



A lightweight contour detection network inspired by biology

Chuan Lin^{1,2,3} · Zhenguang Zhang^{1,2,3} · Jiansheng Peng^{1,2} · Fuzhang Li¹ · Yongcai Pan¹ · Yuwei Zhang^{1,2}

Received: 11 June 2023 / Accepted: 16 February 2024
© The Author(s) 2024

Abstract

In recent years, the field of bionics has attracted the attention of numerous scholars. Some models combined with biological vision have achieved excellent performance in computer vision and image processing tasks. In this paper, we propose a new bio-inspired lightweight contour detection network (BLCDNet) by combining parallel processing mechanisms of bio-visual information with convolutional neural networks. The backbone network of BLCDNet simulates the parallel pathways of ganglion cell–lateral geniculate nucleus and primary visual cortex (V1) area, realizing parallel processing and step-by-step extraction of input information, effectively extracting local features and detailed features in images, and thus improving the overall performance of the model. In addition, we design a depth feature extraction module combining depth separable convolution and residual connection in the decoding network to integrate the output of the backbone network, which further improves the performance of the model. We conducted a large number of experiments on BSDS500 and NYUD datasets, and the experimental results show that the BLCDNet proposed in this paper achieves the best performance compared with traditional methods and previous biologically inspired contour detection methods. In addition, BLCDNet still outperforms some VGG-based contour detection methods without pre-training and with fewer parameters, and it is competitive among all of them. The research in this paper also provides a new idea for the combination of biological vision and convolutional neural networks.

Keywords Contour detection · Parallel path · Convolutional neural network · Lightweight network

Introduction

As one of the low-level tasks in the field of computer vision [1], contour detection plays a crucial role in enhancing the performance of various mid-level and advanced vision tasks. These tasks include target detection [2], semantic segmentation [3], saliency detection [4] and occlusion reasoning [5], among others.

Traditional edge detection methods, such as Prewitt [6], Sobel [7], and Canny [1], primarily extract edges by calculating local gray-level changes in the image using differential operators. During the image edge extraction process, these methods concentrate on detecting the underlying image features [8], but often struggle to differentiate important background and texture. This limitation results in lower accuracy and performance in contour extraction, failing to meet the requirements of certain mid-level and advanced visual tasks. Hence, many experts and scholars have started to explore high-performance contour detection methods. In addition, as one of the new research hotspots, contour detection has also attracted the attention of the field of biology [9].

Inspired by the early discovery and suggestion by Hubel and Wiesel [10] that primary visual cortex (V1) neurons have the function of detecting edges and lines, several experts and scholars have proposed many bionic contour detection models based on the biological visual mechanisms effective for contour detection [11, 12]. For example, Grigorescu et al.

✉ Zhenguang Zhang
zhangzhenguang1227@163.com

✉ Jiansheng Peng
sheng120410@163.com

¹ School of Automation, Guangxi University of Science and Technology, Liuzhou 545006, People's Republic of China

² Key Laboratory of AI and Information Processing (Hechi University), Education Department of Guangxi Zhuang Autonomous Region, Hechi University, Yizhou, China

³ Guangxi Key Laboratory of Automobile Components and Vehicle technology, Guangxi University of Science and Technology, Liuzhou 545006, China

[13] used the Gabor operator, Gabor energy operator, and difference of Gaussian (DOG) operator to simulate simple cell response, complex cell response, and non-classical receptive field (nCRF) inhibitory characteristics on classical receptive field (CRF), and proposed a new contour detection model. Yang et al. [14] proposed the biomimetic contour detection model, double-opponency and spatial sparseness constraint (SCO), based on color antagonism mechanism and spatial sparseness constraint strategy (SSC). Akbarinia et al. [15] realized target edge extraction based on the color opposition mechanism from the retina to the visual cortex (V1) and the surround modulation characteristics of the receptive field of cells in the V1 area. Although the contour detection model, which simulates the biological vision mechanism, achieves better performance by reducing the background and texture to a certain extent compared to traditional methods, there are still some issues worthy of investigation. In previous methods, researchers typically employed mathematical formulas to simulate visual mechanisms or biological characteristics effective for contour detection in biological vision systems for extracting image contours. However, interactions between neurons in biological vision systems are typically complex and diverse. Thus, relying solely on a single mathematical function to simulate their functions is evidently inappropriate [8]. To this end, Tang et al. [8] proposed a method combining biological vision with deep learning. Tang et al. designed a learnable contour detection model using convolution kernels of different sizes to simulate nCRF and CRF's processing of feature maps. At the same time, the combination of image pyramids achieves the fusion of feature information at different scales, which further increases the complexity and diversity of the model and also provides new ideas for the design of bionic contour detection models. Later, Lin et al. [16], inspired by the effective mechanism of contour detection in the biological vision system, combined it with convolutional neural networks and self-attention mechanisms to propose a multi-level interactive contour detection model, MI-Net, achieving good performance.

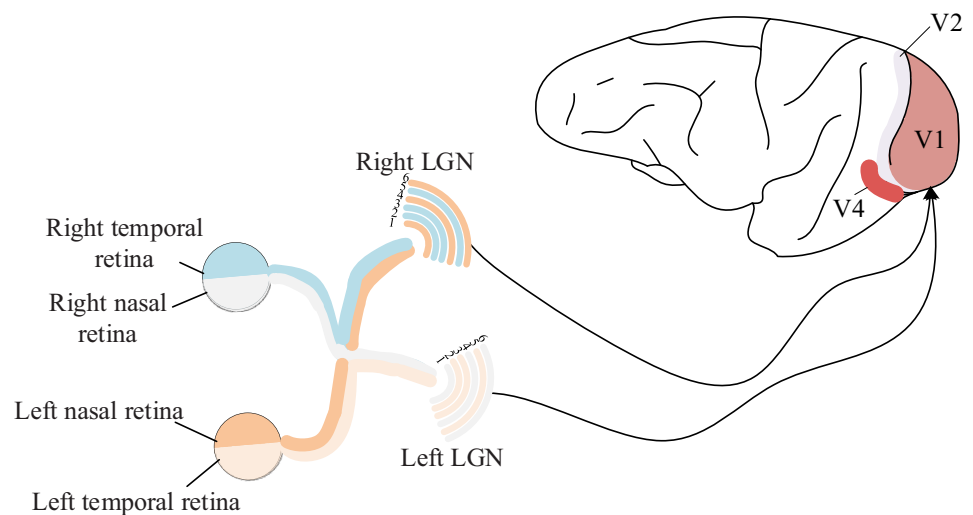
In the past period of time, the end-to-end contour detection model based on the convolutional neural network [17–22] has made breakthrough progress. For example, on the BSDS500 [23], the detection performance has been boosted from 0.598 [24] to 0.828 [22] in ODS (optimal dataset scale) F -measure. Recently, a transformer-based edge detection model [25] has achieved higher performance with an ODS of 0.848. However, although these methods achieve the best performance, they generally have high complexity, a large number of parameters, and slow processing speed, occupying a large number of computing resources. Furthermore, researchers incorporate parameters trained on ImageNet [26] into their models during training to achieve enhanced performance through the application of transfer learning. To minimize computational resource consumption and enhance model

processing speed, some researchers have initiated investigations into achieving high-performance image contour extraction under conditions of a simple model, minimal parameters, fast operation speed, and low resource consumption while examining contour detection models based on transfer learning. Later, inspired by the lightweight models in other visual tasks [27–29], some researchers proposed a lightweight model for contour detection. For example, Wibisono et al. [30] proposed a lightweight edge detection model called fast inference network for edge detection (FINED) by using expansion convolution to design a backbone network. Combined with the steps of edge extraction by traditional contour detection methods, a lightweight contour detection model traditional method inspired deep neural network (TIN2) [31] is proposed. Su et al. [32] proposed pixel difference network (PiDiNet), a simple and efficient edge detection network based on pixel difference, and achieved the best results in the lightweight model.

To sum up, it can be found that the design of the lightweight model is becoming a new research hotspot, attracting the attention of more and more researchers. For contour detection, although the lightweight model has achieved better performance than the traditional methods and some CNN-based models [17–19], there are still some problems to be solved. As we know, the emergence of CNN is inspired by the biological vision system [35], while the current lightweight model is mainly designed based on the experience of researchers, lacking the guidance of relevant biological vision mechanisms. Therefore, this paper proposes a new bio-inspired lightweight contour detection network (BLCDNet) combining biological vision and deep learning technology. Among them, our backbone network simulates three parallel channels formed by ganglion cells, lateral geniculate nucleus (LGN), and primary visual cortex (V1) in the biological visual system [33, 34], and simulates the different characteristics of the three parallel channels to achieve visual information processing and feature extraction. The transmission process of visual information from the retina to LGN to V1 is shown in Fig. 1. In addition, we also design a depth feature extraction module by using the depth separable convolution [29] which is widely used in lightweight networks. By further processing the output of the backbone network, we can comprehensively extract feature information and enhance the overall performance of the model. It is worth noting that our method has achieved the most advanced performance in the bionic contour detection model, and our method of combining biological vision with deep learning also provides new ideas for future research. Our contributions are summarized as follows:

1. We simulated the three parallel pathways formed by ganglion cells, LGN, and the primary visual cortex (V1) in

Fig. 1 The information transfer process of retina to LGN to V1. The formation of parallel channels starts from retinal ganglion cells (redrawn from the Refs. [33, 34])



the biological visual system and designed corresponding backbone networks. These include the large receptive field network simulating the pathway from ganglion cells to the V1 area with large cells, the small receptive field network simulating the pathway with small cells, and the hybrid network simulating the color pathway from ganglion cells to the V1 area. Finally, we combine the outputs of these three pathways to comprehensively extract and fuse the feature information.

2. We design the deep feature extraction module using deep separable convolutions. By further processing the features output by the backbone network, the contextual information is fully integrated and the overall detection performance of the model is improved.
3. We combine a backbone network that models parallel pathways with a designed deep feature extraction module to propose a biologically inspired lightweight contour detection network with simple structure and high efficiency and accuracy.

Related work

This paper mainly involves contour detection, biological vision mechanism, and lightweight network. We will briefly review the work in these three aspects.

Contour detection

The existing contour detection methods can be divided into traditional contour detection methods, bio-inspired contour detection methods, and learnable contour detection methods. Among them, the learnable contour detection methods can be divided into traditional machine learning methods and deep

learning methods. The traditional contour detection methods [1, 6, 7] mainly calculate the local gradient change of the image by the derivative of the differential operator to detect the edge. While these early contour detection methods can extract contours in images, their performance and accuracy are lacking. They struggle to precisely differentiate between the background and the image contours, making them susceptible to noise interference. In contrast, the bionic contour detection method [13–15] simulates the characteristics of a specific area or cell in the biological vision system using mathematical formulas. To some extent, this method achieves background and texture suppression in the image, resulting in commendable performance. Methods based on traditional machine learning [23, 36–38] use supervised learning and manual design features to extract contour. They regard the contour detection task as a binary classification task, classify the target image at pixel level by using the designed features and extract the target contour from the image successfully. For example, the oriented edge forests (OEF) algorithm based on a random forest classifier proposed by Hallman et al. [38] achieves the probability fusion of edges according to pixel points and then obtains image edges. The deep learning-based method [17–22] utilizes the excellent feature extraction capability of a convolutional neural network to fully extract feature information and achieve better contour detection performance. Xie et al. [17] first proposed an end-to-end detection model based on CNN, which extracted the target contour by outputting the features of the middle layer and fusing the features of different scales. Liu et al. [18] improved on this basis and proposed a contour detection model RCF with richer features. He et al. [22] improved model performance to a higher level by designing cascade networks and scale enhancement modules. Amaren et al. [39] proposed a new framework based on VGG16 by designing the fire module and incorporating residual learning. The framework

achieves a significant reduction in network complexity and can increase the depth of the network while preserving its low-complexity characteristics. Fang et al. [40] designed a novel local contrast loss to learn edge mapping as a representation of local contrast, addressing the edge ambiguity problem extracted by the current method. This design results in clear edge extraction and achieves good performance. Recently, Pu et al. [25] proposed a new contour detection model using transformer as the backbone network, which achieved better performance than bi-directional cascade network (BDCN) [22].

Biological visual mechanisms

In the biological visual system, the processing and transmission process of visual information from the retina to LGN to V1 is called the first visual pathway [41]. In this pathway, visual information undergoes transformation and processing by the retina before being transmitted through ganglion cells to the LGN. The LGN receives and processes visual information from the retina, subsequently transmitting the processed information to area V1. Within area V1, the visual information received from the LGN undergoes further processing and integration [33, 34]. As a research hotspot in the field of computer vision, contour detection has received much attention in the field of physiology. Studies have shown that there are many biological visual mechanisms in the first visual pathway that have been proved to be important for contour detection. For example, the modulation of CRF by nCRF in neurons in area V1 [13], the color antagonism mechanism in retina to V1 [14], and the dynamic modulation mechanism in the receptive field after neurons in area V1 are stimulated [42]. In addition, Hubel and Wiesel [10] also found and proposed in an earlier study that neurons in the V1 region of the biological visual system have the function of detecting edges and lines. At present, the research on the bionic contour detection model is mostly focused on the first visual pathway. Recently, Fan et al. [43] proposed a hierarchical scale convolutional neural network for facial expression recognition. In this method, they not only use enhanced kernel scale information extraction and high-level semantic features to guide low-level learning, but also propose a method to mimic human cognitive learning with knowledge transfer learning (KTL). The KTL process shares similarities with human cognitive ability in that it can be progressively enhanced by knowledge acquired from other tasks. In contrast, our approach takes inspiration from the parallel processing in biological vision and the step-by-step handling of visual information. Simultaneously, by integrating the characteristics of convolutional neural networks, we have designed a new backbone network. The network achieves good contour detection performance by extracting and fusion feature information step by step.

Lightweight network

Recently, to solve the problems of the contour detection method based on deep learning [17–22], such as complex models, a large number of parameters, and slow calculation speed, researchers proposed a lightweight network for contour detection [30–32]. They design the backbone network using existing experience or by combining traditional contour detection methods, thus reducing the complexity of the model, reducing the parameters of the model, and increasing the computational speed of the model. For example, Wibisono et al. [31] designed a convolutional neural network framework corresponding to the traditional edge detection scheme inspired by the edge extraction step in traditional methods. Su et al. [32] combined traditional central difference, angular difference, and radial difference with 2D convolution to propose differential convolution operation and construct pixel difference network (PiDiNet) for edge detection. Among them, the PiDiNet proposed by Su et al. [32] achieves the best performance.

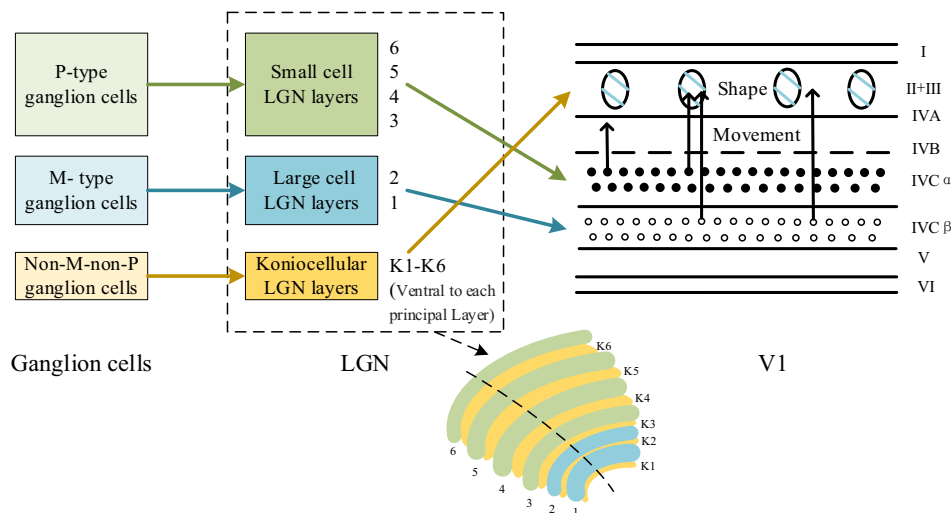
Based on the above analysis, we combined the design of a lightweight network with a biological visual mechanism and designed a new lightweight network for contour detection (BLCDNet) by simulating the processing and transmission process of visual information from the retina ganglion cells to LGN to V1. BLCDNet has the characteristics of low complexity, less parameter number, and less memory resource occupation, and achieves good results without the need for pre-training. Compared with other bio-inspired contour detection models, the results are the most advanced. In addition, this approach of combining lightweight networks based on deep learning with biological vision mechanisms also provides a new direction for further research.

Proposed methods

Information processing and transmission mechanism from ganglion cells to LGN to V1

Physiological studies have revealed that ganglion cells in mammalian retinas can be categorized based on appearance, connectivity, and electrophysiological properties. In both the macaque monkey retina and the human retina, three primary types of ganglion cells have been identified: large M-type ganglion cells, smaller P-type ganglion cells, and non-M–non-P ganglion cells [33, 44–46]. As shown in Fig. 2. They have different visual response characteristics and play different roles in visual perception. Among them, M-type ganglion cells have a larger receptive field, which is considered to be of great significance for the detection of motor stimulation. P-type ganglion cells have small receptive fields,

Fig. 2 The transmission process of visual information. Ganglion cells to LGN to V1 (redrawn from the Refs. [33, 34])



which are very suitable for distinguishing tiny details. Non-M–non-P cells are equally sensitive to different wavelengths of light, and they and some P-type ganglion cells are also known as color-opponent cells, reflecting the phenomenon that the response of a neuron’s receptive field centers to one color is canceled out by another color around the receptive field. In non-M–non-P ganglion cells, the two opponent colors information are blue and yellow [33]. Then the visual information processed by different ganglion cells is projected to the LGN layer.

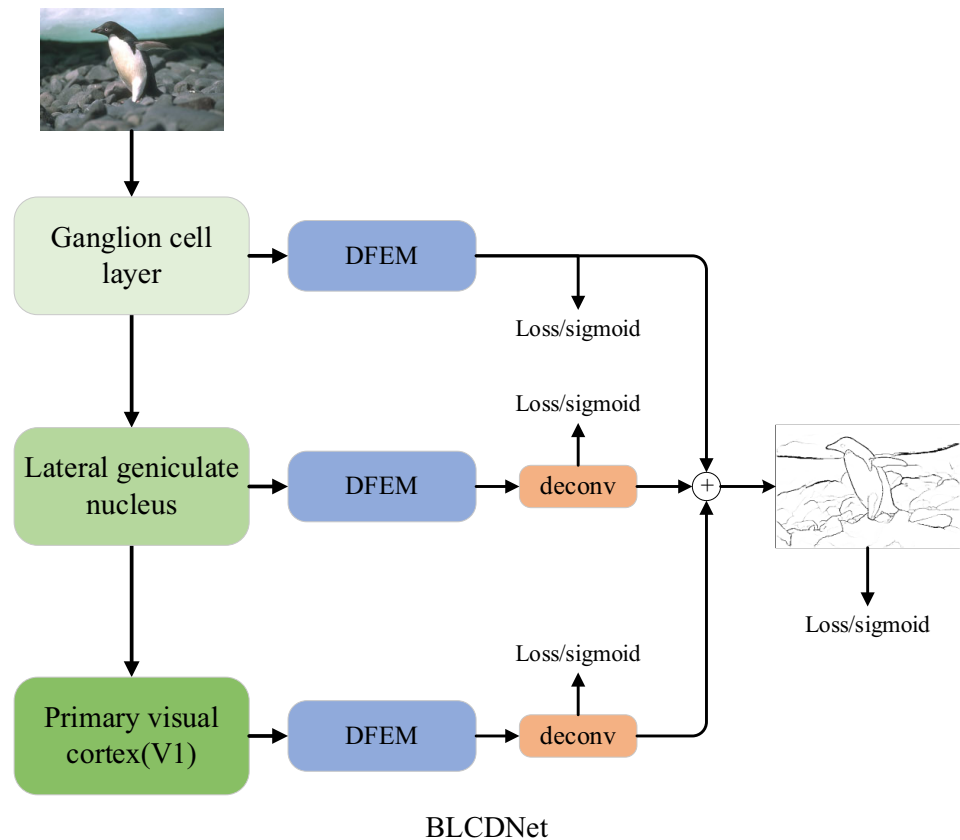
The research shows that LGN can be divided into six layers, starting from the most ventral layer and superimposed layer by layer [33, 47]. The detailed structure is shown in Fig. 2. Among them, the ventral layers 1 and 2 contain larger neurons, which are called large-cell LGN layers, and correspondingly receive the output from M-type ganglion cells. The neurons of dorsal layer 3–6 are called the small-cell LGN layers, which receives the output from P-type ganglion cells. Many tiny neurons on the ventral side of each of layers 1–6 make up koniocellular LGN layers to receive the output from non-M–non-P ganglion cells. Furthermore, through physiological experiments, the researchers concluded that neurons in LGN have similar characteristics to their corresponding ganglion cells. Specifically, large-cell LGN neurons share similarities with M-type ganglion cells, small-cell LGN neurons are akin to P-type ganglion cells, and koniocellular LGN layer neurons resemble non-M–non-P-type ganglion cells [33].

LGN-processed visual information was projected to the primary visual cortex (V1) [44, 47]. Region V1 is divided into six layers according to its cell arrangement and structure and Brodmann’s [33, 48] convention that the neocortex has six layers of cells. As shown in the rightmost part of Fig. 2, the IV layer contains three sub-layers (IVA, IVB, IVC), and the IVC sub-layer contains two sub-layers (IVC α ,

IVC β). In the same way that LGN receives output from ganglion cells, some of the different layers in V1 receive output from LGN’s different layers. Among them, the IVC α layer receives the projection from the large-cell LGN layer, the IVC β layer receives the projection from the small-cell LGN layer, and part of the cells in the III layer receive the projection from koniocellular LGN layers. ###Then, the visual information processed by the IVC α layer was transferred to the IVB layer, and the visual information processed by the IVC β layer was transferred to the III layer. It is noteworthy that the region of V1 receiving visual information has similar characteristics to the corresponding LGN neuron. In addition, through relevant experiments, the researchers found that visual information began to mix after being transmitted to the III and IVB sub-layers of the V1 region, and before that, they were independent in the processing transmission process of ganglion cells to LGN to V1.

In summary, we can find that visual information is processed by different channels in the processing and transmission process of ganglion cells to LGN to V1 [33, 34, 47]. That is, M-type ganglion cells, LGN layer of large cells and IVC α layer of V1 region form a large cell channel, which has the characteristics of a large receptive field and is more sensitive to motor stimulation. P-type ganglion cells, LGN layer of small cells and IVC β layer of V1 region constitute small cell channels, which have small receptive fields and are sensitive to detailed information. Non-M–non-P ganglion cells, koniocellular LGN layers, and some regions in layer III of the V1 area constitute yellow–blue antagonistic color channels, which are sensitive to blue–yellow antagonistic information. Inspired by this, this paper simulates three parallel channels composed of ganglion cells, LGN, and the V1 area. It models the characteristics of these three channels in processing visual information to design a new lightweight contour detection network with commendable performance.

Fig. 3 Overall structure diagram of BLCDDNet. The green part on the left is the backbone network, whose overall detailed structure is introduced in “[Backbone network](#)”. The right part is the decoding network, in which DFEM is the depth feature extraction module proposed in this paper. We introduced it in detail in “[Depth feature extraction module](#)”



Overall structure of bionic lightweight contour detection model

Figure 3 shows the overall structure of BLCDDNet, which includes two parts: the backbone network and the decoding network. The backbone network is responsible for extracting feature information of different scales and inputting the extracted features into the decoding network. It is inspired by three parallel pathways in the retinal ganglion cells to LGN to V1 region. In the decoding network part, we designed a new feature extraction module named DFEM (depth feature extraction module). It uses residual error and depth separable convolution to further process the output of the backbone network, which realizes the feature extraction and fusion more fully and improves the overall performance of the model.

Backbone network

Figure 4 shows the detailed structure of our backbone network, corresponding to the green section in Fig. 3. In the biological visual system, visual information processed by the retina is transmitted to the LGN through different types of ganglion cells. Upon receiving this visual information, the LGN processes it once again and transmits it to the primary

visual cortex, V1. After that, the V1 region consciously processes the received visual information, and after the initial processing is completed, it is transmitted to the higher regions via the ventral and dorsal pathways. It is worth noting that the process of processing and transmitting visual information from ganglion cells to LGN to V1 is divided into three parallel channels, and each channel has different characteristics and features, which do not interfere with each other when processing visual information. As the end point of parallel pathways and the starting point of ventral and dorsal pathways, the V1 region also plays a crucial role in the conscious processing of visual stimuli.

Inspired by this, in this paper, we designed a new backbone network named parallel path feature extraction network (PFENet) by using a convolution neural network to simulate three parallel paths of ganglion cells to LGN to V1. Their detailed composition is shown in a–e in Fig. 5, the large receptive field feature extraction network is composed of a dilated convolution with a convolution kernel size of 3×3 and a dilated rate of 5, and a maximum pooling layer, which simulates the magnocellular pathway in ganglion cells to LGN to V1. The feature extraction network of a small receptive field consists of conventional convolution and pool layer with a convolution kernel size of 3×3 , which simulates the parvocellular–interblob pathway in ganglion cells to LGN to V1.

Fig. 4 Detailed diagram of backbone network structure. The blue part corresponds to the big cell channel from ganglion cells to V1, the green part corresponds to the small cell channel, and the gold part corresponds to the color antagonistic channel. Block_b, Block_s, Block_c, Block_S_C and Block_C_M are different structures, which are explained in detail in Fig. 5. 32,64,128 indicates the number of channels in the feature map

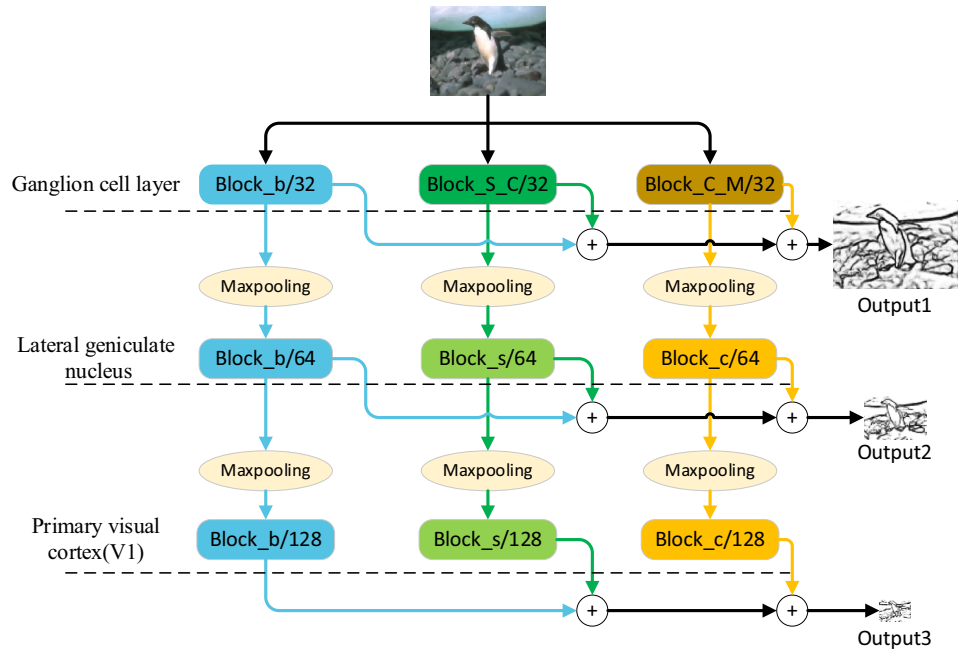
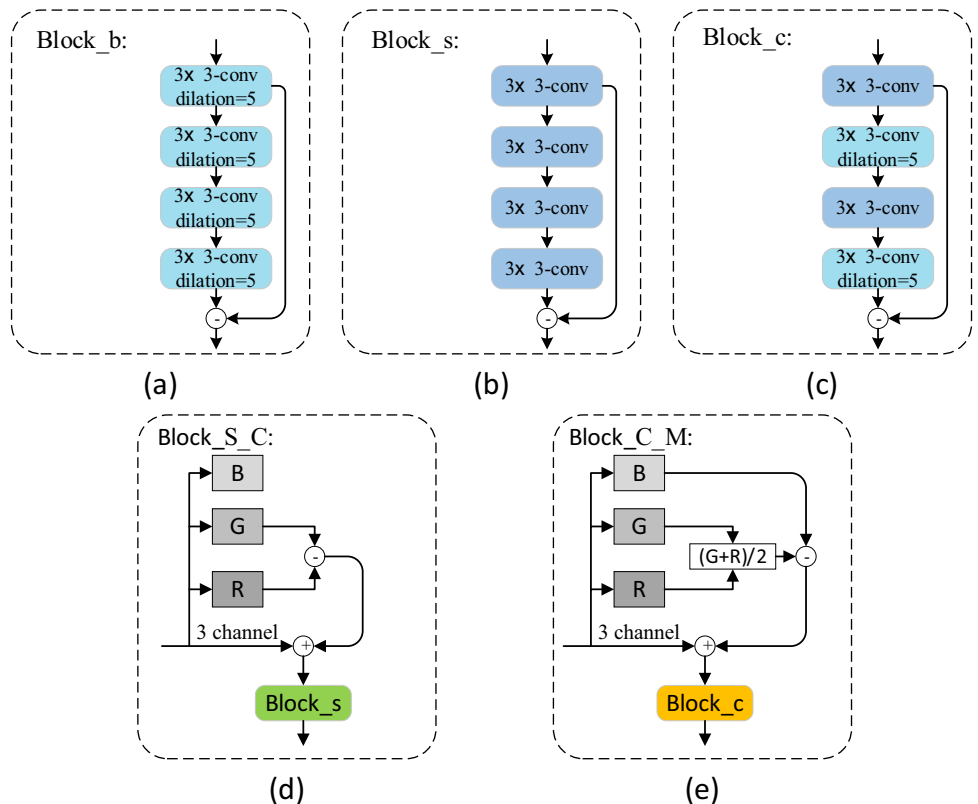


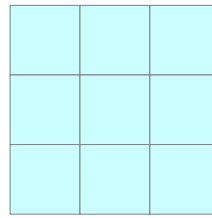
Fig. 5 a–e The detailed structures of Block_b, Block_s, Block_c, Block_S_C and Block_C_M in Fig. 4, where, B, G, and R in (d) and I represent the three channels in the color image



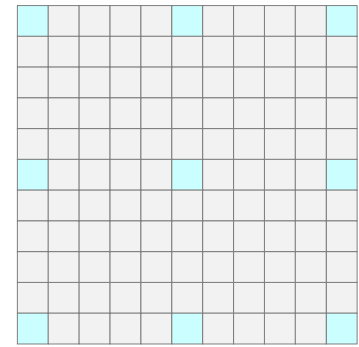
The color adversarial feature extraction network is composed of a conventional convolution with a convolution kernel size of 3×3 , a dilated convolution with a convolution kernel size of 3×3 , and a dilated rate of 5, and a pooling layer, simulating the blob pathway in ganglion cells to LGN to V1. Although

conventional convolution and dilated convolution have convolution cores of the same size, we set the dilation rate of dilated convolution to 5. Therefore, dilated convolution has a larger receptive field [49]. In addition, as in [17, 18, 50], we used the pooling layer to divide the network, and divided the

Fig. 6 The receptive field of conventional convolution and dilated convolution. **a** A conventional convolution with convolution kernel size 3×3 , that is, the dilated rate is 1. **b** A dilated convolution with convolution kernel size 3×3 , and the dilated rate is 5



(a) kernel size is 3×3 , $r=1$



(b) kernel size is 3×3 , $r=5$

large receptive field feature extraction network, small receptive field feature extraction network, and color adversarial feature extraction network into three stages, corresponding to retinal ganglion cells, LGN and V1, respectively. This also reflects the feature of extracting feature information step by step in the biological vision system.

a–e in Fig. 5 are represented by the following equation:

$$\text{Block_b} = C_{3 \times 3, 5} (C_{3 \times 3, 5} (C_{3 \times 3, 5} (C_{3 \times 3, 5} (I_{\text{input}})))) - C_{3 \times 3, 5} (I_{\text{input}}), \quad (1)$$

$$\text{Block_s} = C_{3 \times 3} * (C_{3 \times 3} * (C_{3 \times 3} * (C_{3 \times 3} * I_{\text{input}}))) - C_{3 \times 3} * I_{\text{input}}, \quad (2)$$

$$\text{Block_c} = C_{3 \times 3, 5} * (C_{3 \times 3} * (C_{3 \times 3, 5} * (C_{3 \times 3} * I_{\text{input}}))) - C_{3 \times 3} * I_{\text{input}}, \quad (3)$$

$$\text{Block_S_C} = \text{Block_s}((I_G - I_R) + I_{\text{input}}), \quad (4)$$

$$\text{Block_C_M} = \text{Block_c} \left(\left(\frac{(I_G + I_R)}{2} - I_B \right) + I_{\text{input}} \right), \quad (5)$$

I_{input} is the input image. $C_{3 \times 3, 5}$ and represents a dilated convolution with a convolution kernel size of 3×3 and dilation of 5. It is equivalent to a conventional convolution with a convolution kernel size of 11×11 and has a large receptive field. $C_{3 \times 3}$ is a conventional convolution with a convolution kernel of size 3×3 . Figure 6 shows the conventional convolution kernel dilated convolution with the same convolution kernel size. I_R , I_G and I_B represent the three channels of the color image. $\frac{(I_G + I_R)}{2}$ indicates the yellow information. “*” represents the convolution.

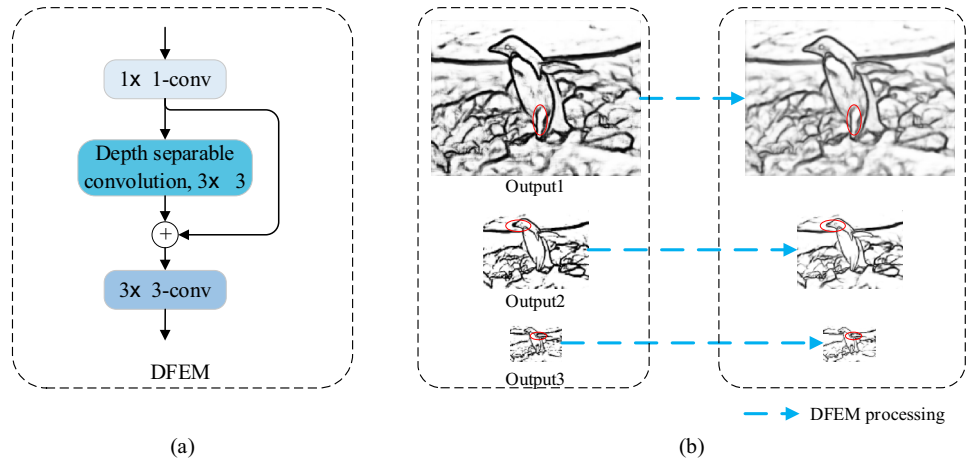
Depth feature extraction module

As shown in Fig. 3, the right part is the new decoding network proposed in this paper. Different from the previous decoding networks, we design a depth feature extraction module (DFEM) in the new decoding network to enhance the overall performance of the model. In previous methods [17, 18], the decoding network adjusts the number of channels through 1×1 -conv after receiving the input from the backbone network and then restores the output feature graphs of different scales to the original image size for fusion through deconvolution or other up-sampling methods, so as to obtain the final contour output. However, in our decoding network, the input from the backbone network is processed by DFEM to further extract and fuse the feature information, while the number of channels is adjusted using 3×3 -conv convolution. Finally, the output feature maps of different scales are resized to the original image size through deconvolution and then fused to obtain the final contour output. The feature information processed by DFEM incorporates more effective details, reducing unnecessary background and texture. This enhancement contributes to an overall improvement in the model’s performance. In the experiments conducted in “Ablation study”, we validated the effectiveness of this module.

As shown in Fig. 7, DFEM is the depth feature extraction module proposed by us. It consists of a 1×1 conventional convolution, a 3×3 conventional convolution, and a 3×3 depth separable convolution. The input from the backbone network is firstly processed by 1×1 -conv to increase the number of channels and then processed by 3×3 depth separable convolution to further extract the feature information. After that, the output of the depth separable convolution is added with the result of 1×1 -conv, and the final output is obtained after a 3×3 convolution.

The calculation formula of DFEM is shown as follows:

Fig. 7 **a** The detailed structure diagram of DFEM. **b** The output results before and after DFEM processing. We use red ellipses for simple marking



$$F_{out} = C_{3 \times 3} * [DSC_{3 \times 3} * (C_{1 \times 1} * Output_i) + C_{1 \times 1} * Output_i]. \tag{6}$$

Among them, F_{out} is the output after DFEM processing, $Output_i (i = 1, 2, 3)$ is the side output of backbone network, $C_{m \times n}$ is the conventional convolution, $m, n = (1, 2, 3, \dots)$ is the size of the convolution kernel, and $DSC_{m \times n}$ is the depth separable convolution, $m, n = (1, 2, 3, \dots)$ is the size of the convolution kernel, “*” represents the convolution.

Loss of function

To illustrate the effectiveness of the method proposed in this paper, we choose the same strategy as the previous method [21], and use the class-balanced cross-entropy loss function to solve the unbalanced distribution of positive and negative samples. The threshold η is introduced to distinguish positive and negative sample sets in consideration of the problem of labels being tagged by multiple people. η is set to 0.2. For a true edge graph $Y = (y_j, j = 1, \dots, |Y|), y_j \in \{0, 1\}$ we define $Y^+ = \{y_j, y_j > \eta\}$ and $Y^- = \{y_j, y_j = 0\}$. However, when $0 < y_j \leq \eta$, this point is considered controversial, so we ignore this point, that is, it does not belong to the positive sample or the negative sample. Y^+ and Y^- represent the positive and negative sample sets. Therefore, $l(\cdot)$ is calculated as follows:

$$l(P, Y) = -\alpha \sum_{j \in Y^-} \log(1 - p_j) - \beta \sum_{j \in Y^+} \log(p_j). \tag{7}$$

In Eq. (3), P represents the predicted contour, and p_j represents the value processed by a sigmoid function at pixel j . $\alpha = \lambda \cdot \frac{|Y^+|}{|Y^+| + |Y^-|}$ and $\beta = \frac{|Y^-|}{|Y^+| + |Y^-|}$ are used to balance the positive and negative samples, and $\lambda (\lambda = 3.0)$ is the weight that controls the coefficient.

As can be seen from Fig. 3, the network uses multiple losses for training. We formulate the total loss as follows:

$$L = \sum_{i=1}^3 (\omega_i \cdot l(P_i, Y)) + \omega_{fuse} \cdot l(P_{fuse}, Y). \tag{8}$$

In the above formula, $\omega_i (i = 1, 2, 3)$ and ω_{fuse} , respectively, represent the weight of loss of three side output results and the weight of loss of final prediction results, P_i represents three different outputs, P_{fuse} represents the final contour prediction, and Y represents the real contour map. $\omega_i = \omega_{fuse} = 0.25$.

Experiment

In this section, we introduce the building environment of the model and the related parameter settings of the model. And experimental analysis is carried out on several publicly available data ets. For example, BSDS500 [23], NYUD [51]. In addition, we validate the effectiveness of the proposed backbone network and depth feature extraction module through ablation experiments. Finally, we compare them with existing lightweight contour detection models and contour detection models based on deep learning.

Datasets

BSDS500 and NYUDv2 are the two publicly available datasets and the most commonly used datasets in the field of contour detection.

As one of the most commonly used datasets in the field of contour detection, the BSDS500 dataset contains a total of 500 images. Among them, there are 200 pictures of the training set, 100 pictures of the verification set, and 200 pictures of the test set. We adopt the same strategy as in [18–22] to enhance the training set and verification set through rotation,

flipping, and random scaling and finally obtain the amplified BSDS500 dataset. In addition, to further enhance the dataset, the researchers mixed the amplified BSDS500 dataset with the flipped PASCAL VOC Context dataset [52] to obtain the mixed training set BSDS500-VOC.

NYUDv2 dataset, like the previous methods [17, 18, 20, 22], we rotated the 381 training pictures, 414 verification pictures, and their corresponding annotation information by four different angles (0, 90, 180, 270) and flipped the rotated results, thus increasing the number of training sets. In addition, because the NYUDv2 dataset contains RGB images and HHA images, we train and test BLCDNet models on the two images, respectively, and finally average the outputs of RGB and HHA as the final contour output. The NYUDv2 dataset has more test images than the BSDS500 dataset, and it contains 654 test images.

Implementation details

Parameter setting

We completed the design of BLCDNet in the PyTorch environment. In the training, we do not use the method of transfer learning to load other model parameters but train the model from the beginning. We used the SGD optimizer to update the parameters, setting the global learning rate to 1×10^{-6} , momentum and weight decay to 0.9 and 2×10^{-4} , respectively. When training on the BSDS500-VOC dataset and the NYUD-v2 dataset, we use the original image size and do not crop. We set the maximum allowable error distance between BSDS500 dataset contour prediction and true contour matching to 0.0075 during the evaluation process according to different datasets. Since the images in NYUD-v2 are larger than those in BSDS500, the maximum allowable error distance is set to 0.011. We used the same loss function as [17, 18, 22] to ensure the fairness of the experiment. All the experiments are conducted on a NVIDIA GeForce3090 GPU with 24GB memory.

Performance metrics

Similar to the previous method [17–22], we first perform non-maximum suppression on the results of network output, so as to obtain the final contour output. We then evaluated the final contours using common evaluation metrics, including the optimal dataset scale (ODS), optimal image scale (OIS) and average precision (AP).

Optimal dataset scale (ODS). The F -score of each image in the dataset is tested at a fixed threshold and the average is calculated. Different average F -score can be computed at different fixed thresholds, and the maximum of all average

F -score is the ODS. The threshold range for calculating F -score is [0,1].

Optimal image scale (OIS). The F -score of each image in the dataset is tested at different thresholds and the maximum F -score corresponding to each image is calculated. At this point, the threshold is also the optimal threshold for the image. OIS is the average of the F -score under the optimal threshold for each image.

Average precision (AP). AP is the average precision between the given threshold ranges [0, 1], and is the area under the precision–recall (PR) curve.

Precision–recall curve. The abscissa and ordinate of the PR curve are Recall and Precision, respectively. Recall and precision are calculated as in Eqs. (11) and (10). PR curve can reflect the classification performance of the model [53].

The F -score is calculated as follows:

$$F\text{-score} = \frac{(P \times R)}{[(1 - \alpha) + \alpha R]} \quad (9)$$

α is the weight, generally 0.5. P and R stand for precision and recall, respectively.

P is calculated as follows:

$$P = \frac{TP}{(TP + FP)} \quad (10)$$

TP and FP represent the correct number and false number of contour pixels.

R is calculated as follows:

$$R = \frac{TP}{(TP + FN)} \quad (11)$$

TP and FN represent the correct number and missed number of contour pixels.

In addition, the recent lightweight method [30–32] tested the parameters, floating-point operations per second (FLOPs), and frame per second (FPS) of the model. To verify the competitiveness of the model, we also tested the parameters, FLOPs, and FPS of BLCDNet in this paper.

Ablation study

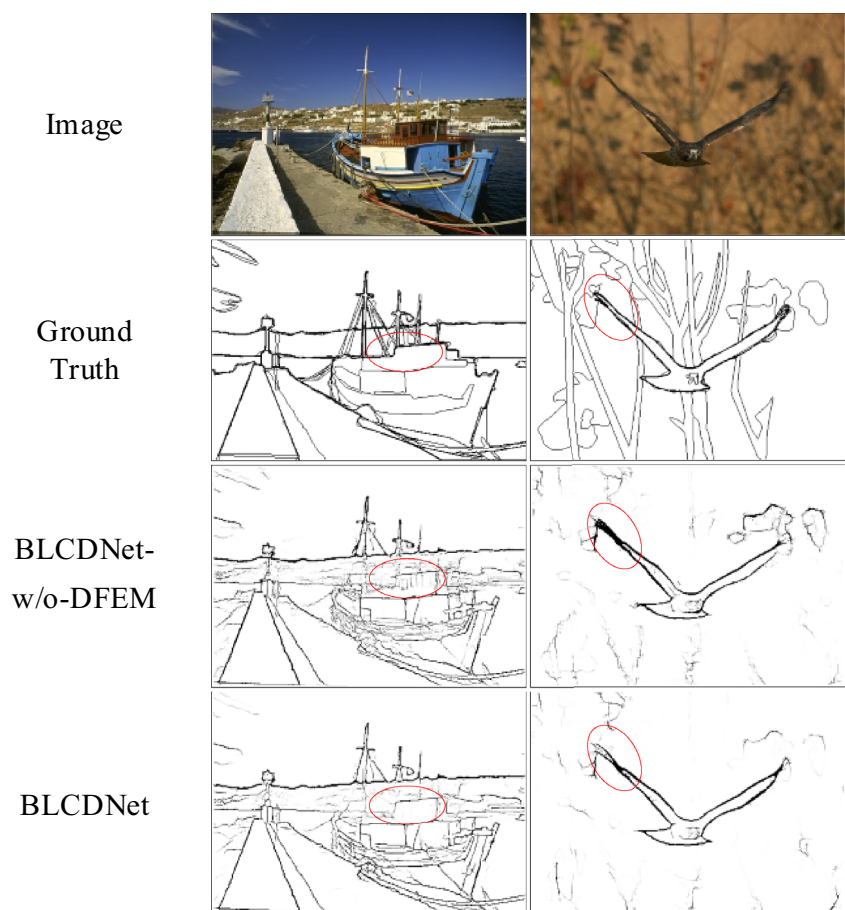
In this section, we conduct a detailed experimental analysis and evaluation of the backbone network of BLCDNet using the BSD500 dataset. First, we trained only the large receptive field feature extraction network (LRF-FENet), the small receptive field feature extraction network (SRF-FENet) and the color confrontation feature extraction network (CC-FENet) under the same conditions and tested their output results. The experimental results are shown in Table 1. Subsequently, we proceeded to train and test the results of

Table 1 Test results of network on BSDS500 without using mixed training set, SS denotes single scale

Method		ODS	OIS	AP
Test results for a single network				
LRF-FENet	SS	0.769	0.783	0.612
SRF-FENet	SS	0.758	0.776	0.610
CC-FENet	SS	0.779	0.796	0.597
The two paths are combined with the results of the test				
LSRF-FENet	SS	0.779	0.795	0.543
LCRF-FENet	SS	0.780	0.794	0.555
SCRf-FENet	SS	0.779	0.796	0.553
The three paths are combined with the results of the test				
BLCDNet	SS	0.784	0.800	0.537
BLCDNet-w/o-DFEM	SS	0.776	0.791	0.531

BLCDNet-w/o-DFEM indicates that the DFEM module is not used

Fig. 8 The output result of BLCDNet and BLCDNET-w/o-DFEM. We make simple markings in the same places with red ovals



combining two different networks. The experimental outcomes are presented in Table 1. Specifically, LSRF-FENet signifies the fusion of the large cell feature extraction network and the small cell feature extraction network; LCRF-FENet denotes the fusion of the large cell feature extraction network and the color confrontation feature extraction network; while SCRf-FENet represents the fusion of the small cell feature

extraction network and the color confrontation feature extraction network. Finally, we trained and tested our entire model in the same way that the three channels in biological vision are parallel. The results show that the three channels are processed in parallel, and the final fusion achieves the best result $ODS = 0.784$.

Table 2 The quantitative comparison results of the proposed method and other methods on BSDS500 test set

Type	Method	ODS	OIS	AP
Bio-inspired contour detection method	BLCDNet-SS	0.799	0.816	0.697
	Tang [8]	0.762	0.778	0.809
	Multiscale integration [54]	0.680	–	–
	SCO [14]	0.670	0.710	0.710
	Contrast dependent [55]	0.630	–	–
	Multifeature based [56]	0.620	–	–
	SED [15]	0.710	0.740	0.740
	Adaptive inhibition [57]	0.580	–	–
Lightweight contour detection method	PiDiNet [32]	0.807	0.823	–
	BLCDNet-SS	0.799	0.816	0.697
	TIN2 [[30]	0.772	0.795	–
	FINED [31]	0.790	0.808	–
	BDCN2 [22]	0.766	0.787	–
	BDCN3 [22]	0.796	0.817	–
	Deep learning contour detection method	DeepContour [58]	0.757	0.776
DeepEdge [59]	0.753	0.772	787	
	HED [17]	0.788	0.808	0.840
	RCF [18]	0.806	0.823	0.839
	CED [19]	0.794	0.811	0.847
	LPCB [20]	0.808	0.824	–
	DRNet [21]	0.802	0.818	0.800
	DSCD [60]	0.813	0.836	0.847
	MI-Net [16]	0.820	0.837	0.873
	LRDNN [39]	0.825	0.840	–
	LLCED [40]	0.805	0.818	–
Non-deep learning contour detection method	gPb [23]	0.729	0.755	0.745
	OEF [38]	0.746	0.770	0.815
	SE [61]	0.743	0.764	0.800
	MCG [62]	0.744	0.777	–
	SCG [37]	0.739	0.758	0.773
	Sketch tokens [63]	0.727	0.746	0.780

Bold shows our results

In addition, we also verify the effectiveness of the proposed DFEM on the BSDS500 dataset, as shown in Table 1 are the results of our experiments. BLCDNet indicates that the DFEM module is used in the decoding network, and BLCDNet-w/o-DFEM indicates that the DFEM module is not used in the decoding network. It can be seen from Table 1 that the performance of BLCDNet using DFEM is higher than that of BLCDNet without DFEM, with ODS exceeding 0.8%. The results show that DFEM blocks achieve further feature extraction and improve the overall performance of the model. Figure 8 is the output result of BLCDNet and BLCDNET-w/o-DFEM. As noted in the red box in Fig. 8,

we can see that the DFEM treatment reduces the texture in the output and adds more useful details.

Comparison with other works

BSDS500

We trained BLCDNet on the BSDS500-VOC hybrid training set and conducted a detailed experimental analysis and evaluation of the test results. We compare the results of BLCDNet with previous contour detection methods, including biologically inspired contour detection methods, lightweight contour detection methods, deep learning contour detection

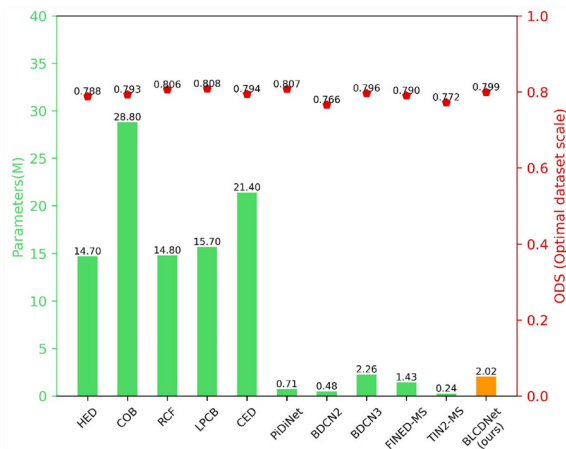


Fig. 9 Performance comparison between the proposed method BLCDNet and some existing contour detection methods. The green axis on the left indicates the number of parameters and corresponds to the bar graph in the figure. The red axis on the right indicates ODS (optimal dataset scale), corresponding to the red fill above the bar chart in the figure

methods, and non-deep learning contour detection methods. For example, Tang [8], multiscale integration [54], SCO [14], contrasts-dependent [55], multifeature based [56], SED [15], adaptive inhibition [57]. PiDiNet [32], FINED [30], TIN2 [31], BDCN2 [22], BDCN3 [22]. DeepContour [58], DeepEdge [59], HED [17], RCF [18], CED [19], LPCB [20], DRNet [21], DSCD [60], MI-Net [16], LRDNN [39], LLCED [40], gPb [23], OEF [38], SE [61], MCG [62], SCG [37], and sketch tokens [63]. In addition, Tang [8], PiDiNet [32], FINED [30] and so on can also be deep learning methods. Table 2 shows the quantitative comparison results between BLCDNet and other methods.

According to the results in Table 2, it can be found that BLCDNet achieves the best result among all biologically inspired contour detection models, with ODS = 0.799, exceeding Tang [8] 3.7%. Combining the results in Table 2, Figs. 9 and 10, it can be seen that BLCDNet also achieves good results among all lightweight models, just below the best PiDiNet. In addition, the results still exceed some deep learning-based contour detection methods when the number of model parameters is small, the calculation is simple and the pre-trained model is not used. ODS exceeds HED and CED by 1.1% and 0.5%, respectively. It further proves that our model has strong competitiveness. The PR curves of our method and other methods are shown in Fig. 11. As can be seen from the figure, our method is closest to the test results of humans and is competitive among all methods. Among them, the vertical coordinate represents the precision rate, and the horizontal coordinate represents the recall rate. The area under the curve represents AP in the performance indicator.

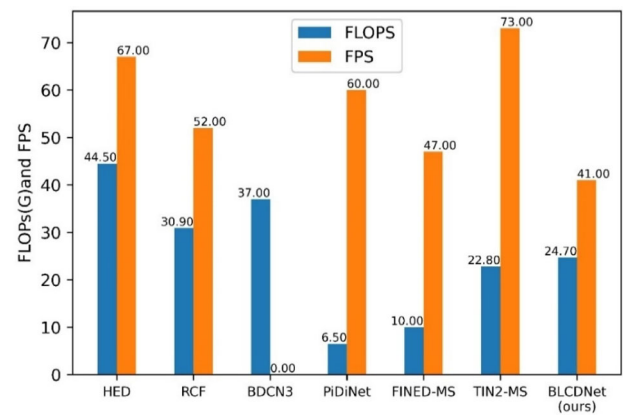


Fig. 10 FPS is the speed we achieved based on the P100. FLOPs are calculated based on a 200×200 image. Some of the results are from other relevant literature [30–32]

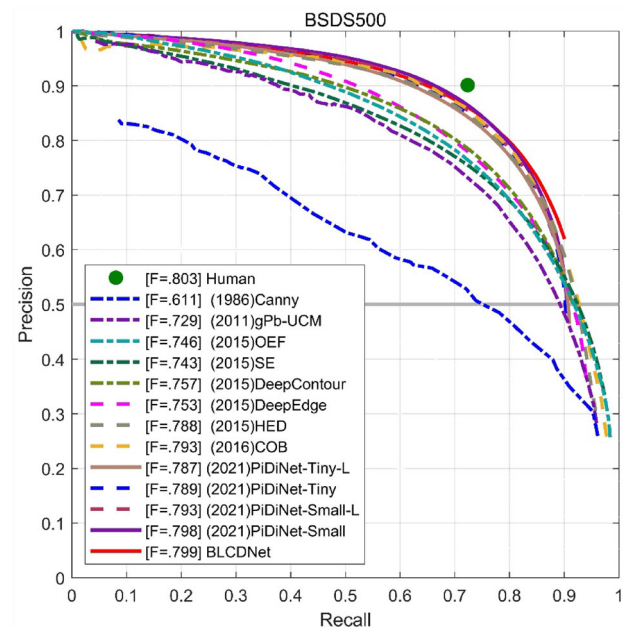


Fig. 11 PR curves of the proposed method and other methods on BSDS500 datasets

NYUD

Like the previous methods [17, 18, 20, 22], we trained our model on RGB images and HHA feature maps, and then tested them, respectively. Finally, the test results were output to obtain RGB, HHA, and RGB-HHA. Where RGB-HHA is the average output of RGB and HHA. We compare the three outputs with results from other methods. For example, gPb-UCM [23], SE [61], gPb + NG [64], SE + NG + [65], OEF [38], HED [17], RCF [18], LPCB [20], TIN2 [31], and PiDiNet [32]. The experimental results are shown in Table 3.

Table 3 Quantitative comparison results between the proposed method and other methods on nyud-v2 test set

Method	Input	ODS	OIS	AP
gPb-UCM [23]	RGB	0.631	0.661	0.562
SE [61]		0.695	0.708	0.719
gPb + NG [64]		0.687	0.716	0.629
SE + NG + [65]		0.706	0.734	0.549
OEF [38]		0.651	0.667	0.653
HED [17]	RGB	0.717	0.732	0.704
	HHA	0.681	0.695	0.674
	RGB-HHA	0.741	0.757	0.749
RCF [18]	RGB	0.729	0.742	0.693
	HHA	0.705	0.715	0.650
	RGB-HHA	0.757	0.771	0.749
LPCB [20]	RGB	0.739	0.754	–
	HHA	0.707	0.719	–
	RGB-HHA	0.762	0.778	–
PiDiNet [32]	RGB	0.733	0.747	–
	HHA	0.715	0.728	–
	RGB-HHA	0.756	0.773	–
TIN1 [31]	RGB	0.706	0.723	–
	HHA	0.661	0.681	–
	RGB-HHA	0.729	0.750	–
TIN2 [31]	RGB	0.729	0.745	–
	HHA	0.705	0.722	–
	RGB-HHA	0.753	0.773	–
BLCDNet	RGB	0.726	0.741	0.696
	HHA	0.703	0.717	0.650
	RGB-HHA	0.751	0.766	0.758

Bold shows our results

Fig. 12 The contour extracted from the NYUD-V2 dataset by our proposed model. From left to right, the original image, HHA features, real contour, BLCDNet extraction results on HHA, BLCDNet extraction results on RGB, and BLCDNet extraction results on RGB-HHA are successively shown

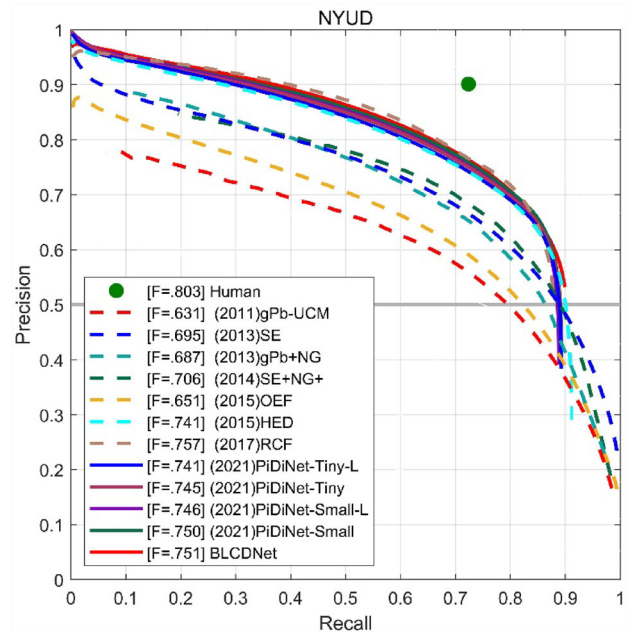
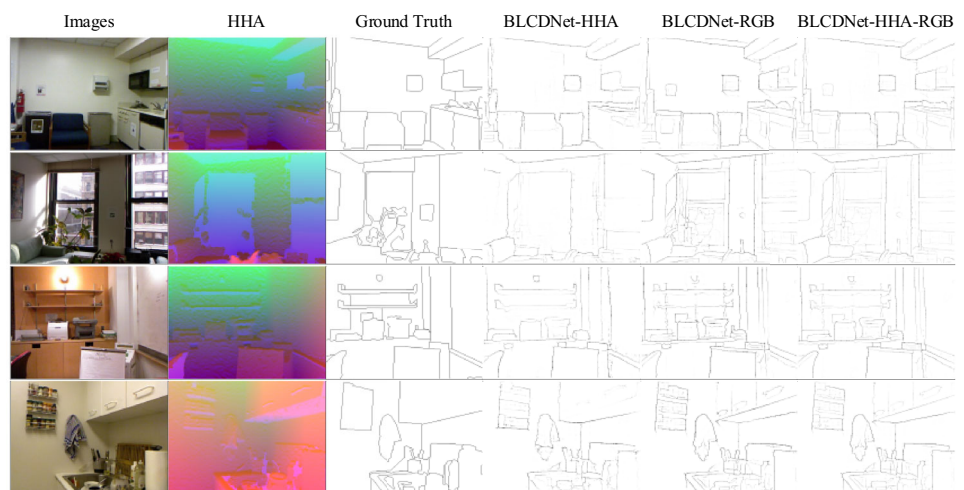


Fig. 13 PR curves of the proposed method and other methods on NYUD datasets

According to the results in Table 3, our method also achieves good performance on the NYUD dataset. It surpasses the results of all biomimetic contour detection models. It also exceeds some deep learning-based methods and lightweight methods, such as HED [17] and TIN1 [31]. This proves that our method shows consistent performance on different data and is more competitive than other methods. Figure 12 is the partial output result of our random selection. It can be seen from the figure that BLCDNet can extract the contour information of the input image relatively completely. Figure 13 shows the PR curves of the proposed method and other methods.

Conclusion

In this paper, we propose a novel biologically inspired lightweight contour detection network, BLCDNet, by combining biological vision mechanisms and convolutional neural networks. We perform experiments and tests on several publicly available datasets, BSDS5000, NYUD, and the results show that BLCDNet obtains an advanced performance among all the biologically inspired models, which is highly competitive among all the deep learning methods. In addition, the combination of biological vision mechanisms also makes BLCDNet more interpretable than other methods, indicating the importance of visual mechanisms for future research. In BLCDNet, we designed the corresponding network structure by simulating three parallel pathways from ganglion cells to V1. We designed a large receptive field network with dilated convolution to simulate the large cell channel from ganglion cells to the V1 region, designed a small receptive field network with conventional convolution to simulate the small cell channel from ganglion cells to the V1 region, and designed a mixed network with conventional convolution and dilated convolution to simulate the color channel from ganglion cells to V1 region. Finally, the combination of the three as the backbone network to achieve full extraction of feature information. In addition, we designed a depth feature extraction module by using deep separable convolution and realize the full fusion of context information by further processing the characteristics of the output of the backbone network. Experiments and tests on publicly available datasets BSDS5000 and NYUD show that our method achieves good performance and has strong competitiveness. It is worth noting that although BLCDNet performs well in all methods, in this paper we pay more attention to the three parallel pathways from ganglion cells to V1, without further exploration and research on the characteristics of neuronal cells in them, which makes the performance of our model limited to a certain extent. In future work, fully considering the overall structure and neuronal properties of visual pathways will be the focus of our study. Furthermore, based on the recent excellent performance of Transformer and the connection between the selectivity mechanism in biological vision systems and the attention mechanism in transformer, it is our future direction to use transformer to improve the proposed method.

Acknowledgements The authors appreciate the anonymous reviewers for their helpful and constructive comments on an earlier draft of this paper.

Funding This work was supported by the National Natural Science Foundation of China (Grant No. 62266006, Grant No. 61866002, Grant No. 62063006), Project of the Key Laboratory of AI and Information Processing (Hechi University), Education Department of Guangxi Zhuang Autonomous Region (2022GXZDSY013). Guangxi Natural Science Foundation (Grant No. 2020GXNSFDA297006, Grant No.

2018GXNSFAA138122, Grant No. 2015GXNSFAA139293), and 2022 Independent Research Project of Guangxi Key Laboratory of Auto Parts and Vehicle Technology (Project No. 2022GKLACVTZZ06).

Data availability statement Data can be obtained from <https://github.com/zwuser1227/BLCDNet>.

Declarations

Conflict of interest The authors declare that they have no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Canny J (1986) A computational approach to edge detection. *IEEE Trans Pattern Anal Mach Intell* 6:679–698
2. Ferrari V, Fevrier L, Jurie F, Schmid C (2007) Groups of adjacent contour segments for object detection. *IEEE Trans Pattern Anal Mach Intell* 30(1):36–51
3. Bertasius G, Shi J, Torresani L (2016) Semantic segmentation with boundary neural fields. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 3602–3610
4. Wang Y, Zhao X, Hu X, Li Y, Huang K (2019) Focal boundary guided salient object detection. *IEEE Trans Image Process* 28(6):2813–2824
5. Sundberg P, Brox T, Maire M, Arbeláez P, Malik J (2011) Occlusion boundary detection and figure/ground assignment from optical flow. In: *CVPR 2011*. IEEE, pp 2233–2240
6. Prewitt JM (1970) Object enhancement and extraction. *Picture Process Psychopictories* 10(1):15–19
7. Sobel I, Feldman G (1968) A 3x3 isotropic gradient operator for image processing. In: *A talk at the Stanford artificial project*, pp 271–272
8. Tang Q, Sang N, Liu H (2019) Learning nonclassical receptive field modulation for contour detection. *IEEE Trans Image Process* 29:1192–1203
9. Yang D, Peng B, Al-Huda Z, Malik A, Zhai D (2022) An overview of edge and object contour detection. *Neurocomputing* 488:470–493
10. Hubel DH, Wiesel TN (1962) Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J Physiol* 160(1):106
11. Jones H, Grieve K, Wang W, Sillito A (2001) Surround suppression in primate V1. *J Neurophysiol* 86(4):2011–2028
12. Solomon SG, Lennie P (2007) The machinery of colour vision. *Nat Rev Neurosci* 8(4):276–286
13. Grigorescu C, Petkov N, Westenberg MA (2003) Contour detection based on nonclassical receptive field inhibition. *IEEE Trans Image Process* 12(7):729–739

14. Yang K-F, Gao S-B, Guo C-F, Li C-Y, Li Y-J (2015) Boundary detection using double-opponency and spatial sparseness constraint. *IEEE Trans Image Process* 24(8):2565–2578
15. Akbarinia A, Parraga CA (2018) Feedback and surround modulated boundary detection. *Int J Comput Vision* 126(12):1367–1380
16. Lin C, Pang X, Hu Y (2023) Bio-inspired multi-level interactive contour detection network. *Digit Signal Process* 141:104155
17. Xie S, Tu Z (2015) Holistically-nested edge detection. In: *Proceedings of the IEEE international conference on computer vision*, pp 1395–1403
18. Liu Y, Cheng M-M, Hu X, Wang K, Bai X (2017) Richer convolutional features for edge detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 3000–3009
19. Wang Y, Zhao X, Huang K (2017) Deep crisp boundaries. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 3892–3900
20. Deng R, Shen C, Liu S, Wang H, Liu X (2018) Learning to predict crisp boundaries. In: *Proceedings of the European conference on computer vision (ECCV)*, pp 562–578
21. Lin C, Cui L, Li F, Cao Y (2020) Lateral refinement network for contour detection. *Neurocomputing* 409:361–371
22. He J, Zhang S, Yang M, Shan Y, Huang T (2019) Bi-directional cascade network for perceptual edge detection. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 3828–3837
23. Arbelaez P, Maire M, Fowlkes C, Malik J (2010) Contour detection and hierarchical image segmentation. *IEEE Trans Pattern Anal Mach Intell* 33(5):898–916
24. Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. *IEEE Trans Pattern Anal Mach Intell* 24(5):603–619
25. Pu M, Huang Y, Liu Y, Guan Q, Ling H (2022) EDTER: edge detection with transformer. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 1402–1412
26. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: *2009 IEEE conference on computer vision and pattern recognition*. IEEE, pp 248–255
27. Gao S-H, Tan Y-Q, Cheng M-M, Lu C, Chen Y, Yan S (2020) Highly efficient salient object detection with 100k parameters. In: *Computer vision—ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, part VI*. Springer, pp 702–721
28. Yu C, Wang J, Peng C, Gao C, Yu G, Sang N (2018) Bisenet: bilateral segmentation network for real-time semantic segmentation. In: *Proceedings of the European conference on computer vision (ECCV)*, pp 325–341
29. Howard AG et al. (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. *arXiv preprint [arXiv:1704.04861](https://arxiv.org/abs/1704.04861)*
30. Wibisono JK, Hang H-M (2020) Fined: fast inference network for edge detection. *arXiv preprint [arXiv:2012.08392](https://arxiv.org/abs/2012.08392)*
31. Wibisono JK, Hang H-M (2020) Traditional method inspired deep neural network for edge detection. In: *2020 IEEE international conference on image processing (ICIP)*. IEEE, pp 678–682
32. Su Z et al. (2021) Pixel difference networks for efficient edge detection. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp 5117–5127
33. Bear M, Connors B, Paradiso MA (2020) *Neuroscience: exploring the brain, enhanced*. Jones & Bartlett Learning, Burlington
34. Zhong H, Wang R (2021) Neural mechanism of visual information degradation from retina to V1 area. *Cogn Neurodyn* 15:299–313
35. Fukushima K (1980) Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol Cybern* 36(4):193–202
36. Martin DR, Fowlkes CC, Malik J (2004) Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans Pattern Anal Mach Intell* 26(5):530–549
37. Xiaofeng R, Bo L (2012) Discriminatively trained sparse code gradients for contour detection. *Adv Neural Inf Process Syst* 25:593–601
38. Hallman S, Fowlkes CC (2015) Oriented edge forests for boundary detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 1732–1740
39. Al-Amaren A, Ahmad MO, Swamy M (2023) A low-complexity residual deep neural network for image edge detection. *Appl Intell* 53(9):11282–11299
40. Fang X-N, Zhang S-H (2023) Learning local contrast for crisp edge detection. *J Comput Sci Technol* 38(3):554–566
41. Nicholls JG, Martin AR, Wallace BG, Fuchs PA (2001) *From neuron to brain*. Sinauer Associates, Sunderland, MA
42. Zhang Q, Lin C, Li F (2021) Application of binocular disparity and receptive field dynamics: a biologically-inspired model for contour detection. *Pattern Recogn* 110:107657
43. Fan X, Jiang M, Shahid AR, Yan H (2022) Hierarchical scale convolutional neural network for facial expression recognition. *Cogn Neurodyn* 16(4):847–858
44. Kandel ER, Schwartz JH, Jessell TM, Siegelbaum S, Hudspeth AJ, Mack S (2000) *Principles of neural science*. McGraw-Hill, New York
45. Purves D et al (2008) *Cognitive neuroscience* (no. 4). Sinauer Associates Inc, Sunderland
46. Stone J (2013) *Parallel processing in the visual system: the classification of retinal ganglion cells and its impact on the neurobiology of vision*. Springer Science & Business Media, Berlin
47. Briggs F, Usrey WM (2011) Corticogeniculate feedback and visual processing in the primate. *J Physiol* 589(1):33–40
48. Garey LJ (1999) Brodmann’s localisation in the cerebral cortex. World Scientific, Singapore
49. Yu F, Koltun V (2015) Multi-scale context aggregation by dilated convolutions. *arXiv preprint [arXiv:1511.07122](https://arxiv.org/abs/1511.07122)*
50. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. *arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)*
51. Silberman N, Hoiem D, Kohli P, Fergus R (2012) Indoor segmentation and support inference from rgbd images. *ECCV* 5(7576):746–760
52. Mottaghi R et al (2014) The role of context for object detection and semantic segmentation in the wild. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 891–898
53. Davis J, Goadrich M (2006) The relationship between precision-recall and ROC curves. In: *Proceedings of the 23rd international conference on machine learning*, pp 233–240
54. Wei H, Lang B, Zuo Q (2013) Contour detection model with multi-scale integration based on non-classical receptive field. *Neurocomputing* 103:247–262
55. Tang Q, Sang N, Liu H (2016) Contrast-dependent surround suppression models for contour detection. *Pattern Recogn* 60:51–61
56. Yang K-F, Li C-Y, Li Y-J (2014) Multifeature-based surround inhibition improves contour detection in natural images. *IEEE Trans Image Process* 23(12):5020–5032
57. Zeng C, Li Y, Li C (2011) Center-surround interaction with adaptive inhibition: a computational model for contour detection. *Neuroimage* 55(1):49–66
58. Shen W, Wang X, Wang Y, Bai X, Zhang Z (2015) Deepcontour: a deep convolutional feature learned by positive-sharing loss for contour detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 3982–3991

59. Bertasius G, Shi J, Torresani L (2015) Deepedge: a multi-scale bifurcated deep network for top-down contour detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4380–4389
60. Deng R, Liu S (2020) Deep structural contour detection. In: Proceedings of the 28th ACM international conference on multimedia, pp 304–312
61. Dollár P, Zitnick CL (2014) Fast edge detection using structured forests. *IEEE Trans Pattern Anal Mach Intell* 37(8):1558–1570
62. Arbeláez P, Pont-Tuset J, Barron JT, Marques F, Malik J (2014) Multiscale combinatorial grouping. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 328–335
63. Lim JJ, Zitnick CL, Dollár P (2013) Sketch tokens: a learned mid-level representation for contour and object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3158–3165
64. Gupta S, Arbelaez P, Malik J (2013) Perceptual organization and recognition of indoor scenes from RGB-D images. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 564–571
65. Gupta S, Girshick R, Arbeláez P, Malik J (2014) Learning rich features from RGB-D images for object detection and segmentation. In: *Computer vision—ECCV 2014: 13th European conference, Zurich, Switzerland, September 6–12, 2014, proceedings, part VII 13*. Springer, pp 345–360

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.