularized Random Walk Ranking for Saliency Detection," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1311-1322, March 2018, doi: 10.1109/TIP.2017.2762422.

**Yin Lyu** was born in Shenyang, Liaoning, in 1981 in China. He received a postgraduate master's degree from the Belarusian State University of Culture and Arts. Now, studying for a PhD in Dance at Taipei University of the Arts, Taiwan. Associate Professor of the

Dance Department of the Music College of Huaiyin Normal University. His research interests include: new media dance and digital dance research. His research interests include image processing, dance big data and cloud computing.

**Chen Zhang** was born in Harbin, P.R. China ,in 1965. Professor of Harbin sport university, master's tutor, dean of the College of Sports Art, concurrently vice chairman of the Heilongjiang Dancers Association, and chairman of the Heilongjiang Dance Sports Association. Main research direction: physical education training (sports dance); sports art.

*Received: March 20, 2022; Accepted: September 01, 2022.*