



Bio-inspired XYW parallel pathway edge detection network

Xintao Pang¹, Chuan Lin^{1,*}, Fuzhang Li¹, Yongcai Pan¹

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



ARTICLE INFO

Keywords:

Convolutional neural network
Edge detection
Lightweight methods
Parallel pathway
Biological visual inspiration

ABSTRACT

Edge detection is of critical importance for middle-level and high-level tasks in computer vision. Existing edge detection methods usually use VGG16 as the encoding network and achieve exceptional performance through transfer learning, which has the characteristics of high parameters and high computational cost. Researchers have implemented edge detection by designing decoding networks. Unlike existing methods, this paper is inspired by the effective mechanisms of edge detection in the parallel pathway of biological vision and proposes a new lightweight encoding-decoding structure for edge detection networks, which we refer to as XYW-Net. In the encoding network, we drew inspiration from the receptive field properties of X-type cells, Y-type cells, and W-type cells involved in the parallel pathway, and developed more meticulous convolutional models. Based on the structure of the parallel pathway, the three cell modules are combined into a novel encoding network and excellent feature extraction performance is obtained. Within the decoding network, this paper draws inspiration from the feature integration ability of the inferotemporal cortex (IT) and introduces a novel feature integration module to constitute the decoding network. Experiments show that the proposed network in this paper obtains the Optimal Dataset Scale (ODS) = 0.812 on the BSDS500 dataset with only 0.79 M parameters. Additionally, the model has exhibited outstanding ODS performance on various publicly available edge detection datasets, including NYUD, BIPEDv1, and Multicue. Moreover, future research could concentrate on implementing more efficient antagonistic mechanisms and devising networks grounded in higher-level visual cortical mechanisms. This could further augment the performance and efficiency of edge detection networks. We encourage researchers to delve into these directions within the field. The codes are available at <https://github.com/PXinTa/o/XYW-Net>.

1. Introduction

Edges describe significant information about the shapes and boundaries of objects in an image. Therefore edge information is critical for various intermediate and advanced vision tasks (Jing, Liu, Wang, Zhang, & Sun, 2022), such as target detection and recognition (Kyrou, Tofis, & Theocarides, 2013), image segmentation (Bertasius, Shi, & Torresani, 2016), 3D reconstruction from a 2D image (Santhanam, Doiphode, & Shi, 2023), saliency detection (Tu, Ma, Li, Tang, & Luo, 2020), etc. Traditional edge detection is devoted to studying discontinuous image features (Canny, 1986; Y. Liu, Xie, & Liu, 2020; Prewitt, 1970) to extract edge information, such as changes in brightness, color or texture, and other features. However, these operators are unable to distinguish well between edge information and texture information differences. Later, in order to obtain better edge detection performance,

the edge detection models were designed by some researchers using a single function to simulate the biological vision mechanism (Akbarinia & Parraga, 2018; Spratling, 2012; Tang, Sang, & Liu, 2016), which has a better texture suppression capability compared to the traditional detection operators, and thus achieves better edge detection performance. It should be noted, however, that relying on a single function is not sufficient for fully representing the complex bio-visual mechanism (Tang, Sang, & Liu, 2019). As deep learning has developed, CNNs have shown strong performance in a wide range of vision tasks, and encoding-decoding CNNs (Lin, Zhang, & Hu, 2022; Yun Liu, Cheng, Hu, Wang, & Bai, 2017; Xie & Tu, 2017; J. Yang, Price, Cohen, Lee, & Yang, 2016) have made significant advances in edge detection. However, the above-mentioned edge detection networks use VGG16 as the coding network, which was initially designed for classification tasks, and their models have the characteristics of large parameters and complex

* Corresponding author.

E-mail address: chuanlin@gxust.edu.cn (C. Lin).

¹ Contributing authors: Xintao Pang was responsible for programming and data processing, Chuan Lin was responsible for review & editing, Fuzhang Li and Yongcai Pan participated in discussions and revisions.

structure, and they also need to rely on transfer learning to achieve better performance in training. Later, Tang et al (Tang, et al., 2019), inspired by the perceptual field mechanism of the biological visual system, first proposed a network model combining CNN and biological visual mechanisms to achieve good performance by building a self-designed encoding network with multi-scale fusion. At the same time, neurophysiological research indicates that X, Y, and W cells in the parallel pathways play a crucial role in processing edge information (Shou, 1997). This provides us with inspiration.

Since Hubel et al. (Hubel & Wiesel, 1962) first revealed the functionality of primary visual cortex (V1) neurons in detecting edges and lines, V1 has been widely recognized as the primary cortex involved in edge formation. However, an analysis of the neural mechanisms underlying edge detection suggests that the extraction of edge information requires the collaborative involvement of ganglion cells (GC), the lateral geniculate nucleus (LGN), and the primary visual cortex (V1) (Hubel & Wiesel, 1962). Specifically, GCs are responsible for extracting edge information, while the LGN integrates the information extracted by these cells and transmits it to the cells in the V1 layer for further processing (Zhong & Wang, 2021).

It is worth noting that the process of edge information detection is not accomplished through a single pathway but rather through three parallel visual information pathways composed of X cells, Y cells, and W cells (Shou, 1997), which are referred to as parallel pathways in this paper. Research studies have demonstrated the presence of a center-surround antagonistic mechanism in the perception of X and Y cells, which enhances their sensitivity to edge information (Ankri, Ezra-Tsur, Maimon, Kaushansky, & Rivlin-Etzion, 2020). Additionally, W cells control the micromovements of the eye (Stone & Fukuda, 1974), which contribute to edge detection (Kuang, Poletti, Victor, & Rucci, 2012). These three parallel pathways, consisting of X, Y, and W cells, transmit information to the V1 layer for processing. Ultimately, feature integration occurs in the IT layer, leading to edge detection. The above bio-visual mechanism clearly demonstrates the fundamental role played by the parallel pathway, comprising X, Y, and W cells, along with the IT feature integration mechanisms, in the extraction of edges.

We designed the bio-inspired lightweight CNNs network model-XYW edge detection network-with the above bio-visual mechanism as a guide. Excellent performance is obtained. Our contributions are summarized as follows.

1. We have designed a bio-inspired lightweight edge detection network, drawing inspiration from principles in biological vision, which has yielded exceptional performance. This contribution not only provides novel research avenues but also establishes a bridge between biological vision and computer vision.
2. By incorporating the principles inspired by the receptive field properties of X, Y, and W cells in the parallel pathway, we have developed specific modules for each cell type. These modules were then combined according to the parallel pathway structure, resulting in a novel encoding network. This approach has achieved significant results in extracting edge features.
3. We have been inspired by the feature integration property of the IT cortex and have designed a new feature integration module in the decoding network to achieve better integration of edge features.

The rest of the paper is structured as follows: Section II explains the work related to edge detection. In Section III we describe in detail the relevant bio-visual mechanisms involved in this paper and our proposed method. In Section IV we perform a detailed experimental analysis of the proposed method on the BSDS500 (Arbelaez, Maire, Fowlkes, & Malik, 2010) dataset. In addition we show the results of our experiments on a variety of publicly available datasets and the comparison results with other edge detection methods. In Section V we conclude the full paper with a discussion of the directions worthy of further exploration in this paper.

2. Related work

Traditional methods for edge detection primarily rely on leveraging the local pixel relationships in an image to extract important edge information. These methods consider various factors such as brightness, color, gradient, and other image characteristics. For instance, the well-known Canny operator (Canny, 1986) is commonly used for edge detection. It incorporates Gaussian filtering to reduce noise and smooth the image, followed by the application of a convolution kernel to compute grayscale image gradients, enabling the extraction of edge information.

However, these conventional operators encounter difficulties when dealing with interference from texture information during the extraction of edge information. This interference poses a challenge in extracting meaningful semantic edge information, leading to decreased accuracy in the results. Consequently, meeting the requirements of modern applications that demand reliable edge extraction results for intermediate and advanced vision tasks becomes increasingly challenging.

Biological visual systems possess remarkable accuracy in comprehending complex scenes, which has spurred researchers' interest in designing edge detectors based on these mechanisms. Grigorescu et al. (Grigorescu, Petkov, & Westenberg, 2003) employed the Difference of Gaussians (DoG) operator to simulate the independent suppression of non-classical receptive fields of cells in the primary visual cortex from classical receptive fields. This approach facilitated the edge detection algorithm in achieving texture suppression. In a similar vein, Yang et al. (K. Yang, Gao, Li, & Li, 2013) introduced the CO algorithm, which is founded on the color antagonism mechanism exhibited by receptive fields in optic nerve cells. Their model successfully simulates both red-green and yellow-blue color antagonism, resulting in more stable detection of color edges in natural images. Akbarinia et al. (Akbarinia & Parraga, 2018) proposed four peripheral modulations of the receptive field to obtain edge images. They incorporated a feedback mechanism from neurons in area V2 of the higher visual cortex to area V1 and integrated V2 information with V1 subcortically. Zhang et al. (Zhang, Lin, & Li, 2021) presented a depth-guided dynamic receptive field edge detection model that considers the dynamic properties of optic nerve cell receptive fields and binocular depth properties.

Researchers have explored various approaches for edge detection, including both unsupervised and supervised learning methods. In the domain of unsupervised learning, Martin et al. (Martin, Fowlkes, & Malik, 2004) utilized a logistic regression classifier to predict the probability of boundaries (Pb) by considering brightness, color, and texture features. Building upon this work, Arbelaez et al. (Arbelaez, et al., 2010) proposed the gPb algorithm, which calculates the probability of edge pixels by considering eight different angles for each pixel in the image. Prasad et al. (Prasad, Zisserman, Fitzgibbon, Kumar, & Torr, 2006) employed features such as color and shape to train a Support Vector Machine (SVM) for edge detection.

Supervised learning methods, leveraging the advancements in deep learning, have achieved significant progress in edge detection tasks. Researchers have approached edge detection as a two-classification problem, aiming to classify each pixel in an image. With the development of deep learning in the field of computer vision, many excellent models have been proposed, such as ResNet (K. He, Zhang, Ren, & Sun, 2016), GoogLeNet (Szegedy, et al., 2015), DenseNet (Huang, Liu, Van Der Maaten, & Weinberger, 2017), and Vision Transformer (Dosovitskiy, et al., 2020). These networks are often used as backbone networks in various tasks through transfer learning, achieving outstanding performance. In edge detection tasks, VGG16, ResNet, and Transformers are commonly used as backbones.

Xie et al. (Xie & Tu, 2017), inspired by Fully Convolutional Networks (FCN) (Long, Shelhamer, & Darrell, 2015), introduced Holistically-nested Edge Detection (HED), the first end-to-end CNN for edge detection. They proposed a class-balanced cross-entropy loss function to enhance the performance. Liu et al. (Yun Liu, et al., 2017) further

improved upon HED and presented Richer Convolutional Features (RCF) for edge detection. RCF fused the features obtained after each convolutional layer to generate the final edge output. Wang et al. (Wang, Zhao, & Huang, 2017) addressed the issue of inaccurate and coarse edge localization in the aforementioned models by introducing Deep Crisp Boundaries (CED). CED employed subpixel convolution instead of bilinear interpolation sampling and incorporated layer-by-layer fusion to mitigate the problem of coarse edges. Inspired by Deng (Deng, Shen, Liu, Wang, & Liu, 2018), Cao et al. (Cao, Lin, & Li, 2020), proposed DRNet to achieve a richer feature representation for edge detection. Chen et al. (Chen, Lin, & Qiao, 2022) introduced a dual-pathway edge detection method that concurrently emulates the structures of the first and second visual pathways. They harnessed the SwinTransformer (Z. Liu, et al., 2021) achieving outstanding performance in the field of edge detection. Pu et al. (Pu, Huang, Liu, Guan, & Ling, 2022) used the Transformer architecture to propose a new edge detection method called

EDTER, which has become a state-of-the-art approach in the field of edge detection.

In summary, many existing deep learning-based edge detection networks employ larger models as the encoding network, while emphasizing the design of the decoding network to improve edge detection accuracy. However, adopting larger models as the foundation presents its own set of challenges, including a larger model size, an increased number of parameters, and a more intricate architecture. Furthermore, transfer learning frequently becomes essential during training to attain enhanced performance.

Taking inspiration from traditional edge detection operators, Su et al. (Su, et al., 2021) introduced Pixel Difference Networks for Efficient Edge Detection (PiDiNet). Unlike using large classification networks, PiDiNet proposed a learnable difference convolution operator and constructed a novel network architecture. This approach achieved competitive performance with fewer model parameters and reduced

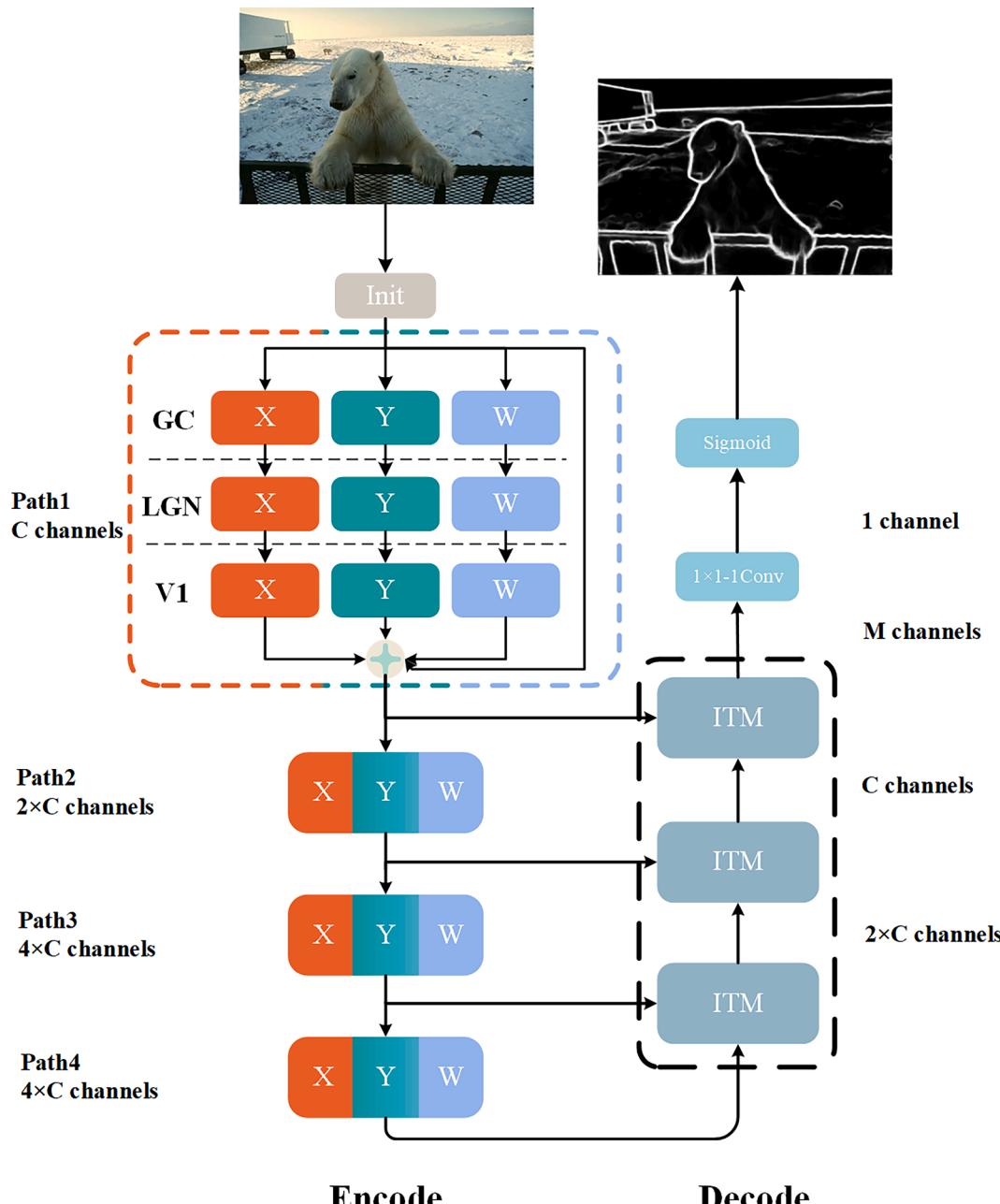


Fig. 1. XYW parallel pathway network, where Path X ($X = 2,3,4$) is the same as the structure of Path1 in the dotted box, and the Init module is a convolutional operation with kernel size = 7, which raises the number of input channels to the number of corresponding pairs of channels in Path1.

computational complexity. Furthermore, Soria et al. (Soria, Sappa, Humanante, & Akbarinia, 2023) have also introduced a method called DexiNed, which achieves excellent edge detection results without the need for transfer learning. These networks trained from scratch bring new insights to the design of edge detection models.

Inspired by the effective mechanisms of edge detection in biological vision and the training of edge detection models from scratch, this paper presents a lightweight edge detection network called XYW-Net. The proposed XYW-Net demonstrates outstanding performance on various publicly available edge detection datasets without the need for transfer learning.

3. Proposed method

Fig. 1 illustrates the model proposed in this paper, which comprises an encoding network that inspires the parallel pathways of biological vision and a decoding network that inspires the integration properties of IT cortical features. The encoding network consists of three parallel pathways, each consisting of three identical cell modules. These modules correspond to the cells processed by the GC-LGN-V1 information pathway in the parallel pathway. The information extracted from the encoding network is then integrated by the ITM to obtain the final contour information. Moreover, by leveraging the method of designing a multi-scale network, we stack this process to enhance the performance of the network. A MaxPooling operation is applied between the two paths.

The physiological mechanisms and design principles of the parallel pathway and the X, Y, and W modules in the encoding network are described in detail in subsection 3.1. The decoding network is designed based on the inspiration of the IT cortex, which possesses feature integration properties. It aims to extract the edge information generated from the integration of the parallel encoding network. The decoding network will be discussed extensively in subsection 3.2.

3.1. Encoding network

3.1.1. Structure design inspired by parallel pathway

Research has revealed that visual information in the visual system is processed in parallel through pathways known as parallel pathways (Shou, 1997). As depicted in **Fig. 2(a)**, visual information is transmitted concurrently through three pathways, each composed of X, Y, and W cells. These pathways enable the transmission of visual information from

the retina to the visual cortex via the LGN, facilitating a comprehensive analysis of complex visual information. In this paper, we aim to inspire the mechanisms of these parallel pathways and integrate the proposed cellular modules to construct a parallel network framework, as illustrated in **Fig. 2(b)**, thereby establishing the coding network for our research.

3.1.2. Module design inspired by the receptive field antagonistic mechanism of X cells

The X cells in the X pathway are classified based on the spatio-temporal summation properties according to Robson et al. (Enroth-Cugell & Robson, 1966). Among the two types of cells in the cat's GC, those that conform to the Rodieck model (Rodieck, 1965) are referred to as X cells. The central region of X cells' receptive field exhibits significantly higher peak sensitivity compared to the surrounding regions, as illustrated in **Fig. 3(a)**. The receptive field of X cells shows a center-surround antagonistic organization in the form of concentric circles. It is precisely this antagonistic mechanism (Ankri, et al., 2020) that equips them with the ability to extract boundaries during image processing.

In this study, we have devised an X-cell module inspired by the physiological mechanisms of X-cells, as discussed earlier and illustrated in **Fig. 4**. Differing from conventional bio-inspired edge detection methods that directly employ Gaussian function templates, as proposed by Rodieck (Rodieck, 1965), to simulate center-surround antagonism, our approach utilizes trainable convolutional kernels for emulating the receptive field of X-cells. The computational process unfolds as follows: when given the input features $F \in \mathbb{R}^{C \times H \times W}$ (where C denotes the number of channels, and H and W represent the input's height and width), the translation is as follows:

$$R_{Xc} = \text{ReLU}(F * K_{Xc}) R_{Xs} = \text{ReLU}(F * K_{Xs}) * K_{1 \times 1}$$

Here, R_{Xc} signifies the feature response of the center mechanism, R_{Xs} represents the feature after undergoing the surround response, K_{Xc} stands for simulating the center mechanism, K_{Xs} simulates the surround mechanism, $K_{1 \times 1}$ denotes the point convolution designed to adjust inter-channel information, and $\text{ReLU}(\cdot)$ stands for the activation function.

The center and surround responses demonstrate antagonistic properties. A 3×3 convolutional kernel aggregates peripheral information towards the center. On the other hand, a 1×1 convolutional output retains solely the information at the same spatial location as the input's center. We obtain the antagonistic response of X-cells, denoted as X_A , by subtracting the features acquired from the surround-to-center

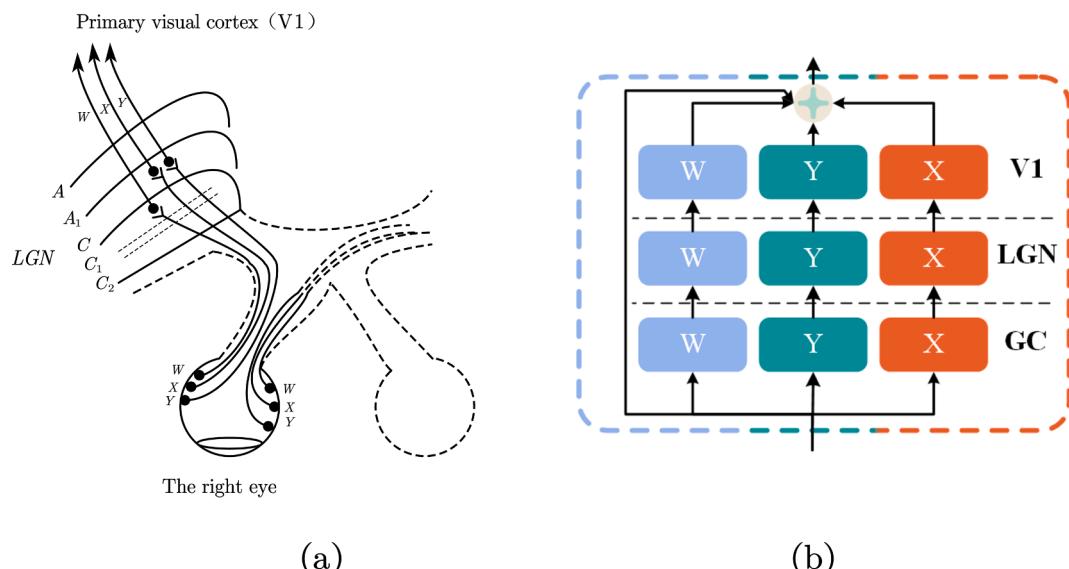


Fig. 2. (a) Parallel pathway (Redrawing from literature (Shou, 1997)). (b) Simulated Parallel Pathway Framework.

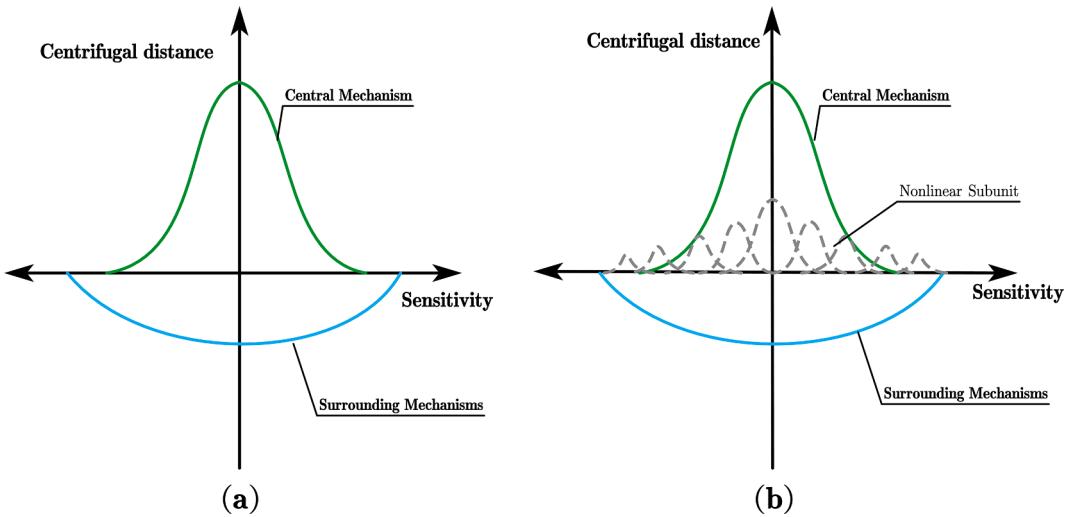


Fig. 3. (a)receptive field model of X cell, (b)receptive field model of Y cell.

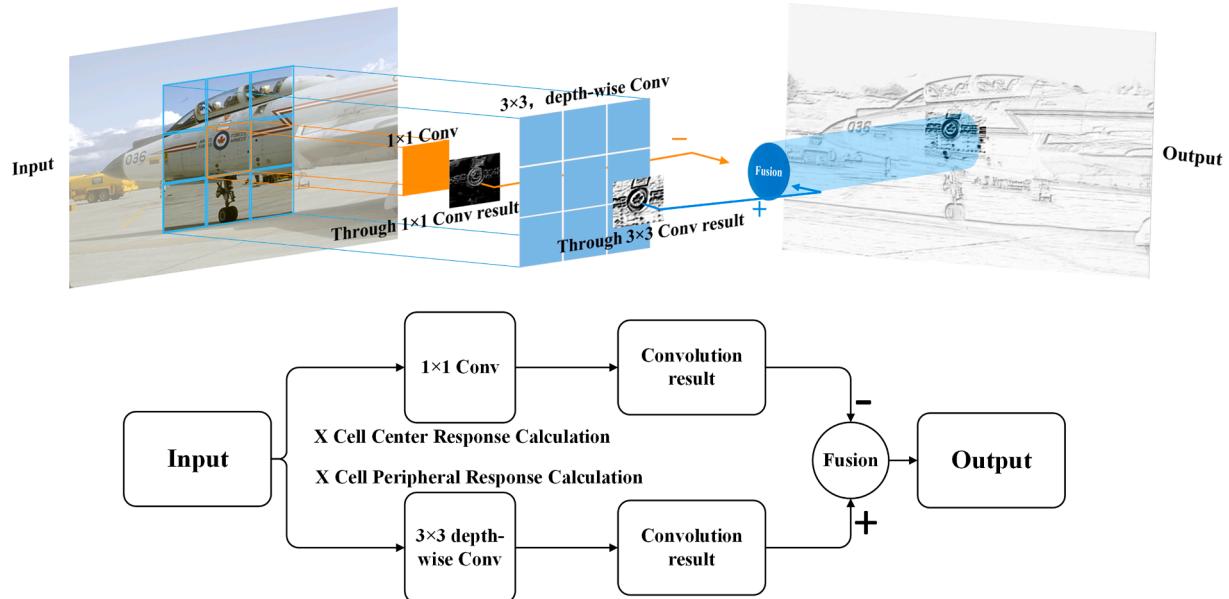


Fig. 4. Biologically inspired X cell module.

convolution. The calculation process is outlined as follows:

$$X_A = R_{X_S} - R_{X_C}$$

3.1.3. Module design inspired by the receptive field antagonistic mechanism of Y cells

The Y cells in the Y-pathway are classified based on the time-space summation property, as described by Robson et al. (Enroth-Cugell & Robson, 1966). Unlike the X cells, the Y cells in the ganglion cells (GC) of the cat do not satisfy the Rodieck model (Rodieck, 1965). The receptive field of Y cells exhibits an antagonistic pattern similar to that of X cells. However, in addition to the antagonistic receptive field, Y cells also possess a wide distribution of nonlinear subunits with rectifying characteristics across their receptive field. Furthermore, Y cells have a larger receptive field size compared to X cells, as illustrated in Fig. 3(b). These nonlinear subunits in the receptive field lead to a nonlinear response, enabling Y cells to exhibit higher contrast sensitivity than X cells in visual function (Shou, 1997). The presence of these properties allows Y cells to specialize in detailed edge extraction.

Inspired by the physiological mechanisms of Y-type cells mentioned

above, we designed a Y-cell module as shown in Fig. 5. In the Y-cell model, the simulation of the surround-center antagonistic mechanism follows the same methodology as in the X-cell model, denoted as $Y_A = R_{Y_S} - R_{Y_C}$. However, the Y-cell module encompasses a broader receptive field and distinctive non-linear subunit properties. To emulate the peripheral response mechanism within Y-cells, we employ a 5x5 dilated convolution K_{Y_S} , for processing input features. The remaining computational process remains unaltered. The adoption of a 5x5 dilated convolution is chosen to reflect the fact that Y-cells possess a larger receptive field than X-cells. Dilated convolutions capture pixels that are not adjacent to the center pixel, and as the distance between the surround pixels and the center pixel increases, the correlation between them decreases. Therefore, by dilating the convolution, we represent the non-linear subunit properties within Y-cells.

3.1.4. Module design inspired by the receptive field microsaccade mechanism of W cells

In the W pathway, W cells are a type of cell that has been discovered by subsequent researchers (Saito, 1983). The axons of W cells are

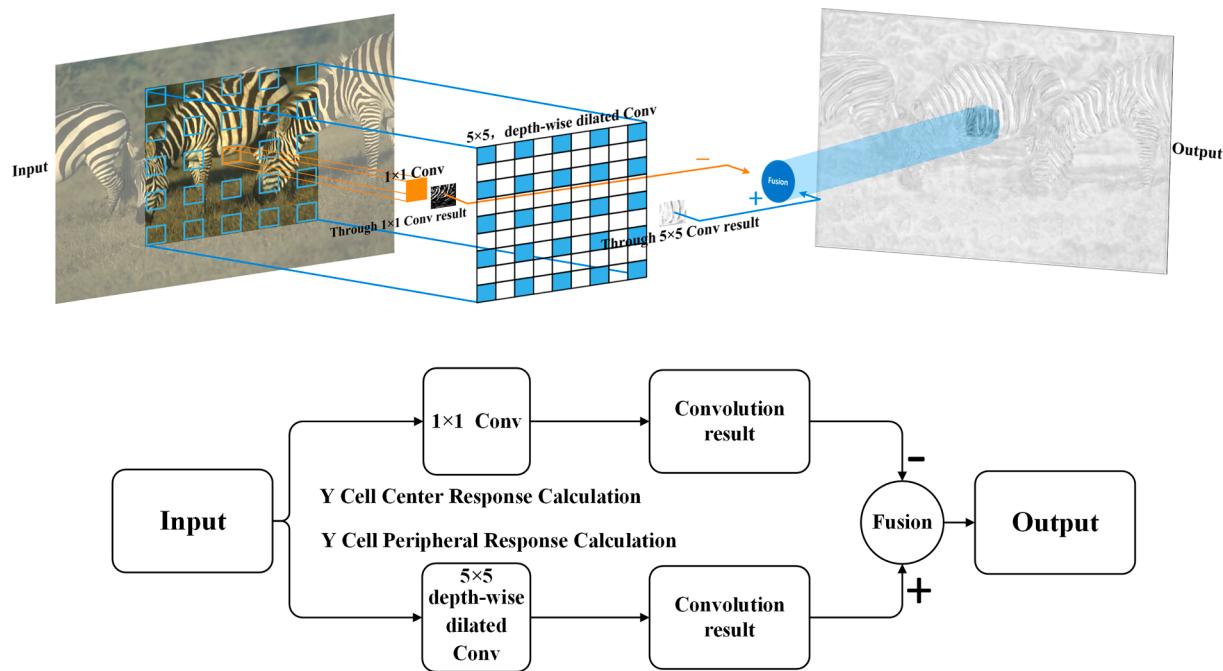


Fig. 5. Biologically inspired Y cell module.

connected to the superior colliculus of the midbrain and play a role in controlling micro eye movements (Stone & Fukuda, 1974). Previous studies (Martinez-Conde, Macknik, & Hubel, 2004) have identified three types of micro eye movements, namely Tremor, Drifts, and micro-saccades. Among them, Tremor refers to non-periodic wave-like movements of the eye within a specific frequency range. Drifts and tremors occur simultaneously to maintain accurate fixation in the absence of visual stimulation. Research has shown (Kuang, et al., 2012) that W-cells, which control eye movements, contribute to edge detection. In this paper, the term “micro-motion” mainly refers to microsaccades, which describe high-speed, small-amplitude eye movements during fixation. Microsaccades enable the eye to capture surrounding information while maintaining attention on a specific point. As depicted in Fig. 6, the dashed circular shape represents the captured information when the eye is fixating, while the solid cross shape represents the information acquired after the occurrence of microsaccades.

Inspired by the aforementioned physiological characteristics, we have designed the W-cell module, as illustrated in Fig. 7. To mimic the W-cells’ ability to attend to surrounding information through microsaccades, we employ 1×3 and 3×1 convolutions. When provided with

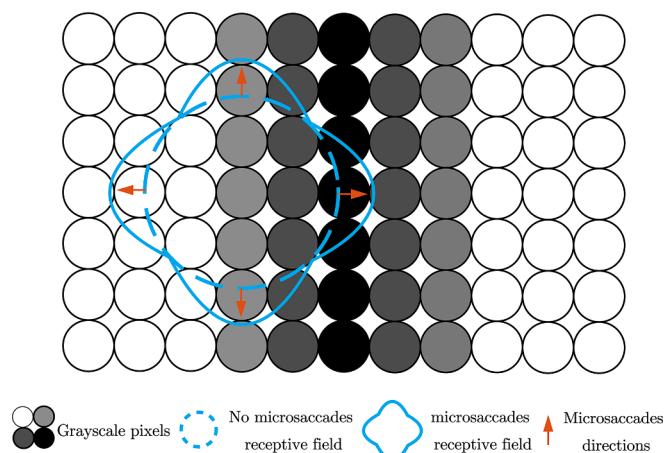


Fig. 6. Microsaccades receptive field schematic.

feature $F \in \mathbb{R}^{C \times H \times W}$, the calculation process of cellular micro-motion response unfolds as follows.

$$W = \text{ReLU}((\text{ReLU}(F * K_{1 \times 3}) * K_{1 \times 1}) * K_{3 \times 1}) * K_{1 \times 1}$$

The dashed rectangle in the figure represents the center during non-microsaccade states, while the solid rectangles depict the flickering state. By utilizing convolutional operations, the information from the solid rectangles is pooled into the dashed rectangle. Through two irregular convolutions, the output forms a cross-like shape, representing the projection of two orthogonal receptive fields. This simulation effectively captures the properties of flickering receptive fields, enabling the capture of information from the surrounding region.

3.2. Decoding network

The inferior temporal gyrus is one of the three gyri in the temporal lobe, situated below the middle temporal gyrus and connected to the inferior occipital gyrus. It extends along the outer inferior border to the ventral surface of the temporal lobe, constrained by the inferior sulcus. This region represents a higher level of the ventral visual pathway responsible for processing visual information. As the final stage of the ventral pathway, it integrates information such as color and shape from other visual cortices (Loffler, 2008), playing a crucial role in biological object recognition. Inspired by the characteristics of the IT cortex, this study introduces the IT module as an integral part of the decoding network, as depicted in Fig. 8.

To achieve a more accurate simulation of the localization and integration characteristics of edge information in the IT cortex, we have devised a design for edge localization convolution, as depicted in Fig. 9. Edges typically exhibit a gradual slope transition and undergo zero-crossing phenomena through second-order differentiation, which play a vital role in edge localization (Gonzalez, 2009). Hence, we have introduced the Edge Localization Convolution (ELC) within the ITM that emulates the IT cortex, aiming to enhance the decoding network’s capability to extract edge features and better simulate the IT cortex’s localization and integration properties. When given features $F_H \in \mathbb{R}^{C \times H \times W}$, $F_L \in \mathbb{R}^{C_0 \times H_0 \times W_0}$, where $H > H_0$, $W > W_0$, the calculation process is as follows.

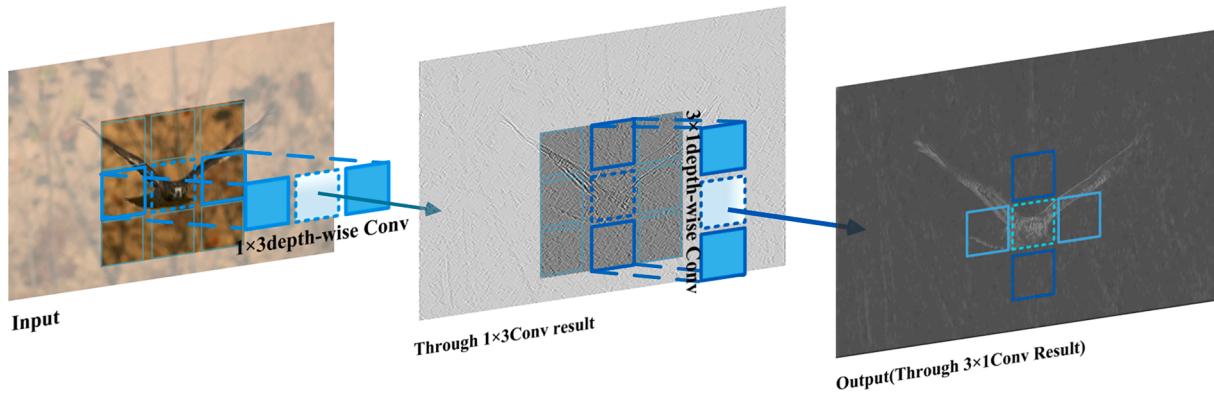


Fig. 7. Biologically Inspired W Cell Module.

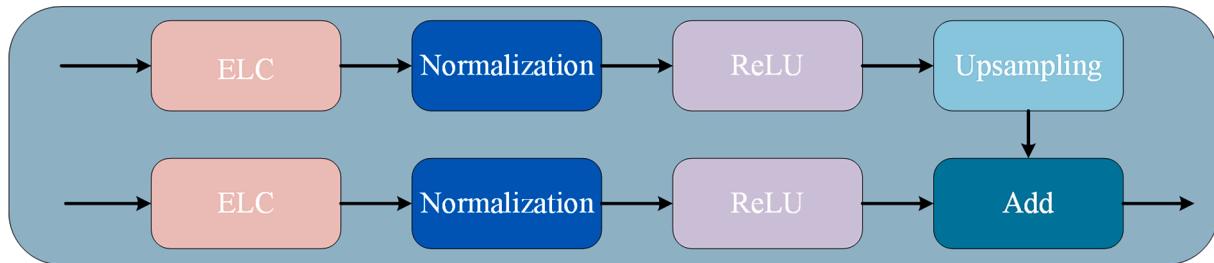


Fig. 8. IT Module(ITM).

| | | |
|--------------------|--------------------|--------------------|
| $X_1 + X_3 - 2X_5$ | $X_2 + X_6 - 2X_5$ | $X_3 + X_9 - 2X_5$ |
| $X_4 + X_2 - 2X_5$ | 0 | $X_6 + X_8 - 2X_5$ |
| $X_7 + X_1 - 2X_5$ | $X_8 + X_4 - 2X_5$ | $X_9 + X_7 - 2X_5$ |

Fig. 9. Edge Localization Convolution (ELC).

$$E_H = \text{ReLU}(\text{Norm}(F_H * K_{ELC})) \\ E_L = \text{UpSample}(\text{ReLU}(\text{Norm}(F_L * K_{ELC})))$$

Where E_H represents the result processed through the high-resolution pathway, and E_L represents the result processed through the low-resolution pathway. K_{ELC} corresponds to the operation ELC, which also ensures that the two features maintain a consistent number of channels upon output. The $\text{ReLU}(\cdot)$ activation function introduces non-linearity to the network. $\text{Norm}(\cdot)$ signifies the normalization operation that can expedite network convergence. $\text{UpSample}(\cdot)$ denotes the upsampling operation, which aligns the low-resolution features with the input's

high-resolution features. By summing these components, we attain E_F . In the ablation experiments, we will explore the impact of ELC and normalization on the network's performance

$$E_F = E_L + E_H$$

4. Experiment

In this paper, we evaluate and train our model using several well-known datasets in the field of edge detection: BSDS500 dataset ([Arbelaez, et al., 2010](#)), NYUD dataset ([Silberman, Hoiem, Kohli, & Fergus, 2012](#)), Multicue dataset ([Mély, Kim, McGill, Guo, & Serre, 2016](#)), and BIPEDv1 dataset ([Poma, Riba, & Sappa, 2020](#)).

The BSDS500 dataset is commonly used for evaluating the performance of edge detection models. It consists of 200 training images, 100 validation images, and 200 test images. In this paper, we follow the approach of ([Cao, et al., 2020](#); [Deng, et al., 2018](#); [Xie & Tu, 2017](#)) and augment the data by incorporating the PASCAL VOC dataset, thereby expanding the training data.

The NYUD dataset comprises 1449 images, including 381 images in the training set, 414 images in the validation set, and 654 images in the test set. Following previous work ([Cao, et al., 2020](#); [Yun Liu, et al., 2017](#)), we train and test on RGB images and HHA (horizontal disparity, height above ground, and angle with gravity) images independently. The outputs of RGB and HHA are averaged to obtain the final edge output. The HHA training exploits depth information encoded into three channels, which can be visualized as a colored picture.

The Multicue dataset consists of 100 challenging natural scenes, with each scene having two frame sequences from the left and right views. Annotations for low-level edges and object boundaries are provided for the last frame of the left sequence. In accordance with ([Su, et al., 2021](#)), we randomly divide the 100 images into 80 training images and 20 test images. The training images are enhanced using a method similar to the one used for the BSDS500 dataset. We perform three independent training tests and average the evaluated results to obtain the final result.

The BIPEDv1 dataset contains 250 outdoor images with a resolution

of 1280×720 . Each image has a single expert annotation for edges, ensuring high accuracy. Following (Poma, et al., 2020), we use 200 images for training and the remaining 50 images for testing.

4.1. Experimental details

In this chapter, we used a computer with RAM: 128 GB, GPU: NVIDIA GeForce GTX 1080TI 11 GB, CPU: Intel Xeon E5-2650 V4 to train the model. The code is implemented with Python and we used PyTorch deep learning framework (Imambi, Prakash, & Kanagachidambaresan, 2021). Four datasets were used: BSDS500, NYUD, Multicue, and BIPED. The models were optimized using the Adam (Kingma & Ba, 2014) optimizer with an initial learning rate set to 1×10^{-3} . The batch size = 1, and the input images are of the original size. The loss function used in this paper for the real contour map Y and the network predicted contour \hat{Y} is

$$L(\hat{Y}, Y) = -\beta \sum_{i \in Y^-} \log(1 - \hat{y}_i) - (1 - \beta) \sum_{i \in Y^+} y_i \log(\hat{y}_i)$$

where $\beta = |Y^-|/|Y^- + Y^+|$ is used as a weighting factor to balance the positive and negative samples, Y^+ and Y^- represent the set of contour pixels and background pixels in the real contour map Y respectively, \hat{y}_i and y_i are the pixel values in the predicted edge map and the real contour map respectively, and $y_i \in (0 - 1)$ is the second term makes the contour annotation considered as a soft label.

For the BSDS500 dataset and BIPEDv1 dataset, the model was trained for 6 iterations on the entire training data. Starting from the 6th iteration, the learning rate was reduced to 0.1 times the value of the previous iteration. As for the NYUD dataset and Multicue dataset, the model underwent 20 iterations on all training data. The learning rate was reduced to 0.1 times the previous iteration at the 10th and 15th iterations.

In evaluating the model performance, we used metrics such as Average Precision (AP), F-measure at Optimal Dataset Scale (ODS), and F-measure at Optimal Image Scale (OIS). These metrics were used to assess the accuracy and effectiveness of the model. Additionally, we introduced the concept of floating point operations (FLOPs) and the parametric number of Params to analyze the model's efficiency in terms of computation and size. The number of Params directly reflects the size of the model, while FLOPs measure the computational workload during data processing. We also considered the model's frames per second (FPS) as a performance measure.

4.2. Ablation study

We conducted a comprehensive experimental analysis and evaluation of the XYW-Net on the hybrid BSDS-VOC dataset. Firstly, we examine the influence of the completeness of our proposed parallel pathway structure by investigating various combinations of the three modules on network performance. All experiments adhere to the experimental protocols outlined in Section 4.1 of the BSDS500 dataset, and the results are documented in Table 1. (X-M, Y-M, W-M represent the respective cell modules proposed in this study).

The results depicted in Table 1 reveal a consistent improvement in the overall performance of the model as the parallel pathways are

Table 1
Parallel Pathway Integrity Exploration Experiment.

| X- M | Y- M | W- M | ODS | OIS | AP |
|------|------|------|-------|-------|-------|
| ✓ | ✗ | ✗ | 0.798 | 0.818 | 0.862 |
| ✗ | ✓ | ✗ | 0.796 | 0.814 | 0.861 |
| ✗ | ✗ | ✓ | 0.793 | 0.812 | 0.856 |
| ✓ | ✓ | ✗ | 0.804 | 0.822 | 0.868 |
| ✓ | ✗ | ✓ | 0.799 | 0.816 | 0.862 |
| ✗ | ✓ | ✓ | 0.799 | 0.817 | 0.862 |
| ✓ | ✓ | ✓ | 0.812 | 0.827 | 0.873 |

incrementally established. Specifically, the complete parallel pathway surpasses any combination of two parallel pathways, while any combination of two parallel pathways outperforms a single pathway. Thus, this finding serves as evidence of the efficacy of the parallel pathway mechanism inspired by biological vision.

To assess the efficacy of the X, Y, and W cellular modules within the parallel pathway in the code, we substituted them with 3x3 convolution kernels, denoted as "Origin" for replacement. The structure of the parallel pathway remained unchanged, as did the experimental details. The experimental outcomes are documented in Table 2 Encode. These results clearly demonstrate a significant advantage of the three cellular models proposed in this paper over the direct utilization of 3x3 convolution. This finding supports the notion that the models inspired by cellular sensory fields, as presented in this paper, exhibit strong performance in edge detection. Moreover, the effectiveness of the three modules is substantiated.

To investigate the effectiveness of the proposed Edge Localization Convolution operator in the decoding network, we replaced the ELC with a 3x3 convolution kernel, indicated as "Common" for replacement manipulation. We maintained the experimental details unchanged and recorded the experimental results in Table 2 Decode. The experimental findings affirm the exceptional performance of the ELC proposed in this paper within the network architecture outlined in this study.

The convergence of a network can be expedited through the use of normalization techniques. In this paper, we investigate the impact of different normalizations on the network's convergence. Specifically, we examine the effects of Batch Normalization (Ioffe & Szegedy, 2015), Instance Normalization (Ulyanov, Vedaldi, & Lempitsky, 2016), and Group Normalization (Wu & He, 2018) on network convergence. The number mentioned within brackets for Group Normalization indicates the number of channel groupings employed. To assess the convergence effect, we conduct the same number of training iterations for each normalization technique. Subsequently, we analyze how different normalizations influence the convergence speed of this paper, based on the ODS evaluation metric, whether it is high or low. We maintain consistency in all experimental details, and the corresponding results are documented in Table 3.

The results presented in Table 3 clearly indicate that Instance Normalization outperforms other normalization techniques in terms of accelerated convergence within the same number of iterations in this paper's model. Consequently, the decoding network in this paper adopts Instance Normalization as a normalization technique to expedite the convergence of the model on the dataset.

4.3. Comparison with other state-of-the-art models

Model size and performance. The proposed method aims to achieve both parameter lightweighting and excellent detection performance. To evaluate its effectiveness, this paper compares it with two types of models: transfer learning-based methods (TL-based methods) and lightweight deep learning methods (Lightweight methods). The models used for comparison include HED (Xie & Tu, 2017), CED (Wang, et al., 2017), BDCN (J. He, Zhang, Yang, Shan, & Huang, 2019), FINED (Wibisono & Hang, 2020a), TIN2 (Wibisono & Hang, 2020a), and PiDiNet (Su, et al., 2021). The experimental results on the BSDS500 dataset are summarized in Table 4.

Based on the data presented in Table 4, the model proposed in this

Table 2

Explore the effectiveness of X, Y, and W cell modules as well as ELC.

| Location | Method | ODS | OIS | AP |
|----------|--------|-------|-------|-------|
| Encode | Origin | 0.800 | 0.818 | 0.844 |
| | XYW | 0.812 | 0.827 | 0.873 |
| Decode | Common | 0.807 | 0.824 | 0.866 |
| | ELC | 0.812 | 0.827 | 0.873 |

Table 3

Exploring the impact of different normalizations in this paper's model.

| Normalization | ODS | OIS | AP |
|------------------------|-------|-------|-------|
| Batch Normalization | 0.799 | 0.816 | 0.851 |
| Group Normalization(2) | 0.806 | 0.824 | 0.868 |
| Group Normalization(3) | 0.806 | 0.824 | 0.864 |
| Group Normalization(6) | 0.807 | 0.825 | 0.864 |
| Instance Normalization | 0.812 | 0.827 | 0.873 |

Table 4

Model size and performance experiments. Calculated based on 200×200 images.

| Method | | ODS | Params (M) | FLOPs (G) | FPS |
|--------------------|--------------------------------|-------|------------|-----------|-----------------|
| TL-base Method | HED (Xie & Tu, 2017) | 0.788 | 14.72 | 24.3 | 54 [†] |
| | CED (Wang, et al., 2017) | 0.806 | 21.80 | 60.8 | 22 [†] |
| | BDCN (J. He, et al., 2019) | 0.820 | 16.30 | 37.0 | 23 [†] |
| Lightweight Method | DPED (Chen, et al., 2022) | 0.823 | 67.9 | 63.4 | 5 [†] |
| | FINED (Wibisono & Hang, 2020a) | 0.772 | 1.43 | 22.8 | 26 [†] |
| | TIN2 (Wibisono & Hang, 2020b) | 0.788 | 0.24 | 10.0 | 47 [†] |
| | PiDiNet (Su, et al., 2021) | 0.807 | 0.72 | 6.5 | 42 [†] |
| | XYW-Net (Ours) | 0.812 | 0.79 | 6.3 | 27 [†] |

paper achieves an ODS of 0.812 with FLOPs of 6.3G, which demonstrates the best performance among the lightweight models. When compared to the TL-based method BDCN (ODS = 0.820), our method achieves only a 1% lower performance while utilizing only 5.4% of the parameters. Furthermore, it requires less computational effort (6.3G vs. 37.0G FLOPs). [†]The experimental data were measured on the NVIDIA-Tesla T4 provided with the Colab platform.

BSDS500 dataset. In this paper, we compared our proposed network with various previous edge detection models that were tested on the BSDS500 dataset. These models include biologically inspired approaches such as Tang (Tang, et al., 2019), Multiscale integration (Wei, Lang, & Zuo, 2013) SCO (K.-F. Yang, Gao, Guo, Li, & Li, 2015), Contrast-dependent (Tang, et al., 2016), Multifeature-based (K.-F. Yang, Li, & Li, 2014), SED (Akbarinia & Parraga, 2018), and Adaptive inhibition (Zeng, Li, & Li, 2011). We also compared our model with lightweight models like PiDiNet (Su, et al., 2021), TIN2 (Wibisono & Hang, 2020b), FINED (Wibisono & Hang, 2020a), BDCN2 (J. He, et al., 2019), BDCN3 (J. He, et al., 2019), CHRNet (Elharrouss, El Fallah-Seghrouchni, 2023). Additionally, transfer learning-based methods including DeepContour (Shen, Wang, Wang, Bai, & Zhang, 2015), DeepEdge (Bertasius, Shi, & Torresani, 2015), HED (Xie & Tu, 2017), RCF (Yun Liu, et al., 2017), CED (Wang, et al., 2017), LPCB (Deng, et al., 2018), DRNet (Cao, et al., 2020), PCL-Net (Xuan, Huang, Liu, & Du, 2022), BDCN (J. He, et al., 2019), UAED (Zhou, et al., 2023), EDETER (Pu, et al., 2022), DPED (Chen, et al., 2022) were considered. Furthermore, we compared our model with non-deep learning approaches like gPb (Arbelaez, et al., 2010), OEF (Hallman & Fowlkes, 2015), SE (Dollár & Zitnick, 2014), MCG (Arbeláez, Pont-Tuset, Barron, Marques, & Malik, 2014), SCG (Xiaofeng & Bo, 2012), Sketch tokens (Lim, Zitnick, & Dollár, 2013). The results of these comparisons are summarized in Table 5 and Fig. 11. According to the results, the proposed network in this paper exhibits exceptional edge detection performance compared to CHRNet and PiDiNet. However, when compared to larger models based on transfer learning, both our model and CHRNet lag behind in terms of ODS and OIS scores, especially when compared to state-of-the-art. Nevertheless, considering the perspective of parameter count and computational complexity, our lightweight model remains competitive, which is a focal point of our research too. Therefore, in subsequent comparisons and analyses, we will pay particular attention to lightweight methods.

Table 5

Comparison with other methods on BSDS500 dataset.

| Type | Method | ODS | OIS | AP |
|---|---|-------|-------|-------|
| Bio-inspired edge Detection method | Tang (Tang, et al., 2019) | 0.762 | 0.778 | – |
| | Multiscale integration (Wei, et al., 2013) | 0.680 | – | – |
| | SCO (K.-F. Yang, et al., 2015) | 0.670 | 0.710 | – |
| | Contrast-dependent (Tang, et al., 2016) | 0.630 | – | – |
| | Multifeature-based (K.-F. Yang, et al., 2014) | 0.620 | – | – |
| | SED (Akbarinia & Parraga, 2018) | 0.710 | 0.740 | – |
| | Adaptive inhibition (Zeng, et al., 2011) | 0.580 | – | – |
| | XYW-Net (Ours) | 0.812 | 0.827 | 0.873 |
| | TIN2 (Wibisono & Hang, 2020b) | 0.772 | 0.795 | – |
| | FINED (Wibisono & Hang, 2020a) | 0.790 | 0.808 | – |
| Lightweight edge detection method | BDCN2 (J. He, et al., 2019) | 0.766 | 0.787 | – |
| | BDCN3 (J. He, et al., 2019) | 0.796 | 0.817 | – |
| | PiDiNet (Su, et al., 2021) | 0.807 | 0.823 | – |
| | CHRNet (Elharrouss, et al., 2023) | 0.787 | 0.788 | – |
| | XYW-Net (Ours) | 0.812 | 0.827 | 0.873 |
| | DeepContour (Shen, et al., 2015) | 0.757 | 0.776 | 0.790 |
| | DeepEdge (Bertasius, et al., 2015) | 0.753 | 0.772 | 0.787 |
| | HED (Xie & Tu, 2017) | 0.788 | 0.808 | 0.840 |
| | RCF (Yun Liu, et al., 2017) | 0.806 | 0.823 | – |
| | CED (Wang, et al., 2017) | 0.794 | 0.811 | 0.847 |
| Deep learning edge detection method | LPCB (Deng, et al., 2018) | 0.808 | 0.824 | – |
| | DRNet (Cao, et al., 2020) | 0.802 | 0.818 | 0.800 |
| | BDCN (J. He, et al., 2019) | 0.820 | 0.838 | 0.888 |
| | PCL-Net (Xuan, et al., 2022) | 0.815 | 0.834 | – |
| | EDTER (Pu, et al., 2022) | 0.832 | 0.847 | 0.886 |
| | DPED (Chen, et al., 2022) | 0.823 | 0.840 | 0.832 |
| | UAED-EfficientNet (Zhou, et al., 2023) | 0.838 | 0.855 | 0.902 |
| | UAED-VGG16 (Zhou, et al., 2023) | 0.820 | 0.840 | 0.889 |
| | XYW-Net (Ours) | 0.812 | 0.827 | 0.873 |
| | gPb (Arbelaez, et al., 2010) | 0.729 | 0.755 | 0.745 |
| Non-deep learning edge detection method | OEF (Hallman & Fowlkes, 2015) | 0.746 | 0.770 | 0.817 |
| | SE (Dollár & Zitnick, 2014) | 0.743 | 0.764 | 0.800 |
| | MCG (Arbeláez, et al., 2014) | 0.744 | 0.777 | – |
| | SCG (Xiaofeng & Bo, 2012) | 0.739 | 0.758 | – |
| | Sketch tokens (Lim, et al., 2013) | 0.727 | 0.757 | – |
| | XYW-Net (Ours) | 0.812 | 0.827 | 0.873 |

To be specific, our XYW-Net achieves ODS = 0.812 and OIS = 0.827, surpassing TIN2 by 5.1% and 4.4% respectively; outperforming FINED by 2.8% and 2.2%; exceeding BDCN3 by 2.7% and 1.8%; demonstrating improvements of 0.62% and 0.48% over PiDiNet; and showcasing advancements of 3.3% and 5.1% over CHRNet. Taking into account the information in Table 4, the model presented in this paper achieves outstanding comprehensive performance on the BSDS500 dataset. Fig. 10 shows the output of this paper and PiDiNet.

NYUD dataset. we compared our model with several edge detection models that were tested on the NYUD dataset. These models include HED (Xie & Tu, 2017), BDCN (J. He, et al., 2019), PiDiNet (Su, et al., 2021), TIN1 (Wibisono & Hang, 2020b), CHRNet (Elharrouss, et al.,

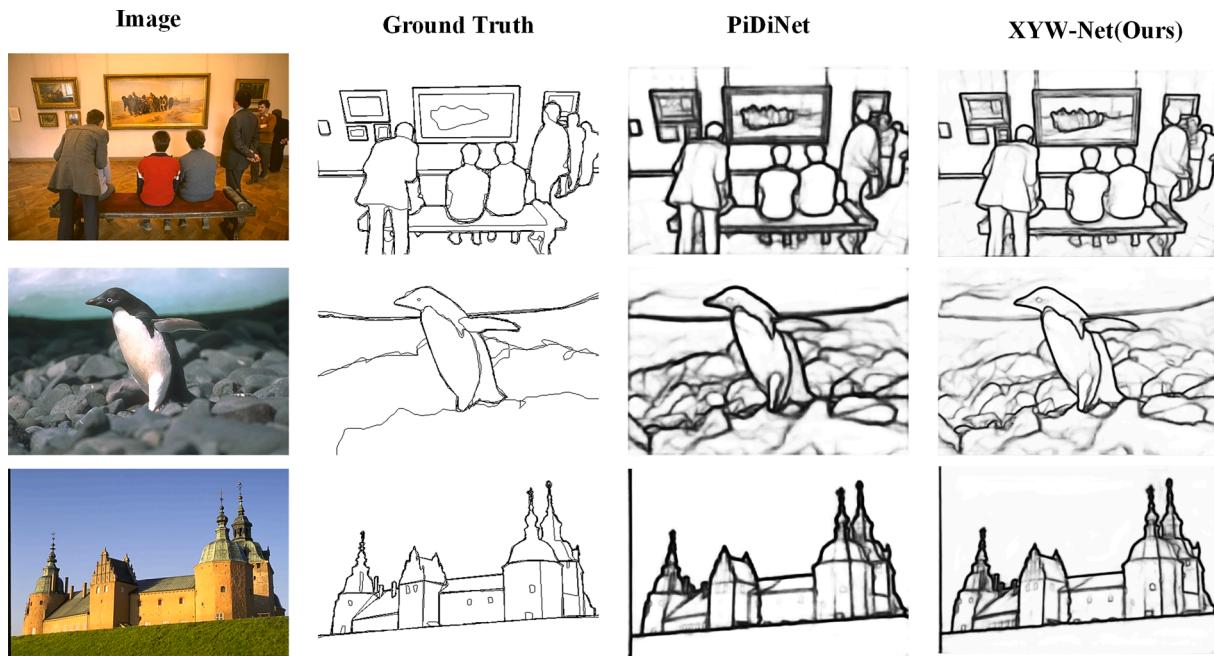


Fig. 10. A qualitative comparison of network outputs with PiDiNet. It is obvious from the figure that our method outputs smoother and finer edge information compared to the lightweight PiDiNet.

2023) and TIN2 (Wibisono & Hang, 2020b). The results of the comparisons are summarized in Table 6 and Fig. 12. According to the results, the RGB-HHA model proposed in this paper achieves the outstanding performance, with an ODS score slightly lower than that of BDCN and CHRNet. It shows comparable performance to PiDiNet in terms of ODS, but outperforms PiDiNet in terms of OIS. It is important to note that HHA images, which represent depth information as color rendering maps, deviate from natural scene images. Additionally, the HHA feature map is prone to strong noise interference, leading to distortion and deformation of object boundaries. As a result, the performance of the bio-inspired model in this paper may be compromised on the HHA feature map.

Overall, considering the performance of our method compared to other advanced models, it demonstrates competitive results on the NYUD dataset.

BIPEDv1 dataset. Introduced by Poma et al. (Poma, et al., 2020), is a dataset specifically designed for edge detection. It addresses the issue of

Table 6
Comparison with other methods on NYUD dataset.

| Method | Input | ODS | OIS | FLOPs(G) |
|-----------------------------------|---------|-------|-------|----------|
| HED (Xie & Tu, 2017) | RGB | 0.717 | 0.732 | 24.3 |
| | HHA | 0.681 | 0.695 | |
| | RGB-HHA | 0.741 | 0.771 | |
| BDCN (J. He, et al., 2019) | RGB | 0.748 | 0.763 | 37.0 |
| | HHA | 0.707 | 0.719 | |
| | RGB-HHA | 0.765 | 0.781 | |
| TIN1 (Wibisono & Hang, 2020b) | RGB | 0.706 | 0.723 | – |
| | HHA | 0.661 | 0.681 | |
| | RGB-HHA | 0.729 | 0.750 | |
| TIN2 (Wibisono & Hang, 2020b) | RGB | 0.729 | 0.745 | 10.0 |
| | HHA | 0.705 | 0.722 | |
| | RGB-HHA | 0.753 | 0.773 | |
| PiDiNet (Su, et al., 2021) | RGB | 0.733 | 0.747 | 6.5 |
| | HHA | 0.715 | 0.728 | |
| | RGB-HHA | 0.756 | 0.773 | |
| CHRNet (Elharrouss, et al., 2023) | RGB | 0.730 | 0.737 | – |
| | HHA | 0.710 | 0.719 | |
| | RGB-HHA | 0.757 | 0.769 | |
| XYW-Net (Ours) | RGB | 0.730 | 0.747 | 6.3 |
| | HHA | 0.701 | 0.715 | |
| | RGB-HHA | 0.756 | 0.775 | |

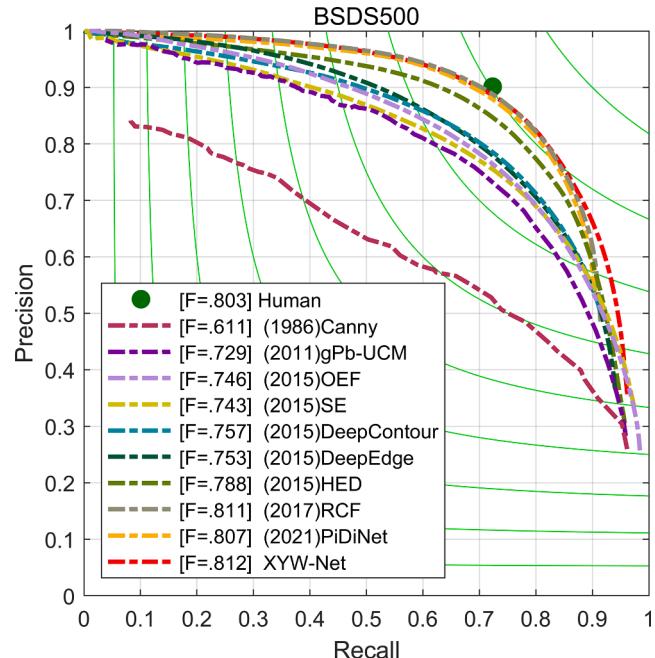


Fig. 11. Precision-Recall curves of our models and some competitors on BSDS500 dataset.

mislabeling caused by multiple annotations by providing annotations from a single expert. The performance of several edge detection models has been evaluated on this dataset, including SED (Akbarinia & Parraga, 2018), HED (Xie & Tu, 2017), CED (Wang, et al., 2017), RCF (Yun Liu, et al., 2017), BDCN (J. He, et al., 2019), and DexiNed (Poma, et al., 2020). In this paper, we compared our proposed method with the aforementioned edge detection models on the BIPEDv1 dataset. The results are recorded in Table 7. Based on the results, our method outperforms the other models on the BIPEDv1 dataset. Compared to the latest DexiNed detection method, our method achieves a 3.3% improvement in both ODS and OIS, as well as a 2.2% improvement in

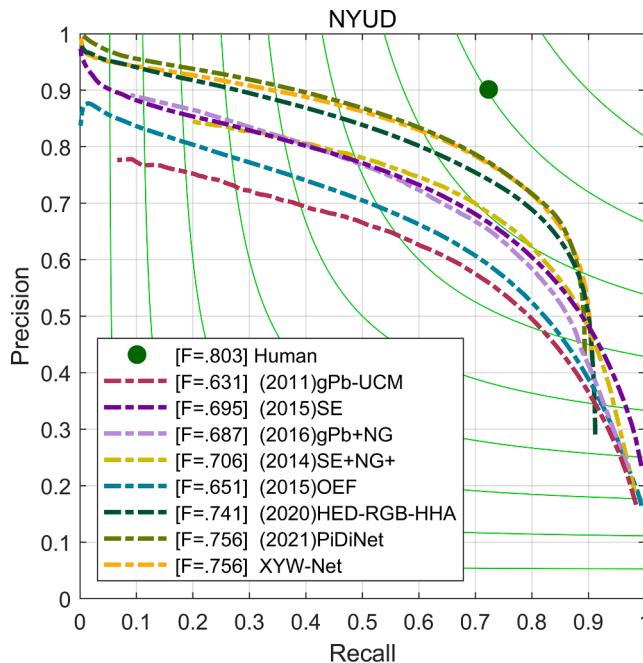


Fig. 12. Precision-Recall curves of our models and some competitors on NYUD dataset.

Table 7
Comparison with other methods on BIPEDv1 dataset.

| Method | ODS | OIS | AP |
|--------------------------------|--------------|--------------|--------------|
| SED(Akbarinia & Parraga, 2018) | 0.717 | 0.731 | 0.756 |
| HED(Xie & Tu, 2017) | 0.829 | 0.847 | 0.869 |
| CED(Wang, et al., 2017) | 0.795 | 0.815 | 0.830 |
| RCF(Yun Liu, et al., 2017) | 0.843 | 0.859 | 0.882 |
| BDCN(J. He, et al., 2019) | 0.839 | 0.854 | 0.887 |
| DexiNed(Poma, et al., 2020) | 0.859 | 0.867 | 0.905 |
| XYW-Net(Ours) | 0.887 | 0.896 | 0.925 |

AP. When compared to the migration-dependent edge detection model BDCN, our method shows even greater improvements, with a 5.7% improvement in ODS, a 4.9% improvement in OIS, and a 4.3% improvement in AP.

Overall, considering the results on the BIPEDv1 dataset, the proposed bio-inspired model in this paper demonstrates excellent performance.

Multicue dataset. Our paper compares the proposed XYW model with previous edge detection models, including Multicue (Mély, et al., 2016), HED (Xie & Tu, 2017), RCF (Yun Liu, et al., 2017), BDCN (J. He, et al., 2019), and PiDiNet (Su, et al., 2021). The experimental results are recorded in Table 8. Based on the experimental results, the XYW model shows excellent performance when the label is “edge”. It achieves an ODS that is 2% lower than BDCN, but the model stability is comparable to BDCN. However, when it comes to the model with the label “boundary”, the performance of XYW deviates from its performance on the “edge” label. In computer vision, the terms “edge” and “object boundaries” are often used interchangeably, but in biological vision, there is a strict distinction between the two (Xie & Tu, 2017). “Edge” refers to low-level edge information formed in the V1 region, while “object boundaries” refer to high-level edge information formed in the V4 region. The Multicue dataset provides both low-level “edge” and high-level “boundary” labeling, making it unique among publicly available datasets. Our XYW model is designed based on the parallel pathway of biological vision, specifically targeting the extraction of low-level edge information. Therefore, it performs well on the low-level edges of the Multicue dataset but lacks in performance when it comes

Table 8
Comparison with other methods on Multicue dataset.

| Method | Low-level edge | | Object boundary | |
|------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| | ODS | OIS | ODS | OIS |
| Multicue(Mély, et al., 2016) | 0.830 (0.002) | – (0.014) | 0.720 (0.014) | – (0.008) |
| HED(Xie & Tu, 2017) | 0.851 (0.014) | 0.864 (0.011) | 0.814 (0.011) | 0.822 (0.008) |
| RCF(Yun Liu, et al., 2017) | 0.857 (0.004) | 0.862 (0.004) | 0.817 (0.004) | 0.825 (0.005) |
| BDCN(J. He, et al., 2019) | 0.891 (0.001) | 0.898 (0.002) | 0.836 (0.001) | 0.846 (0.003) |
| PiDiNet(Su, et al., 2021) | 0.855 (0.007) | 0.860 (0.005) | 0.818 (0.003) | 0.830 (0.005) |
| XYW-Net(Ours) | 0.871 (0.003) | 0.877 (0.002) | 0.804 (0.006) | 0.810 (0.008) |

to high-level boundary information. Taking all these factors into consideration, the model proposed in this paper demonstrates good performance on the Multicue dataset.

5. Conclusion and discussion

This manuscript introduces XYW-Net, an encoding-decoding architecture for edge detection that draws inspiration from biological paradigms and boasts of lightweight attributes. The encoding network takes cues from the parallel pathway mechanism, meticulously emulating the receptive field mechanisms of X, Y, and W cells via convolution operations. Meanwhile, the decoding network incorporates an ELC module, strategically devised to augment the precision of edge localization and information extraction. The remarkable efficacy of XYW-Net is substantiated through its exceptional performance across four diverse benchmark edge detection datasets, effectively underscoring its propensity for exceptional generalization. Additionally, this study illuminates several promising avenues that beckon further exploration.

I. Augmenting Computational Efficiency: The utilization of the feature subtraction antagonism mechanism, while effective, is accompanied by certain computational limitations. Future research endeavors could revolve around the pursuit of more efficient adversarial mechanisms, aimed at elevating computational efficiency. **II. Thorough Scrutiny of Physiological Mechanisms:** Although the design of the encoding network draws inspiration from the physiological mechanisms of the primary visual cortex, its purview remains restricted to that developmental stage. Subsequent research could embark on a more profound investigation into the intricate mechanisms of the advanced visual cortex, thereby proffering avant-garde and more streamlined models for edge detection.

In conclusion, we believe that the contributions of this work can stimulate novel ideas at the intersection of computer vision and biological vision, providing fresh insights into structural considerations.

Declaration of Competing Interest

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

Data availability

<https://github.com/PXinTao/XYW-Net>

Acknowledgements

The authors appreciate the anonymous reviewers for their helpful and constructive comments on an earlier draft of this paper. This work was supported by the National Natural Science Foundation of China

(Grant No.62266006), Guangxi Natural Science Foundation (Grant No.2020GXNSFDA297006), National Natural Science Foundation of China (Grant No.61866002), Guangxi Natural Science Foundation (GrantNo.2018GXNSFAA138122, Grant No.2015GXNSFAA139293), 2022 Autonomous Research Project of Guangxi Key Laboratory of Automobile Components and Vehicle Technology (No. 2022GKLACVTZ06) and The Basic Ability Enhancement Program for Young and Middle-aged Teachers of Guangxi (2023KY0356).

References

- Akbarinia, A., & Parraga, C. A. (2018). Feedback and surround modulated boundary detection. *International Journal of Computer Vision*, 126, 1367–1380.
- Ankri, L., Ezra-Tsur, E., Maimon, S. R., Kaushansky, N., & Rivlin-Etzion, M. (2020). Antagonistic center-surround mechanisms for direction selectivity in the retina. *Cell reports*, 31, Article 107608.
- Arbelaez, P., Maire, M., Fowlkes, C., & Malik, J. (2010). Contour detection and hierarchical image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 33, 898–916.
- Arbeláez, P., Pont-Tuset, J., Barron, J. T., Marques, F., & Malik, J. (2014). Multiscale combinatorial grouping. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 328–335).
- 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).
- 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).
- Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, 679–698.
- Cao, Y.-J., Lin, C., & Li, Y.-J. (2020). Learning crisp boundaries using deep refinement network and adaptive weighting loss. *IEEE Transactions on Multimedia*, 23, 761–771.
- Chen, Y., Lin, C., & Qiao, Y. (2022). DPED: Bio-inspired dual-pathway network for edge detection. *Frontiers in Bioengineering and Biotechnology*, 10, 1876.
- 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).
- Dollár, P., & Zitnick, C. L. (2014). Fast edge detection using structured forests. *IEEE Transactions on pattern analysis and machine intelligence*, 37, 1558–1570.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., & Gelly, S. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv: 2010.11929*.
- Elharrouss, O., Hmamouche, Y., Idrissi, A. K., El Khamlichi, B., & El Fallah-Seghrouchni, A. (2023). Refined edge detection with cascaded and high-resolution convolutional network. *Pattern Recognition*, 138, Article 109361.
- Enroth-Cugell, C., & Robson, J. G. (1966). The contrast sensitivity of retinal ganglion cells of the cat. *The Journal of physiology*, 187, 517–552.
- Gonzalez, R. C. (2009). *Digital image processing*. Pearson education India.
- Grigorescu, C., Petkov, N., & Westenberg, M. A. (2003). Contour detection based on nonclassical receptive field inhibition. *IEEE Transactions on Image Processing*, 12, 729–739.
- Hallman, S., & Fowlkes, C. C. (2015). Oriented edge forests for boundary detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1732–1740).
- 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).
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700–4708).
- Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160, 106.
- Imambi, S., Prakash, K. B., & Kanagachidambaresan, G. (2021). PyTorch. In *Programming with TensorFlow* (pp. 87–104). Springer.
- Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448–456). PMLR.
- Jing, J., Liu, S., Wang, G., Zhang, W., & Sun, C. (2022). Recent advances on image edge detection: A comprehensive review. *Neurocomputing*.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kuang, X., Poletti, M., Victor, J. D., & Rucci, M. (2012). Temporal encoding of spatial information during active visual fixation. *Current Biology*, 22, 510–514.
- Kyrkou, C., Tofios, C., & Theoccharides, T. (2013). A hardware architecture for real-time object detection using depth and edge information. *ACM Transactions on Embedded Computing Systems (TECS)*, 13, 1–19.
- Lim, J. J., Zitnick, C. L., & 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).
- Lin, C., Zhang, Z., & Hu, Y. (2022). Bio-inspired feature enhancement network for edge detection. *Applied Intelligence*, 1–16.
- 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).
- Liu, Y., Xie, Z., & Liu, H. (2020). An Adaptive and Robust Edge Detection Method Based on Edge Proportion Statistics. *IEEE Transactions on Image Processing*, 29, 5206–5215.
- Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 10012–10022).
- Loffler, G. (2008). Perception of contours and shapes: Low and intermediate stage mechanisms. *Vision research*, 48, 2106–2127.
- Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431–3440).
- Martin, D. R., Fowlkes, C. C., & Malik, J. (2004). Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26, 530–549.
- Martinez-Conde, S., Macknik, S. L., & Hubel, D. H. (2004). The role of fixational eye movements in visual perception. *Nature Reviews Neuroscience*, 5, 229–240.
- Mély, D. A., Kim, J., McGill, M., Guo, Y., & Serre, T. (2016). A systematic comparison between visual cues for boundary detection. *Vision Research*, 120, 93–107.
- Poma, X. S., Riba, E., & Sappa, A. (2020). Dense extreme inception network: Towards a robust cnn model for edge detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 1923–1932).
- Prasad, M., Zisserman, A., Fitzgibbon, A., Kumar, M. P., & Torr, P. H. (2006). Learning class-specific edges for object detection and segmentation. In *Computer vision, graphics and image processing* (pp. 94–105): Springer.
- Prewitt, J. M. (1970). Object enhancement and extraction. *Picture processing and Psychopictorics*, 10, 15–19.
- 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).
- Rodeck, R. W. (1965). Quantitative analysis of cat retinal ganglion cell response to visual stimuli. *Vision Research*, 5, 583–601.
- Saito, H. A. (1983). Morphology of physiologically identified X-, Y-, and W-type retinal ganglion cells of the cat. *Journal of Comparative Neurology*, 221, 279–288.
- Santhanam, H., Doiphode, N., & Shi, J. (2023). Automated line labelling: dataset for contour detection and 3D reconstruction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 3136–3145).
- 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).
- Shou, T. (1997). *Brain mechanisms of visual information processing*. Shanghai: Shanghai Scientific & Technological Education Publishing House.
- Silberman, N., Hoiem, D., Kohli, P., & Fergus, R. (2012). Indoor segmentation and support inference from rgbd images. In *European conference on computer vision* (pp. 746–760): Springer.
- Soria, X., Sappa, A., Humanante, P., & Akbarinia, A. (2023). Dense extreme inception network for edge detection. *Pattern Recognition*, 139, Article 109461.
- Spratling, M. W. (2012). Image segmentation using a sparse coding model of cortical area V1. *IEEE Transactions on Image Processing*, 22, 1631–1643.
- Stone, J., & Fukuda, Y. (1974). Properties of cat retinal ganglion cells: A comparison of W-cells with X-and Y-cells. *Journal of Neurophysiology*, 37, 722–748.
- Su, Z., Liu, W., Yu, Z., Hu, D., Liao, Q., Tian, Q., ... Liu, L. (2021). Pixel difference networks for efficient edge detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 5117–5127).
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).
- Tang, Q., Sang, N., & Liu, H. (2016). Contrast-dependent surround suppression models for contour detection. *Pattern Recognition*, 60, 51–61.
- Tang, Q., Sang, N., & Liu, H. (2019). Learning nonclassical receptive field modulation for contour detection. *IEEE Transactions on Image Processing*, 29, 1192–1203.
- Tu, Z., Ma, Y., Li, C., Tang, J., & Luo, B. (2020). Edge-guided non-local fully convolutional network for salient object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 31, 582–593.
- Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2016). Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*.
- 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).
- Wei, H., Lang, B., & Zuo, Q. (2013). Contour detection model with multi-scale integration based on non-classical receptive field. *Neurocomputing*, 103, 247–262.
- Wibisono, J. K., & Hang, H.-M. (2020a). Fined: Fast inference network for edge detection. *arXiv preprint arXiv:2012.08392*.
- Wibisono, J. K., & Hang, H.-M. (2020b). Traditional method inspired deep neural network for edge detection. In *2020 IEEE International Conference on Image Processing (ICIP)* (pp. 678–682): IEEE.
- Wu, Y., & He, K. (2018). Group normalization. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 3–19).
- Xiaofeng, R., & Bo, L. (2012). Discriminatively trained sparse code gradients for contour detection. In *Advances in neural information processing systems* (p. 25).
- Xie, S., & Tu, Z. (2017). Holistically-nested edge detection. *International Journal of Computer Vision*, 125, 3–18.

- Xuan, W., Huang, S., Liu, J., & Du, B. (2022). FCL-Net: Towards accurate edge detection via Fine-scale Corrective Learning. *Neural Networks*, *145*, 248–259.
- Yang, J., Price, B., Cohen, S., Lee, H., & Yang, M.-H. (2016). Object contour detection with a fully convolutional encoder-decoder network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 193–202).
- 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 Transactions on Image Processing*, *24*, 2565–2578.
- Yang, K.-F., Li, C.-Y., & Li, Y.-J. (2014). Multifeature-based surround inhibition improves contour detection in natural images. *IEEE Transactions on Image Processing*, *23*, 5020–5032.
- Yang, K., Gao, S., Li, C., & Li, Y. (2013). Efficient color boundary detection with color-opponent mechanisms. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2810–2817).
- Zeng, C., Li, Y., & Li, C. (2011). Center-surround interaction with adaptive inhibition: a computational model for contour detection. *NeuroImage*, *55*, 49–66.
- Zhang, Q., Lin, C., & Li, F. (2021). Application of binocular disparity and receptive field dynamics: a biologically-inspired model for contour detection. *Pattern Recognition*, *110*, Article 107657.
- Zhong, H., & Wang, R. (2021). A new discovery on visual information dynamic changes from V1 to V2: corner encoding. *Nonlinear Dynamics*, *105*, 3551–3570.
- Zhou, C., Huang, Y., Pu, M., Guan, Q., Huang, L., & Ling, H. (2023). The treasure beneath multiple annotations: an uncertainty-aware edge detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 15507–15517).