[Array 17 (2023) 100270](https://doi.org/10.1016/j.array.2022.100270)

|  |  |  |
| --- | --- | --- |
|  | Contents lists available at [ScienceDirect](https://www.elsevier.com/locate/array) |  |
| Array |
| journal homepage: [www.elsevier.com/locate/array](http://www.elsevier.com/locate/array) |
|  | | |

Salient object contour extraction based on pixel scales and hierarchical convolutional network

Xixi Yuana, Youqing Xiaob, Zhanchuan Caib,∗, Leiming Wuc,∗

a *College of Electronics and Information Engineering, Shenzhen University, Shenzhen 518061, China*   
b *School of Computer Science and Engineering, Macau University of Science and Technology, 999078, Macao Special Administrative Region of China* c *School of Information Engineering, Guangdong University of Technology, Guangzhou 510006, China*

|  |  |  |
| --- | --- | --- |
| A R T I C L E | I N F O | A B S T R A C T |
| *Keywords:*  Salient objects  Object contours  Adaptive loss function Hierarchical network Pixel scales |  | Image salient object contours are helpful for many advanced computer vision tasks, such as object segmen-tation, action recognition, and scene understanding. We propose a new contour extraction method for salient objects based on pixel scale knowledge and hierarchical network structure, which improves the accuracy of object contours. First, a deep hierarchical network is designed to capture rich feature details. Then, a new loss function with adaptive weighted coefficients is developed, which can reduce the uneven distribution influence of contour pixels and non-contour pixels in training datasets. Next, the object contours are classified based on the scale information of contour pixels. By importing the prior knowledge of scale categories into the network structure, the model requires a small number of training samples. Finally, the regression task of contour scale prediction is added to the network, and the precise contour scales of foreground objects are performed as prior knowledge or an auxiliary task. The experimental results demonstrate that compared with related methods, the proposed method achieves satisfactory results from precision/recall curves and F-measure score estimation on three datasets. |

**1. Introduction**

The salient object (or foreground object) in a natural image is always interested to human beings, and it is a research trend in the era of artificial intelligence. The contours are essential structure character-istics of objects, which provides posture, size, and category information of objects in an image [1–3]. Therefore, it is significant to study approaches of salient object contour extraction.

Object contours play an important role in semantic recognition tasks, such as object extraction, action recognition, and behavior analy-sis [4–6]. In fact, the contour refers to the periphery of an object or the outer frame of a figure. The contour detection aims to extract curves that can reflect the shapes of the object in an image. In addition, there are some differences among edges, boundaries and contours. The edges and boundaries generally means the discontinuities of objects in an image in terms of photometrical, geometrical, and physical features. Firstly, in [7], the boundary is defined as the contour of an image, and it reflects the changes of pixel ownership from one object to another. Other researchers tend to regard contours as boundaries of interesting regions. When the contours are not region boundaries, the closed contours cannot be generated by contour detectors, nor can the image be divided into different regions. Secondly, the image edge is

the discontinuity of characteristic distribution (e.g., pixel grayscale and texture) in an image. It is a collection of pixels with step changes in the image. As one of the primary image features, image edge detection is mainly used to enhance the contour edges, details and grayscale jumps in an image.

Many researches have been conducted around image contour ex-traction. Firstly, the early image processing methods basically locate contours by extracting texture features [8,9]. However, due to the massive computation of traditional methods, it is difficult to deal with complex scenarios in an image. Then, with the spring of deep learning methods, image contour extraction has been further developed. For instance, the holistically-nested edge detection (HED) method [10] is based on ‘‘VGG16’’ (i.e., Visual Geometry Group-16) model [11] and side output strategy to extract contours from natural images, which is illustrated in Fig. 1(a). The edge detection method based on richer convolutional features (RCF) [12] concatenates adjacent convolutions at each stage of VGG16, and it effectively improves feature extraction performance, which is represented in Fig. 1(b). However, the existing end-to-end methods extract all edges in the image, and they cannot rec-ognize the salient object [13]. The edge guidance network (EGNet) [14] adopts the step-by-step method to detect salient object: in the first

∗ Corresponding au[thors.](mailto:yuanxixi@szu.edu.cn)

*E-mail addresses:* [yuanxixi@szu.edu.cn](mailto:yuanxixi@szu.edu.cn) (X. Yuan), [xiaoyq11@vanke.com](mailto:xiaoyq11@vanke.com) (Y. Xiao), [zccai@must.edu.mo](mailto:zccai@must.edu.mo) (Z. Cai), [leiming\_wu@gdut.edu.cn](mailto:leiming_wu@gdut.edu.cn) (L. Wu).

<https://doi.org/10.1016/j.array.2022.100270>  
[Received 10 October 2022; Received in revis](https://doi.org/10.1016/j.array.2022.100270)ed form 2 December 2022; Accepted 5 December 2022   
Available online 7 December 2022   
[2590-0056](http://creativecommons.org/licenses/by-nc-nd/4.0/)/© 2022 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license ([http://creativecommons.org/licenses/by-](http://creativecommons.org/licenses/by-nc-nd/4.0/)

[nc-nd/4.0/](http://creativecommons.org/licenses/by-nc-nd/4.0/)).

*X. Yuan et al.*  *Array 17 (2023) 100270*

step, the end-to-end method is used to obtain salient objects; in the second step, the contours are extracted based on gradient information of salient objects. However, this method with two steps is complicated in computation. At the same time, the error is expanded, including the salient object extraction error and the contour extraction error with gradient information. In the deep category-aware semantic edge detection network (CASENet) [15], each contour pixel is associated with more than one category. The pixels appear in contours or junctions belonging to two or more semantic categories. A saliency detection method based on salient contour-aware with twice learning strategy is proposed in [16], but it cannot always extract saliency contours accurately. Liu et al. [17] presented a simultaneous detection method of salient object, edge, and skeleton using the dynamic feature inte-gration (DFI) and the task-adaptive attention module. It produces most important edges in the image, but it cannot distinguish object contours exactly.

A new salient object contour extraction method is proposed in this paper, which can recognize the foreground object as illustrated in Fig. 1(c). In this method, an improved hierarchical network is designed to integrate rich features. Then, a new Softmax loss function with adap-tive coefficients is proposed to balance unevenly distributed samples, which is used to improve the accuracy of contour extraction. Finally, in order to evaluate object contour extraction performance, three new datasets are presented based on SK-SMALL [18], SK-LARGE [19], and WH-SYMMAX [20] datasets, and they have skeleton scales originally. In this paper, these datasets are regenerated for salient object contour position, and then they are used in training and testing experiments. In addition, there are four main contributions of the proposed method: (1) Different from popular edge extraction methods of HED, RCF, and DFI, the proposed method extracts salient object contours, which is more significant than image edges for intelligent industry. Besides, this is an end-to-end method without calculating gradient information to save computing resources.

(2) The extracted contour pixels contain scale information, which can be regarded as a prior knowledge and an auxiliary task to improve the recognition accuracy.

(3) A new loss function with adaptive weighted coefficients is proposed. The new loss function is suitable for images with uneven distribution of sample categories, which helps to improve the extraction accuracy of contour pixels. While the loss function used in the HED can only be used for binary classification problems.

(4) Furthermore, a hierarchical network for contour extraction is de-signed to extract rich features, and new salient object contour datasets with scale information are built.

The remainders of this paper are organized as follows. Section 2 introduces related works of contour extraction and the basic network structure. The principle of the salient object contour extraction are described in Section 3. Section 4 demonstrates the effectiveness of the proposed method. At last, the conclusion and future works are stated in Section 5.

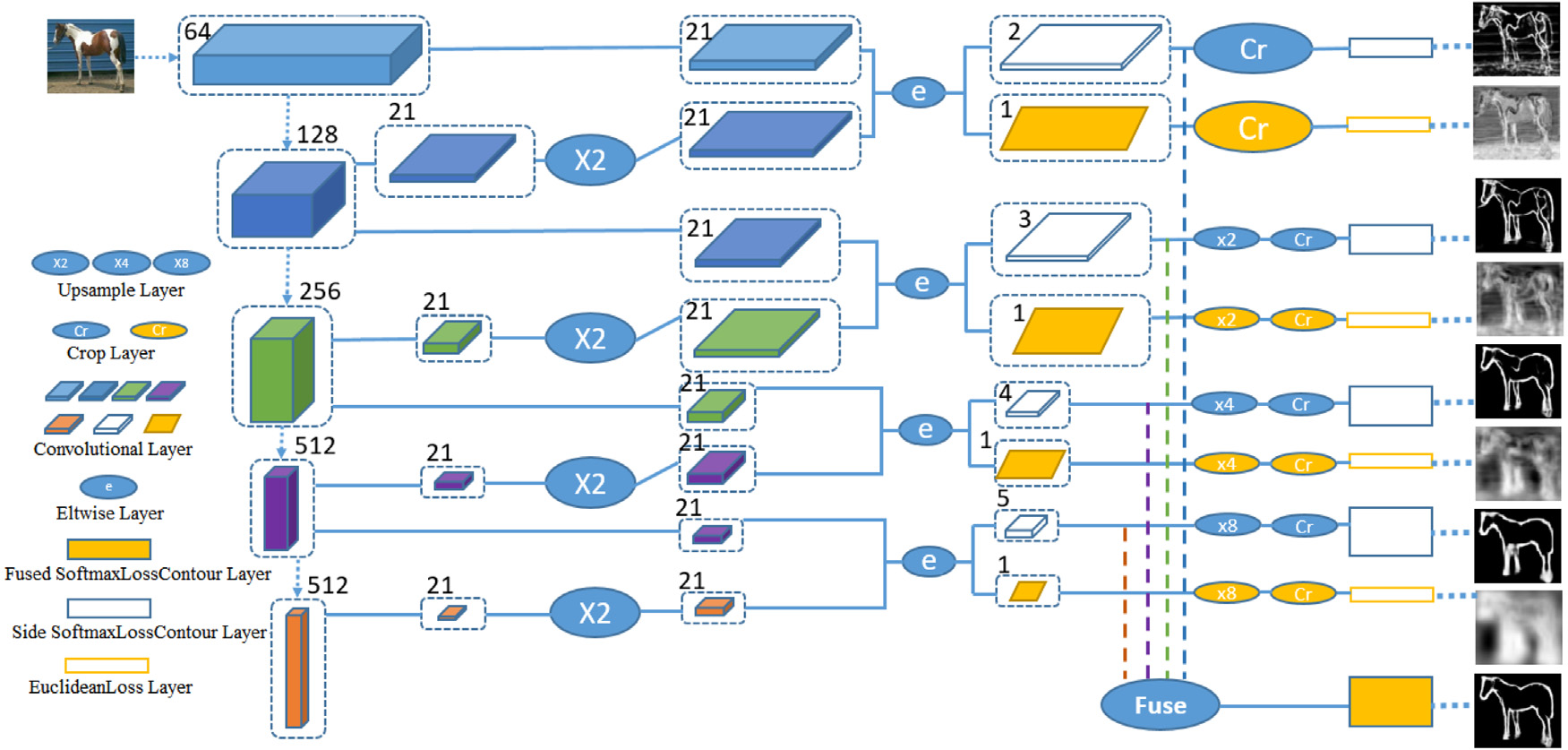
**2. Related work**

Different from the existing contour extraction methods, the pro-posed method can extract salient object contours in an image. The development of contour extraction methods and the multi-task hierar-chical network are introduced below.

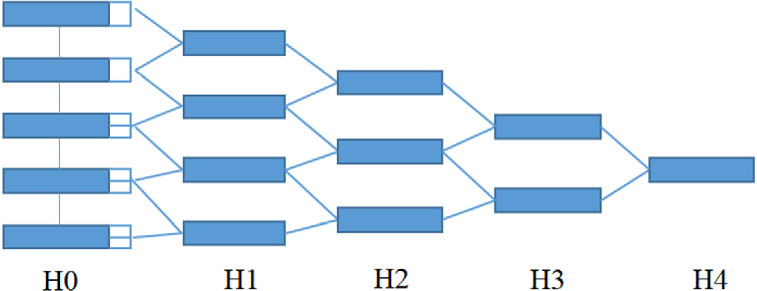
***Contour extraction***: In the past decades, many researches on image contour extraction have emerged. These are mainly classified into two categories. One is the early traditional image processing methods, such as Roberts operator, Sobel operator, and Canny operator. The other one is the machine learning methods. In recent years, machine learning has been widely used in intelligent industries [17,21,22]. In the HED method [10], the ‘‘side output’’ convolution is used to improve the edge extraction accuracy and computational efficiency. Afterwards, the scale-associated side-output (FSDS) is proposed in [18]. It realizes

2

*X. Yuan et al.*  *Array 17 (2023) 100270*



**Fig. 2.** The proposed network structure of salient object contour extraction.



**Fig. 3.** The hierarchical network structure with five levels based on the VGG16 network structure.

*3.1. Hierarchical network structure*

The proposed network structure is shown in Fig. 2, and the new hi-erarchical network structure is designed based on VGG16. The VGG16 has five stages, and the pooling layer is connected at each stage to adjust the receptive field sizes. Only the last convolutional layer at each stage performs hierarchical outputs. It is known that the VGG16-based network structure loses features in pooling processes with the increase of receptive fields. The side-output strategy can reduce feature losses in feature extraction, which is adopted in the designed network structure. The hierarchical network structure is shown in Fig. 3, which con-tains five levels (i.e., H0, H1, H2, H3, and H4), and the performance of them are analyzed in experiments. The method with three levels of H0, H1, and H2 are explained specifically in the following. Fig. 4(a) and (b) represents the hierarchical structure of H1 and H2, and their outputs are H1-out and H2-out. Fig. 4(c) illustrates the fused output of H1-out and H2-out, which is called H1H2-out (when the network structure only contains H0, H1, and H2 levels). It should be noted that there is no output layer at the H0 level, where each output layer corresponds to a loss function. In addition, a new Softmax loss function is customized to deal with imbalance distribution of contour and non-contour pixels. In this paper, the regression tasks are only used as auxiliary tasks to reduce losses of feature information, which are not fused in classification tasks. The contour pixels of a salient object have scale features, which is denoted as a real number. The linear regression is used to fit a prediction model for the values of *𝑌* and *𝑋* in the observed dataset. Therefore, the linear regression task is used to obtain contour scales. In order to reduce feature losses, the regression tasks are added into the network structure named as ‘‘Ours1AR’’ and ‘‘Ours2AR’’. The biggest differences between Figs. 4 and 5 lie in the linear regression tasks and loss functions.

3

*X. Yuan et al.*  *Array 17 (2023) 100270*

|  |  |  |  |
| --- | --- | --- | --- |
| **Table 1**  The contour scale categories and the ranges.  Scales ≥ 150 ∥*<* 1 [1, 10) Category 0 1 | [10, 26) 2 | [26, 60) 3 | [60, 150) 4 |
| sizes of receptive fields increase gradually, so the counter pixels are defined into five categories in this paper.  ***Contour pixel scale piecewise quantization***: As shown in Fig. 6(c), the scale of a contour pixel refers to the radius of the maximum inscribed circle that is centered on the skeleton point and is tangent to the contour of the object. In other words, the scale is the distance between the skeleton point and the contour point of the object that is closest to the skeleton point.  In the classification task of contour pixels, the pixels are divided into *𝑘* categories. The category 0 represents non-contour pixels. The VGG16 network structure has five stages, and the last convolution of each stage corresponds to the sizes of receptive fields with 5, 14, 40, 92, and 196, respectively [19]. According to the ranges of different receptive fields, the contour scale quantization category table is able to be obtained in Table 1. For example, the second category of contour pixels represents the scales of pixels between the range of [12, 26). In Fig. 6(d), the contours with different scales are shown in four kinds of colors, and they are red, yellow, blue, and green from big scales to small scales. Then, the corresponding skeletons are shown in four lighter colors. ***Contour map generation***: One image can be defined as a non-contour pixel map∑final contours of the image are composed of different scales. *𝑆*0 (non-contour pixels) is defined as category 0, and *𝑆𝑖* (*𝑖* ∈ {0*,* 1*,* … *, 𝑘* − 1}) *𝑛𝑜𝑛*−*𝑐𝑜𝑛𝑡𝑜𝑢𝑟*and a contour pixel map ∑ *𝑐𝑜𝑛𝑡𝑜𝑢𝑟*. The  is the predicted category of scales. For instance, *𝑆*2 represents the pixel set with the scales belonging to category 2, i.e., the scales of those pixels are lie in [10*,* 26). More details are shown in the con-tour scale piecewise quantization section. Then, it can be found that the *𝑆*1 following equations are obtained:⋃ *𝑆*2⋃ *𝑆*3⋃ ⋯ ⋃ *𝑆𝑘*−1 is the contour map. Consequently, the  *𝑐𝑜𝑛𝑡𝑜𝑢𝑟*∑ = *𝑆*1⋃*𝑆*2⋃*𝑆*3⋃⋯⋃*𝑆𝑘*−1 (1)  *𝑛𝑜𝑛*−*𝑐𝑜𝑛𝑡𝑜𝑢𝑟*∑ = *𝑆*0 (2)  After *𝑆*0 is obtained, *𝐼* − *𝑆*0 is another form of object contour map with binary image, where *𝐼* is the identity matrix in the trainer, as shown in Fig. 7. | | | |

*3.3. Scale prediction of contour pixels*

***The normalization of contour scales and loss functions in the regression task***: The scale of a contour pixel is a real number that can be predicted in regression tasks. In order to improve the prediction ac-curacy, a normalization method of contour scale is proposed in training steps. The network proposed in this paper consists of five stages, and each stage has independent regression tasks to predict the scales. Noted that there are different numbers of regression tasks at different levels. For example, the H1 level has five regression tasks and the H2 level has four regression tasks. The regression task is used to predict the contour scale at different stages, depending on the size of receptive fields. Therefore, it is necessary to establish the correspondence between the ground truth of contour scales and the values of receptive fields at different stages during the training stage, as shown in Fig. 8.

The contour scale normalized before training is in the following:

*𝐺𝑇* =

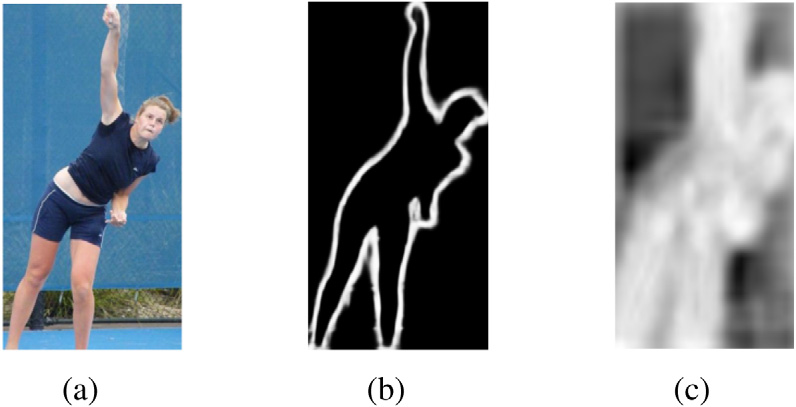
where *𝑎𝑖* denotes the receptive field at stage *𝑖*. *𝐺𝑇𝑛𝑜𝑟𝑚* denotes the contour scale after normalization, which is abbreviated as *𝐺𝑇* in (3). ⎧⎪⎨⎪⎩ 0*,*   
 *𝐺𝑇𝑖𝑛𝑖𝑡*

*𝑎𝑖* × *𝑟,*  *𝐺𝑇𝑖𝑛𝑖𝑡* ≤ *𝑎𝑖*

*𝐺𝑇𝑖𝑛𝑖𝑡 > 𝑎𝑖*   
 (3)

4

|  |  |
| --- | --- |
| *X. Yuan et al.* | *Array 17 (2023) 100270* |
|  | |



**Fig. 9.** The contour scale prediction results. (a) is the original image. (b) is the contour

map generated by classification tasks. (c) is the contour map generated by regression

tasks.

can be calculated by (4):

*𝐿𝑅𝑠𝑐𝑎𝑙𝑒* = { 0*,*  *𝐿𝑅𝑖𝑛𝑖𝑡*

*𝑟* × *𝑎𝑖,*  *𝐿𝑅𝑖𝑛𝑖𝑡* ≤ *𝑎𝑖*

*𝐿𝑅𝑖𝑛𝑖𝑡 > 𝑎𝑖*   
 (4)

where *𝐿𝑅𝑖𝑛𝑖𝑡* is the initial predicted scale and *𝐿𝑅𝑠𝑐𝑎𝑙𝑒* is the final contour scale in the regression task. In addition, *𝑟* is the magnification, and it

is set as ‘‘2’’ in this paper.

If *𝐿𝑅𝑠𝑐𝑎𝑙𝑒* ∈ [*𝑆𝑙𝑜𝑤, 𝑆ℎ𝑖𝑔ℎ*)*,* then:

*𝑠𝑐𝑎𝑙𝑒* = *𝐿𝑅𝑠𝑐𝑎𝑙𝑒,*

If *𝐿𝑅𝑠𝑐𝑎𝑙𝑒* ∉ [*𝑆𝑙𝑜𝑤, 𝑆ℎ𝑖𝑔ℎ*)*,* then: (5)

*𝑠𝑐𝑎𝑙𝑒* = *𝑆𝑙𝑜𝑤* + *𝑆ℎ𝑖𝑔ℎ*

2 *.*

According to the predicted contour pixel category and the introduc-

tion of Section 3.2, the coarse range of pixel scales can be represented

as [*𝑆𝑙𝑜𝑤, 𝑆ℎ𝑖𝑔ℎ*). For example, if a pixel belongs to category 2, that means

the scale of this pixel ranges at [10, 26), where the value of *𝑆𝑙𝑜𝑤* is 10 and the value of *𝑆ℎ𝑖𝑔ℎ* is 26. Combining the predicted values of the regression task and the classification task, the final contour pixel scale

can be got according to (5).

*3.4. Designed softmax loss function*

|  |
| --- |
| **Pseudo Code 1** Computation of Coefficient *𝜆* |
| **Input:** (*ℎ, 𝑤*)  **Output:** *𝜆*  1: % *ℎ* means the width of original image, *𝑤* means the height of original image, and *𝜆* is the weight coefficient.  2: Let *𝑖* represent pixel point and *𝑗* be the category of pixel.  3: *𝑠𝑢𝑚𝑃 𝑖𝑥𝑒𝑙𝑠* = *ℎ* × *𝑤*  4: **for** *𝑖* = 1 to *𝑠𝑢𝑚𝑃 𝑖𝑥𝑒𝑙𝑠* **do**   **if** pixel *𝑖* ∈ {*𝑗*}*, 𝑗* = 1*,* 2*,* ⋯ *𝑘.* **then** 5:  6: Compute *𝑝𝑖𝑥𝑒𝑙𝐶𝑙𝑎𝑠𝑠𝑗*+1  7: **end if**  8: **end for**  9: Compute *𝜆𝑗* = 1 −*𝑝𝑖𝑥𝑒𝑙𝐶𝑙𝑎𝑠𝑠𝑗*  *ℎ*×*𝑤*  , and the weight coefficient *𝜆𝑗* for category *𝑗* can be obtained. |

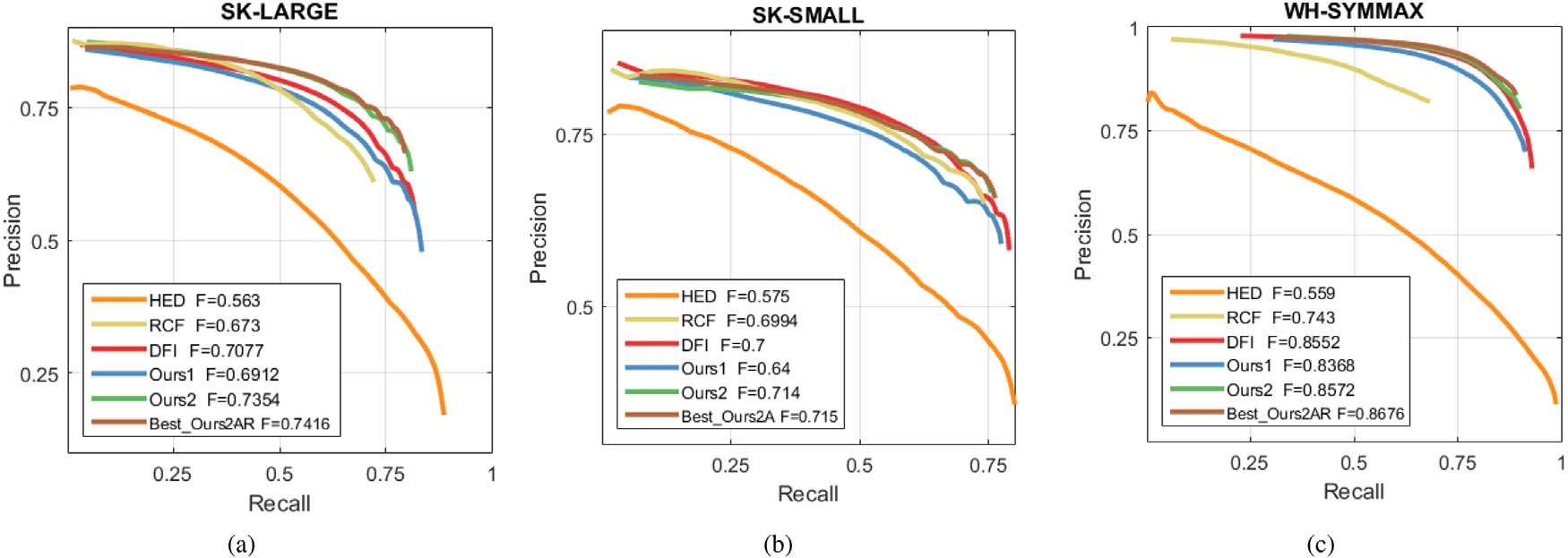
*4.1. Dataset and implementation details*

In order to obtain suitable image datasets with object contour annotations, three different datasets are used for experiments, which are obtained from three open datasets: SK-LARGE [19], SK-SMALL [18], and WH-SYMMAX [20]. On account of the skeleton and scale informa-tion in these datasets, the contour information of objects can be located according to the scales of skeletons. Then, the training datasets with contour labels are generated. The new SK-LARGE contains 746 training images and 745 test images with corresponding contour ground-truth. The new SK-SMALL dataset contains 506 images, and the first 300 images are used for training, which is also a subset of SK-LARGE. The new WH-SYMMAX dataset contains 328 horse images, and the first 228 are used for training.

To ensure justice, the experimental results of all methods are pro-duced under the same conditions (i.e., the same type of server and GPU). Then, the GPU configuration is Titan XP in the operation con-dition.

|  |  |  |
| --- | --- | --- |
| It is known that the standard Softmax loss function [28] is com-monly used to classify multiple classes with balanced number of sam-ples. However, the classification accuracy of the standard Softmax loss function is bad when the classes of different samples are unbalanced. In salient object contour recognition, the contour pixels of the object are far less than the non-contour pixels, so there exists a sample imbalance problem. In order to improve the classification accuracy, a new Softmax loss function is designed, which can be specifically expressed as:  *𝐽*(*𝜃*) = − 1  where *𝑚* marks the number of samples in the training set, *𝑘* is the *𝑚* [ *𝑚*∑∑*𝜆𝑖𝑗𝐼*{*𝑦𝑖* = *𝑗*}*𝑙𝑜𝑔* ∑*𝑘 𝑖*=1*𝑒𝑍𝑗*  *𝑒𝑍𝑗*  ] (6)  number of classes, *𝜆* is the defined balance factor namely the weight of the loss function. For the label *𝑦*, it can take on *𝑘* different values, so in the training set {(*𝑥*1*, 𝑦*1)*,* … *,* (*𝑥𝑚, 𝑦𝑚*)}, *𝑦*(*𝑖*) ∈ {1*,* 2*,* … *, 𝑘*} is obtained. *𝐼*{*𝑦𝑖* = *𝑗*} is the indicator function that indicates whether the sample *𝑖* belongs to category *𝑗*. If it is, the value is 1, otherwise the value is 0.  The coefficient *𝜆𝑖𝑗* is defined as *𝜆𝑖𝑗* = 1 −*𝑌𝑖𝑗 𝑌𝑖*. *𝑖* means the *𝑖*-th sample in one batch of the training dataset. If the parameter of batch-size is 1 (i.e., *𝑖* = 1), that is to say only one image for one batch is selected. |*𝑌𝑖*| the image is *ℎ* and the width of the image is *𝑤*, then the value of |*𝑌𝑖*| is *ℎ* × *𝑤*. |*𝑌𝑖𝑗*| represents the number of pixels belonging to class *𝑗*. | 5 | *4.2. Evaluation protocol* |
| The F-measure indicator and precision/recall (PR) curves are used as the evaluation protocol in this paper. The calculation process of F-measure is defined as follows: |
| F−measure = 2*𝑃 𝑅*  *𝑃* + *𝑅*  (7)  where *𝑃* and *𝑅* means precision and recall values, respectively, and they are calculated from the detected contour map and the ground-truth. Specifically, the calculation process of PR curve complies with the following steps: |
| 1. The predicted contour map is transformed into a binary map according to the threshold value.  2. The predicted binary contour map is matched with the ground-truth contour map. In this process, a relatively small position offset is allowed between the detected contour pixels and the ground-truth. If a detected contour pixel matches at least one ground-truth contour pixel, this point will be marked as a match-ing point *𝑇 𝑃* (true positive). Otherwise, the detected contour point does not match any ground-truth contour point, then this point will be marked as *𝐹𝑃* (false positive). If this predicted pixel does not belong to any contour pixel point, and it is considered as a contour point in ground-truth, then this point is called *𝐹𝑁* (false positive).  3. According to different thresholds, different∑*𝑇 𝑃*,∑*𝐹𝑃* , and∑ *𝐹𝑁* can be calculated. |
| In order to verify the performance of the new method, some com-parative experiments are performed based on different datasets, with precision/recall curves and F-measure score estimation. |

*X. Yuan et al.*  *Array 17 (2023) 100270*



**Fig. 10.** The PR curves of different contour extraction methods on three datasets in experiments.

|  |  |  |  |
| --- | --- | --- | --- |
| **Table 2**  The experiment results of different contour extraction algorithms with F-measures. | | | |
| SK-LARGE | | SK-SMALL | WH-SYMMAX |
| HED [10] RCF [12] DFI [17] | 0.5630  0.6730  0.7076 | 0.5750  0.6994  0.7000 | 0.5590  0.7430  0.8552 |
| Ours1  Ours1A  Ours1AR | 0.6912  0.7135  0.6525 | 0.6900  0.6840  0.7040 | 0.8368  0.8412  0.8385 |
| Ours2  Ours2A  Ours2AR | 0.7354  0.7311  **0.7416** | 0.7150  **0.7150**  0.7110 | 0.8652  0.8652  **0.8676** |
| Ours3  Ours3A  Ours3AR | 0.7166  0.7220  0.7130 | 0.6997  0.7080  0.7110 | 0.8504  0.8637  0.8669 |
| The PR values are calculated according to these formulas:  *𝑃* =  values can be obtained. Finally, these values are fitted to get a PR curve.  According to different thresholds, a series of precision and recall∑ *𝑇 𝑃* + ∑ *𝐹𝑃 ,*∑ *𝑇 𝑃*  *𝑅* =  ∑ *𝑇 𝑃* + ∑ *𝐹𝑁*∑ *𝑇 𝑃*  (8) | | | |

*4.3. Results of comparative experiments*

Three representative algorithms of HED, RCF, and DFI [10,12,17] are selected to compare with the method proposed in this paper. The HED originally exploits ‘‘side output’’ convolution, which greatly improves the edge extraction accuracy and computational efficiency. Then, the RCF further enriches edge features by fused convolutions. Next, the DFI adopts dynamic feature integration to optimize edge features, which is a state-of-the-art edge extraction algorithm. In this paper, some improvements are presented on the basis of HED and RCF network structures, and we optimize the object contours using pixel scale features.

In experiment procedures, firstly, the contour maps are generated based on the trained models. Then, the refined contours are obtained by the standard non-maximum suppression algorithm [29]. Finally, the refined contour maps are evaluated by PR curves and F-measures. In addition, the performance of the proposed algorithm is verified through comparative experiments on three datasets, and the generalization ability of the new algorithm is proved by cross validation.

The contour extraction accuracy results of several different network structures are compared in detail. Generally, the VGG16-based hierar-chical network maximally has four levels. In the experiments of contour extraction, the methods of different levels are named as Ours1, Ours2, and Ours3. It is verified that for the proposed method, when the level exceeds 2, the accuracy will be reduced, so the Ours3 is adopted at

6

*X. Yuan et al.*  *Array 17 (2023) 100270*



**Fig. 11.** Illustration of Object contour extraction results on SK-LARGE dataset for several selected images.

|  |  |  |  |
| --- | --- | --- | --- |
| **Table 6**  The comparison of position accuracy between our algorithm and the RCF. The values in the table are relative errors. | | | |
| SK-LARGE | | SK-SMALL | WH-SYMMAX |
| Ours1  Ours2  Ours1A  Ours2A  Ours1AR Ours2AR | +2.70%  +6.02% −3.05%  +9.27%  +8.63%  +10.19% | −1.34% −8.49%  +0.66%  +2.09%  +2.23%  +1.66% | +12.62%  +13.22%  +12.85%  +15.42%  +16.45%  +16.77% |

0.7416 and 0.8676 (see Table 2), respectively. While the results on the SK-SMALL dataset has change little by using Ours2A and Ours2AR. Tables 5 and 6 list the F-measures of the proposed method com-pared with HED and RCF methods on SK-LARGE, SK-SMALL, and WH-SYMMAX datasets. It is worth noting that if the weighted loss function and the regression task are not used (i.e., Ours1 and Ours2), the performance on the SK-SMALL dataset is slightly worse than that of the RCF method. If only one level and the weighted loss function are used without regression tasks (i.e., Ours1A), the performance on the SK-LARGE dataset is 3.05 percentage points, which is lower than that of the RCF method.

The PR curves in Fig. 10 show that the proposed method is much better than the HED and RCF methods. Specifically, Ours2AR has the best results on SK-LARGE and WH-SYMMAX datasets. The Ours2A is slightly better than the Ours2AR on SK-SMALL dataset, with the F-measure values of 0.715 and 0.711, respectively.

Some typical salient object contour extraction results on SK-LARGE dataset are shown in Fig. 11. From Fig. 11, it can be found that the extracted results of HED and DFI methods contain some non-contour details and background information. In addition, the results extracted by the RCF method lose some contour information. While the proposed method avoids these problems and achieves better results.

other contour extraction methods in terms of salient object contours. Moreover, the salient object contours obtained by the proposed method contain scale information, which can be applied to the subsequent skeleton generation and salient object generation. As a prior knowledge in training, it is also helpful to improve the model performance. In the future work, the new approach will be extendedly optimized in two possible manners. Firstly, a more effective network structure can be designed for object recognition. Secondly, multi-regressive aiding tasks like classification tasks can be merged into the network, which are not appeared in existing methods. After the fusion step, the regression data can be used to estimate more accurate inscribed circle radiuses between the contour and the skeleton. Finally, the contours are limited to five categories, which reduces the contour accuracy, so better results may be generated by increasing scale categories.

**CRediT authorship contribution statement**

**Xixi Yuan:** Investigation, Methodology, Writing – review & editing. **Youqing Xiao:** Conceptualization, Software, Methodology, Writing –original draft. **Zhanchuan Cai:** Supervision, Writing – review, Funding. **Leiming Wu:** Investigation, Writing – review & editing.

**Declaration of competing interest**

The authors declare that they have no known competing finan-cial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Data availability**

Data will be made available on request.

**5. Conclusion**  **Acknowledgments**

In this paper, a salient object contour extraction method is proposed by fusing scale information on hierarchically convolutional network. In the method, a new network structure with regression task and hierarchical feature integration mechanism is established, which has good performance when extracting salient object contours from nat-ural images; a new variable coefficient loss function is proposed to handle the unevenly distributed pixel classes in machine learning. The experimental results show that the proposed method is superior to

7

*X. Yuan et al.*  *Array 17 (2023) 100270*

**References**

[1] Gupta A, Anpalagan A, Guan L, Khwaja AS. Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues. Array 2021;10:100057. <http://dx.doi.org/10.1016/j.array.2021.100057>.

[2] [Li J, Fan J, Zhang Z. Towards noiseless object contours for weakly supervised semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. New Orleans, Louisiana; 2022, p. 16856–65.](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb2)

[3] [Kazuma F, Kazuhiko K. Generative and self-supervised domain adaptation](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb2) [for](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb3)  [one-stage object detection. Array 2021;11:100071.](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb3)

[4] [Gabriel L, Nasim H, Irene C. Semi-supervised](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb3) [learning approach for local-ization and pose estimation of texture-less objects in cluttered scenes. Array 2022;100247.](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb4)

[5] [Cheng M, Mit](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb4)[ra NJ, Huang X, Torr PH, Hu S. Global contrast based salient region](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb5)  [detection. IEEE Trans Pattern Anal Mach Intell 2015;37(3):569–82.](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb5)

[6] [Gong XY, Su H, Xu D, Zhang ZT, Shen F, Yang HB. An overvie](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb5)[w of contour](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb6)  [detection approaches. Int J Autom Comput 2018;15(6):656–72.](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb6)

[7] [Martin DR, Fowlkes CC, Malik J. Learning to detect natural i](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb6)[mage boundaries using local brightness, color, and texture cues. IEEE Trans Pattern Anal Mach Intell 2004;26(5):530–49.](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb7)

[8] [Martin DR, Fowlkes CC,](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb7) Malik J. Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Trans Pattern Anal Mach Intell 2004;26(5):530–49. <http://dx.doi.org/10.1109/TPAMI.2004.1273918>.

[9] [Rishi J, A CD. Preserving](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb9) [boundaries for image texture segmentation using](http://dx.doi.org/10.1109/TPAMI.2004.1273918) [grey](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb9)  [level co-occurring probabilities. Pattern Recognit 2006;39(2):234–45.](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb9)

[10] [Xie S, Tu Z. Holistically-nested edge detection. Int J Comput Vis 2](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb9)[017;125(1–](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb10) [3):3–18.](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb10)

[11] [Simonya](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb10)n K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014, arXiv preprint [arXiv:1409.1556](http://arxiv.org/abs/1409.1556).

[12] [Liu Y, Cheng M, Hu X, Bian J, Zhang L, B](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb12)[ai X, et al. Richer](http://arxiv.org/abs/1409.1556) [convolutional features](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb12)  [for edge detection. IEEE Trans Pattern Anal Mach Intell 2018;41(8):1939–46.](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb12) [13] [Dollár P, Zitnick CL. Fast edge detection using structured forests. IEEE Tra](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb12)[ns](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb13)  [Pattern Anal Mach Intell 2015;37(8):1558–70.](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb13)

[14] [Zhao JX, Liu JJ, Fan DP, Cao Y, Yang JF, Cheng MM. EGNet: Edge guidance network for Salient object detection. In: Proceedings of the IEEE international conference on computer vision. Seoul, Korea; 2019, p. 8778–87.](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb14)

[15] [Yu Z, Feng C, Liu MY, Ramalingam S. CASENet: Deep category-](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb14)[aware semantic edge detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Honolulu, Hawaii; 2017, p. 5964–73.](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb15)

[16] [Zhu C, Yan W, Liu S, Li T, Li G. Salient contour-aware](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb15) [based twice learning strategy for saliency detection. In: Proceedings of the IEEE/CVF international conference on computer vision workshops. Seoul, Korea; 2019, p. 2541–8.](http://refhub.elsevier.com/S2590-0056(22)00103-5/sb16)

8