Complex & Intelligent Systems (2024) 10:2703–2715
https://doi.org/10.1007/s40747-023-01292-0

ORIGINAL ARTICLE

Blsnet: a tri-branch lightweight network for gesture segmentation against cluttered backgrounds

Guoyu Zhou[1,2] · Zhenchao Cui[1,2] · Jing Qi[1,2]

Received: 21 May 2023 / Accepted: 11 November 2023 / Published online: 12 December 2023

Abstract

Hand gesture segmentation is an essential step to recognize hand gestures for human–robot interaction. However, complex backgrounds and the variety of gesture shapes cause low semantic segmentation accuracy in the existing lightweight methods because of imprecise features and imbalance between branches. To remedy the above problems, we propose a new segmentation structure for hand gestures. Based on the structure, a novel tri-branch lightweight segmentation network (BLSNet), is proposed for gesture segmentation. Corresponding to the structure parts, three branches are employed to achieve local features, boundaries and semantic hand features. In the boundary branch, to extract multiscale features of hand gesture contours, a novel multi-scale depth-wise strip convolution (MDSC) module is proposed based on gesture boundaries for directionality. For hand boundary details, we propose a new boundary weight (BW) module based on boundary attention. To identify hand location, a semantic branch with continuous downsampling is used to address complex backgrounds. We use the Ghost bottleneck as the building block for the entire BLSNet network. To verify the effectiveness of the proposed network, corresponding experiments have been conducted based on OUHANDS and HGR1 datasets, and the experimental results demonstrate that the proposed method is superior to contrast methods.

Keywords Gesture segmentation · Deep learning · Lightweight network · Feature extraction

Introduction

Gesture interaction is an intuitive and natural communication method and can provide effective simple, intuitive, and concise human–machine interaction [1, 2]. Thus, hand gesture interaction [3, 4] achieves the attention of many researchers. Generally, gesture interaction can be divided into wearable device-based methods [5] and machine vision-based methods [6–8]. For convenience, machine vision-based methods have become mainstream.

Machine vision-based gesture interaction usually involves three steps: gesture segmentation, feature extraction, and gesture recognition [9]. Since removing background in gesture images, gesture segmentation is the prerequisite of the entire gesture interaction system, and its result improves subsequent feature extraction and recognition accuracy. However, in practical applications, it is a challenging task because of the small difference between foreground and background, uneven lighting conditions, and various shapes of gestures. To minimize the interference of background noise and obtain a complete hand posture for subsequent gesture recognition, we focus on researching gesture segmentation.

For gesture segmentation, machine learning-based and deep learning-based methods were proposed. In practical application, methods based on conventional machine learning methods cannot remedy the problems in gesture segmentation. These methods [10, 11], always explore single and predefined operators to obtain the features of gestures, such as skin color [12, 13], gradient direction histogram [14], Haar features [15], scale-invariant feature transform [16], and motion of hand [17]. However, gesture images with complex backgrounds and various shapes of hand gestures are hardly segmented or detected using the single feature [9]. To fuse multiple features of hand gestures, several ensemble methods [18, 19] were proposed for gesture segmentation. However, the weights of weak classifiers seriously affect the results of segmentation and detection for hand gestures. The training

✉ Zhenchao Cui
cuizhenchao@gmail.com

[1] School of Cyber Security and Computer, Hebei University, Baoding, China

[2] Hebei Machine Vision Engineering Research Center, Hebei University, Baoding, China

efficiency is one of the problems in ensemble methods for hand gesture detection.

With the development and application of deep learning in many areas, deep learning-based methods have become mainstream hand gesture segmentation methods [20, 21]. Several networks based on deep learning were proposed for hand segmentation. For instance, Al-Hammadi et al. [22] utilized multiple deep learning architectures to segment hand regions. Dadashzadeh et al. [9] increased gesture segmentation precision by leveraging residual network structures and dilated spatial pyramid pooling. Wang et al. [23] used the MSF module and LWMS module to enhance the network's multiscale feature extraction capability but ignored the size of gesture segmentation network parameters.

Generally, hand gesture interaction is always employed in edge devices, thus, the light and real-time networks are important for this application. To obtain lightweight networks, several methods [24] were proposed. Dang et al. [25] proposed gesture recognition methods based on DeeplabV3+ and U-net, which can reduce the overall parameter volume by replacing the backbone network. However, because of the imprecise feature representation for hand, the overall segmentation accuracy is still low. To improve segmentation accuracy, Dayananda [26] proposed a new hybrid approach based on RGB-D gesture images. However, compared with RGB images, it is limited to the image dataset. Similar to U-net, Das et al. [27] used an encoder–decoder architecture for real-time pixel-level semantic segmentation. ICNet [28] uses image cascading to accelerate the algorithm, while DFANet [29] uses an architecture based on depth separable convolution to build a lightweight backbone. To remedy the problems of computation for multi-scale and high-resolution, ESPNet [30, 31] was proposed based on spatial pyramid pooling modules. Despite being able to complete inference tasks instantly, these methods only adopt a single feature processing approach. Due to the lack of consideration for underlying details, the accuracy of models is severely reduced.

To fuse multi-feature for hand gesture segmentation, several methods with dual-branch models were proposed. In BiSeNet [32], the dual-branch network model is used for detail analysis and context analysis. To strengthen features from the context branch and detail branch, DDRNet [33] adopts a bridging method to a dual-branch structure. However, these dual-branch methods ignore the diversity of features and the gap between the context branch and detail branch, which would lose many details at a lower resolution, and lead to un-accurate segmentation results for gesture.

Gesture segmentation is a pixel-dense prediction task that relies heavily on the local detail and texture of gestures. However, traditional encoder–decoder networks and dual-branch networks ignore the connection with gesture details, shape and context information. To refine gesture features, we propose a new structure that divides the extracted gesture features into three types: boundary features, local features, and semantic features. By categorizing gesture features into different types, the proposed network can obtain more comprehensive and diverse features. It can flexibly adjust and utilize these features to improve the accuracy and robustness of gesture prediction. In addition, dividing the features can reduce the difficulty of network feature extraction. Based on this, we propose BLSNet for gesture segmentation, which includes three branches: local feature branch, semantic feature branch and boundary branch. The local feature branch has the characteristics of wide channels and shallow levels, mainly used for extracting detailed gesture features. The semantic feature branch involves narrow channels and deep hierarchical structures, learning high-level semantic context through a considerable degree of downsampling. And the boundary branch focuses on extracting boundary information of gestures. To construct a lightweight network architecture, the Ghost bottleneck is used in our method as the backbone.

The main contributions are summarized as follows.

1. A multibranch segmentation structure is proposed in this paper, proving that segmentation can be obtained by multicategory features from deep neural networks.
2. Based on this structure, a tri-branch lightweight network named BLSNet is proposed for gesture segmentation. BLSNet contains three branches for boundary, local and semantic feature extraction to extract three different types of gesture features.
3. To refine gesture features, we propose the BW module and MDSC module for gesture boundary and texture features and use the ASPP module for semantic features. To fuse the three channels, the Bag module is employed to promote network optimization.

The article is organized as follows. "BLSNet" section provides a detailed introduction to the proposed BLSNet. To demonstrate the effectiveness of the proposed BLSNet, corresponding experiments are conducted, and the results are presented in "Experiments" section. The conclusion is drawn in "Conclusion" section.

BLSNet

In this section, we propose a lightweight network, named BLSNet, based on three branches for hand segmentation. The three branches are designed to obtain three different types of gesture features.

Overview

Let $x \in X$, x is a given pixel in hand gesture images, and X is the set of pixels. The predicted label of x is presented as

l'_x, and $l'_x \in \{0, 1\}$. l'_x =1 predicts that x is the foreground of the image, while l'_x =0 presents that x is the background of the image. The probability structure for hand segmentation can be presented as Eq. 1.

$$P\left(l'_x \mid x\right) = \sum_{i=1}^{n} P\left(l'_x \mid x, \theta_i\right) P(\theta_i), \tag{1}$$

where θ_i is the parameter of the i-th branch in the segmentation model and n is the number of branches. One of the key conditions for Eq. 1 is that the features obtained using parameters θ_i are independent. However, it is difficult to determine the parameters. To obtain the predicted label of x, we introduce the slack parameters $\widetilde{\theta_i}$, which are considered with the parameters of branches θ_i. We can obtain the predicted structure $\sum_{i=1}^{m} P(l''_x \mid x, \widetilde{\theta_i}) P(\widetilde{\theta_i})$. l''_x is the label of x detected by the parameters of $\widetilde{\theta_i}$. Because of the independence of features, we can have Eq. 2.

$$P\left(l'_x \mid x\right) \leq \sum_{i=1}^{m} P\left(l''_x \mid x, \widetilde{\theta_i}\right) P\left(\widetilde{\theta_i}\right) \tag{2}$$

We can obtain the same label using the two probability structures of $\sum_{i=1}^{n} P(l'_x \mid x, \theta_i) P(\theta_i)$ and $\sum_{i=1}^{m} P(l''_x \mid x, \widetilde{\theta_i}) P(\widetilde{\theta_i})$. Since we can obtain global minima using deep neural networks [34], the label from our slack parameter $\widetilde{\theta_i}$ is similar to the true label of the predicted pixel, as shown in Eq. 3.

$$l = \underset{l'_x}{\operatorname{argmax}} P\left(l'_x \mid x\right) \approx \underset{l''_x}{\operatorname{argmax}} \sum_{i=1}^{m} P\left(l''_x \mid x, \widetilde{\theta_i}\right) P\left(\widetilde{\theta_i}\right), \tag{3}$$

where l is the label predicted using our structure. Since of the binary of label set and designed networks, the final structure is represented in Eq. 4.

$$l = \underset{l''_x}{\operatorname{argmax}} \sum_{i=1}^{m} P\left(l''_x \mid x, \widetilde{\theta_i}\right) \tag{4}$$

Based on our structure, to achieve independent features for gesture segmentation, we propose a tri-branch network based on local feature branch, boundary branch and semantic branch in this paper. The network is named BLSNet. An overview of the proposed method is shown in Fig. 1. Specifically, after simple feature extraction through a convolutional layer and 2 Ghost bottlenecks [35], the input image's resolution is reduced to 1/4 of its original size. Then, it is fed into three branches for downsampling at different levels, which are the local feature branch, boundary branch, and semantic branch. In the three branches, different shapes of cubes represent the height, width, and number of channels of feature maps. In addition, the number in the top right corner of the cube indicates the relative size between the current feature map and the original input image. The color of the cube is mainly used to distinguish three branches. Finally, the outputs of the three branches are fused using the Bag module. In simple terms, the Bag (Boundary-attention-guided fusion) module uses boundary feature maps to obtain a gesture weight score map and then fuses it with local feature maps and semantic feature maps based on this score. The details will be explained in "Feature fusion" section. To construct a lightweight network, we use the Ghost bottleneck [35] as the building block, which can greatly reduce the computation and parameters compared with other mainstream backbones.

The boundary branch

We set up a boundary branch and use it as the main branch to coordinate the feature extraction work of the local feature branch and the semantic branch.

In addition, gestures come in various shapes, sizes, and directions, and are easily influenced by cluttered backgrounds. Using a multi-scale feature extraction method to learn gesture features is more in line with the characteristics of gesture segmentation tasks. Therefore, we design a Multi-scale Depth-wise Strip Convolution (MDSC) module, as shown in Fig. 2. The module has five depth-wise strip convolution branches and one 1 × 1 convolution branch, and the feature learning results of the six branches are concatenated to obtain the output feature map of the module. The module can be represented as Eq. 5.

$$\begin{aligned} Output_{MDSC} &= Conv_{1\times1}(F) \\ &\quad + \sum_{i=0}^{4} Scale_i\left(DW\_Conv(F)\right), \end{aligned} \tag{5}$$

where F represents the input feature map, $Conv_{1\times1}$ represents 1 × 1 convolution, DW_Conv represents deep convolution, $Scale_i$, and $i \in \{0, 1, 2, 3, 4\}$ represents the ith branch in the Fig. 2. Each branch contains two deep strip convolutions, with respective convolution kernel sizes of $1 \times n$ and $n \times 1$, which are used to simulate the standard 2D convolution with a kernel size of $n \times n$. Here, n takes the values of 3, 7, 11, 15, and 21.

The pseudocode for MDSC is shown below.

The reasons for choosing deep stripe convolution to extract gesture boundary features are as follows. First, deep strip convolution is a lightweight convolution method. Compared with the model that does not use the MDSC, the GFLOPs of the network model increases by only 0.26%, and related discussions are conducted in the ablation exper-

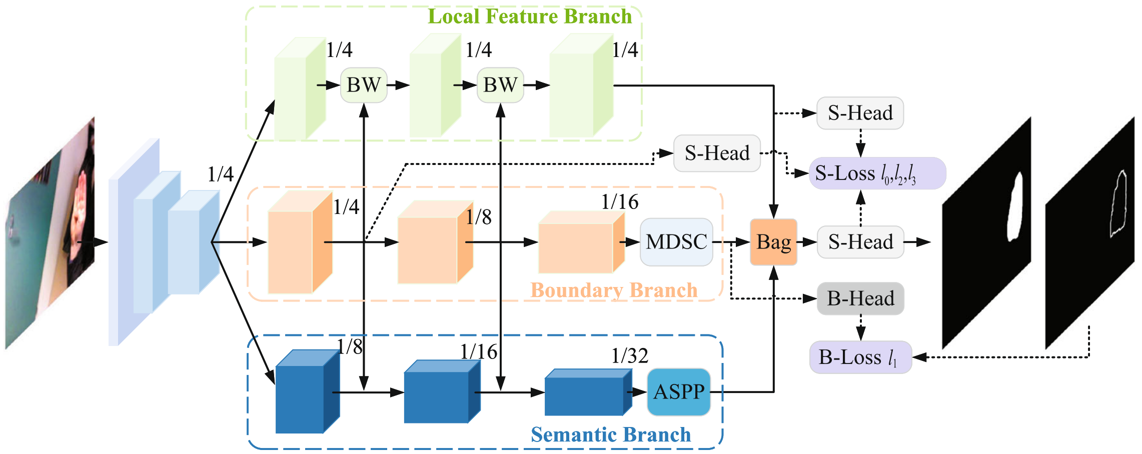

Fig. 1 An overview of the proposed network (BLSNet)

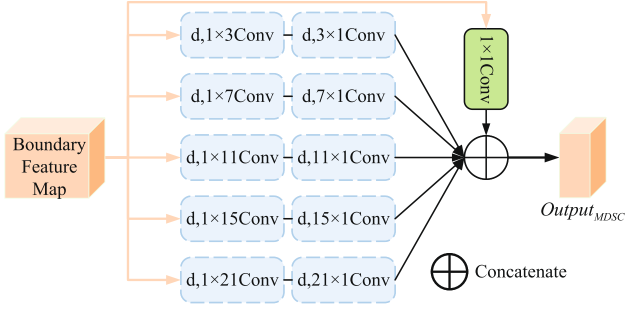

Fig. 2 Multi-scale depth-wise strip convolution (MDSC) Module

Algorithm 1 The MDSC module algorithm

```
Input: boundary feature map $F$
for $i = \{1, 2, 3, 4, 5, 6\}$ do
    if $i = 1$ then
        $Output = Conv_{1\times1}(F)$
        $Output_{MDSC} += Output$
    else
        for $j = \{3, 7, 11, 15, 21\}$ do
            $Output = DW\_Conv_{1\times j}(F) + DW\_Conv_{j\times1}(F)$
            $Output_{MDSC} += Output$
        end for
    end if
end for
Output: Updated boundary feature map $Output_{MDSC}$
```

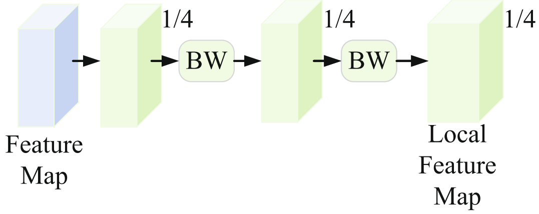

Fig. 3 The illustration of the proposed local branch

iment section. Second, the gesture boundary shape can be a linear, nonlinear, or complex curve. Using strip convolution can accurately locate the gesture boundary and more efficiently extract such band-shaped features. Finally, by changing the size of the stripe convolution kernel, the network model can adapt to different boundary shapes and scales. Since strip convolution performs convolution in only one direction, compared with standard 2D convolution, strip convolution can extract longer sequence information at the same computational complexity, which is crucial for modeling long-distance dependence relationships and improving boundary feature extraction robustness and accuracy.

The local feature branch

One local feature branch is designed to maintain the feature map resolution and reduce the loss of refined features, as shown in Fig. 3. Although this branch feature map has a higher resolution, it has fewer channels, which to some extent ensures the lightweight of the model. In addition, the introduction of the detail branch ensures the sensitivity of the gesture segmentation network to subtle features and provides better detail discrimination ability. By working in synergy with other branches such as the semantic branch and boundary branch, more accurate and detailed gesture segmentation can be achieved.

The semantic branch

To extract contextual semantic information from gestures, we adopt a continuous downsampling strategy in the semantic branch and cooperate with the ASPP [36] context module to rapidly expand the model's receptive field in the final stage and extract high-level semantic features of the gestures. As shown in Fig. 4, the semantic branch has a narrow channel and deep characteristics, and the deeper channel enables it to

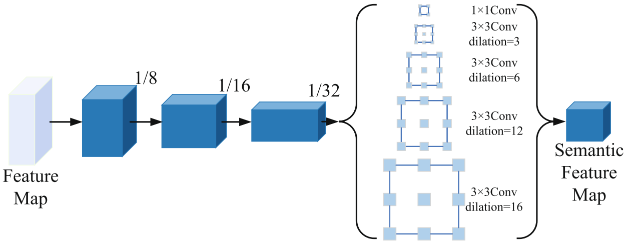

Fig. 4 The illustration of the proposed semantic branch

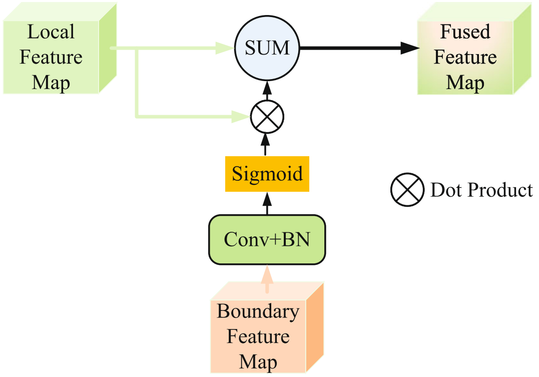

Fig. 5 Boundary weight (BW) module

store sufficient contextual information. Although the number of channels in this branch is relatively large, the resolution size limits parameter growth, conforming to our idea of simplifying a unified segmentation task and ultimately designing a lightweight network.

Feature fusion

This section primarily introduces the fusion between the boundary branch and the local branch, the fusion between the boundary branch and the semantic branch, as well as the final fusion process of the three branches.

The fusion between the boundary branch and the local branch

Since the local feature branch and boundary branch have different feature learning tasks under different supervision, certain information differences and complementarities exist between them. The local feature branch focuses more on the local detail features of the feature map, such as texture and shape; the boundary branch focuses more on the decision boundary between the gesture boundary and background. The local feature branch can selectively learn gesture boundary features, thereby optimizing the focus on each local detail to achieve better segmentation results.

Therefore, we propose a new boundary weight (BW) module, as shown in Fig. 5. First, the feature vector of the boundary branch is subject to 1×1 convolution and batch normalization. Then, the sigmoid is calculated to obtain the attention-mapping image. The vector of the pixel position (x, y) in the feature map provided by the boundary feature branch is defined as $\vec{v}_b\ (x, y)$, where $0 \leq x < H$, $0 \leq y < W$, H and W are the length and width of the boundary feature maps, respectively. The output of the sigmoid function can be represented as Eq. 6.

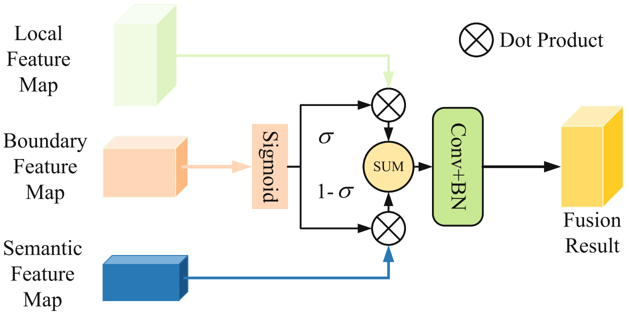

Fig. 6 Boundary-attention-guided fusion (Bag) Module

$$\sigma = Sigmoid\left(f\left(\vec{v}_b\right)\right) \tag{6}$$

We multiply the vector $\vec{v}_d$ at the pixel position (x, y) in the local feature map by the corresponding pixel position σ in the boundary feature map weight matrix after passing through the sigmoid function. This process focuses more on gesture boundaries in the local feature map. Then, we add the local feature map with gesture boundary attention to the original local feature map to obtain the output of the BW module. The purpose of this addition is to prevent excessive boundary weights from causing damage to details in the original feature map. This process can be written as Eq. 7:

$$Output_{BW} = \sigma\ \vec{v}_d + \vec{v}_d\ . \tag{7}$$

The fusion between the boundary branch and semantic branch

We adopt a direct downsampling and addition method to merge boundary features into semantic features, thereby enhancing the semantic branch's recognition capability for gestures and backgrounds.

The final fusion process of the three branches

Since of the differences in feature representations from the local feature branch, boundary branch, and semantic branch for gesture features, it would lead to unbundling features by directly fusing features from three branches. To balance the weights of the three branches, as shown in Fig. 6, a

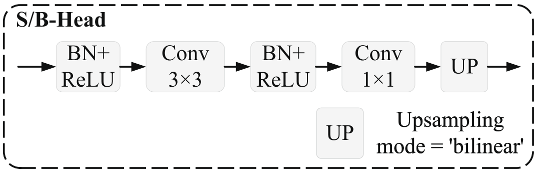

Fig. 7 The construction of the S/B-Head

Boundary-attention-guided fusion (Bag) module [37] is utilized to coordinate the fusion of the feature result maps of the three branches.

Specifically, the outputs of the local feature branch, boundary branch, and semantic branch are defined as $\vec{v}_d$, $\vec{v}_b$, and $\vec{v}_s$, respectively. The output of the sigmoid and Bag can be expressed as Eqs. 8 and 9.

$$\sigma = Sigmoid\left(\vec{v}_b\right), \tag{8}$$

$$Output_{Bag} = f_{Bag}\left(\sigma\ \vec{v}_d + (1-\sigma)\ \vec{v}_s\right) \tag{9}$$

As shown in Fig. 6, when σ is greater than 0.5, more detailed features will be obtained using the local feature and boundary feature; otherwise priority will be given to using the semantic feature with boundary information.

In addition, multiple additional loss functions are employed to help optimize the network model, as shown in Fig. 1. The S/B-Head are placed before the loss functions. Moreover, we provide the specific construction of the S-Head (semantic head) and B-Head in Fig. 7. They are both composed of convolutional layers, batch normalization layers, activation functions, and an upsampling layer. We utilize this structure to transform feature maps with a large number of channels into feature maps with specified channel numbers, adjust their resolution size, and then compare the transformed feature maps with label values for loss calculation.

The total loss function of the network is the sum of each loss function multiplied by their respective coefficients, and the calculation process is shown in Eq. 10.

$$F_{Loss} = \lambda_0 l_0 + \lambda_1 l_1 + \lambda_2 l_2 + \lambda_3 l_3, \tag{10}$$

where the specific positions of l_0, l_1, l_2 and l_3 are shown in Fig. 1. Specifically, l_0, l_2 and l_3 all use the cross-entropy loss function, which is widely applied in semantic segmentation tasks. The specific definition can be seen in Eq. 11.

$$l_j = -\sum_{i}^{H\times W} G^i log\left(P_j^i\right), \tag{11}$$

where $j = \{0, 2, 3\}$, G^i denotes the i-th pixel value on the real value, and P_j^i denotes the i-th pixel value on the predicted output images of the loss function l_j.

For edge supervision, following the MDSC, a boundary head(B-Head) is employed, and to optimize the weight, the Dice loss [38] function l_1, is used in this paper. Compared with conventional loss functions such as cross-entropy loss function, the Dice loss function can deal with the problem of imbalance between positive and negative samples. The l_1 calculation process is given by Eq. 12.

$$l_1 = 1 - \frac{2\cdot\sum_i^{H\times W}\left(P_1^i\cdot G_1^i\right)+\epsilon}{\sum_i^{H\times W}P_1^i+\sum_i^{H\times W}G_1^i+\epsilon} \tag{12}$$

where P_1^i denotes the i-th pixel value on the predicted edge output images, G_1^i denotes the i-th pixel value on the corresponding edge target images and ϵ is set as 1.

The parameters in F_{Loss} in this paper are empirically [33, 39] set to $\lambda_0 = 1$, $\lambda_1 = 0.5$, $\lambda_2 = 0.4$ and $\lambda_3 = 0.4$. In Fig. 1, the dashed lines and related blocks are ignored during inference.

Experiments

To test the BLSNet proposed in this paper, we compare it with several segmentation methods on the two datasets of OUHANDS and HGR1 with the criteria of PixAcc, mIOU, GFLOPs, and the model parameters.

Datasets, computation platform and evaluation criteria

The OUHANDS dataset [40] contains 10 different hand gestures from 23 subjects, of which 2000 were selected for the training set and the remaining 1000 for the test set. The photos in this dataset exhibit complex background and lighting variations, the shapes and sizes of the subjects' hands vary, and their skin color varies as well. Additionally, the images exhibit varying degrees of occlusion between hands and faces.

The HGR1 dataset [41] contains a total of 899 RGB images of 25 different hand gestures performed by 12 subjects. Among them, we selected 630 images as the training set and the remaining 269 images as the test set. The hand gesture images in this dataset are greatly influenced by background variations, but there is no occlusion between the hands and faces.

The model was trained and tested on a platform composed of a single NVIDIA RTX 3080, PyTorch 1.11, CUDA 11.3, cuDNN 8.0, and Anaconda. To optimize the training process of the OUHANDS dataset, the batch size was set to 4 and the input image resolution was fixed to 320–320. We use the Adam optimizer for weight optimization, initialize the learning rate to 0.005, and set the weight decay to 0.0001.

The total number of training epochs is 300. For the HGR1 dataset, the batch size was set to 16, the learning rate was initialized to 0.001, the weight decay was set to 0.0001, and all other settings were kept the same as for the OUHANDS dataset.

Mean intersection over union (mIOU) and pixel accuracy (PixAcc) are used as evaluation metrics for BLSNet. PixAcc is used to measure the pixel classification accuracy. It calculates the ratio of correctly classified pixels to total pixels. mIOU represents the mean value of pixel IOU (intersection over union). This metric provides a comprehensive evaluation of segmentation algorithms' performance across various categories and reflects their overall effectiveness. mIOU is defined by Eq. 13.

$$mIOU = \frac{1}{C}\sum_{i=1}^{C}\frac{TP_i}{TP_i + FP_i + FN_i}, \tag{13}$$

where TP_i, FP_i and FN_i represent the number of pixels predicted as class i and correctly classified, the number of pixels predicted as class i but misclassified, and the number of pixels that should belong to class but are incorrectly classified as other categories, respectively. C is the number of categories.

Overall, PixAcc is a simple and intuitive evaluation metric, while mIOU focuses more on the matching degree between predicted results and ground truth. If the model has made progress in improving the results of target boundary segmentation, the corresponding mIOU will increase accordingly, while pixel accuracy may only show a slight increase. Thus, mIOU better reflects segmentation performance.

GFLOPs are commonly used to evaluate the computational complexity of convolutional neural networks. Parameters refer to the number of parameters in a neural network model, including weights and biases.

Ablation experiments

A series of ablation experiments were conducted on the OUHANDS dataset, including the loss function, BW module, MDSC module, and Bag module.

Effectiveness of extra losses

To investigate the impact of additional training supervision on network performance, we conduct ablation experiments by combining l_1, l_2, and l_3. The results are shown in Table 1. We find that without adding extra semantic supervision, the model's mIOU is only 86.28%. After adding each loss function separately, the model accuracy improved, with the most significant improvement (+ 2.53% mIOU) observed when adding l_1. This provides strong evidence for the importance of boundary loss functions and boundary branches. In the case of adding only two loss functions, the model performance further improves. It is worth noting that the experiments of adding l_1, l_3 and adding l_2, l_3 yielded the same PixAcc but different mIOU. This is normal because their calculation processes are different. mIOU measures the average overlap between predicted segmentation results and true labels, while PixAcc measures the model's ability to correctly classify pixels. mIOU is more accurate in evaluating image segmentation performance compared to PixAcc because it considers interclass relationships and spatially overlapping areas. When all three auxiliary loss functions are added, the model achieves its highest mIOU at 89.42%.

Table 1 Ablation study of extra losses

l_1	l_2	l_3	PixAcc (%)	mIOU (%)
			96.85	86.28
✓			97.52	88.81
	✓		97.16	87.52
		✓	97.40	88.39
✓	✓		97.62	89.36
✓		✓	97.58	89.05
	✓	✓	97.58	89.15
✓	✓	✓	**97.65**	**89.42**

Bold values represent the maximum for PixAcc and mIOU

The effectiveness of BW, MDSC, and Bag, and the threefold cross-validation

The BW, MDSC, and Bag modules were combined in different ways to explore the impact of each module on the overall network performance. As shown in Table 2, we find that when none of the three modules participate in training, the overall model accuracy is the lowest (97.16%), and mIOU is only 87.61%. After adding the MDSC or Bag module, the model's mIOU increased by 0.59% or 0.75%, respectively. When both MDSC and Bag modules are added, the model's mIOU reaches 89.16%, a relative improvement of 1.55% compared to the lowest accuracy. When only the BW module is included in the model, the mIOU improves by 0.28%, but the effect of the BW module is not as significant as that of the MDSC and Bag modules. The combination of MDSC and Bag modules with the BW module improves network performance, and the overall performance is highest when all three are added to the network, with PixAcc at 97.65% and mIOU at 89.42%.

From the results, our network model has a high PixAcc without the BW, MDSC, and Bag modules. However, for some data in the dataset, due to the lack of fine-grained operations on features by these three modules, the model's segmentation results may have inaccurate segmentation or

Table 2 BW, MDSC and Bag's ablation study and threefold cross-validation

BW	MDSC	Bag	PixAcc (%)	mIOU (%)	GFLOPs	Parameters(M)
			97.16	87.61	**3.41**	**2.83**
		✓	97.32	88.20	4.36	2.98
	✓		97.37	88.36	3.67	3.48
✓			97.28	87.89	3.52	2.86
	✓	✓	97.61	89.16	4.62	3.63
✓		✓	97.39	88.35	4.47	3.01
✓	✓		97.42	88.49	3.78	3.51
✓	✓	✓	**97.65**	**89.42**	4.73	3.66
Threefold cross-validation			98.11	91.24	4.73	3.66

Bold values represent the maximum for PixAcc and mIOU(excluding the results of Threefold cross-validation), or the minimum for GFLOPs and Parameters

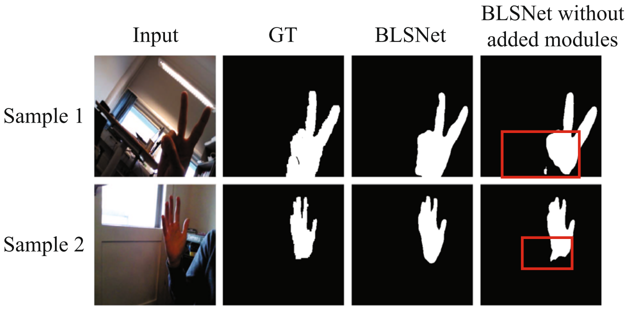

Fig. 8 Comparison of visual results between BLSNet and BLSNet without added modules

larger artifacts at gesture edges as shown in samples 1 and 2 in Fig. 8. This leads to a higher PixAcc but poorer results. Therefore, on the overall dataset, incorporating these three modules does not show significant improvement in PixAcc, but it has more noticeable effects on result visualization and mIOU.

In the threefold cross-validation, we randomly and uniformly partitioned the dataset. The model performed well, with an average PixAcc of 98.11% and an average mIOU of 91.24%.

In addition, we evaluated the GFLOPs and parameters of the model after adding each module. The addition of the BW module, MDSC module, and Bag module increased the GFLOPs of the model by 0.11, 0.26, and 0.95, respectively, while the parameters increased by 0.03 M, 0.65 M, and 0.15 M, respectively. Although the MDSC module has more parameters, it only slightly increases the computational complexity of the model. Moreover, it significantly improves the mIOU of the model (+ 0.75). This shows that with the help of the MDSC module, gesture features extracted by the boundary branch are more effective, also meeting the needs of lightweight network models. When these three modules are added pairwise to the model separately, there is an improvement in both GFLOPs and parameters. However, when all three modules are added to the model, it achieves optimal performance with only a slight increase of 1.32 GFLOPs and 0.83 M parameters compared to not adding any modules at all.

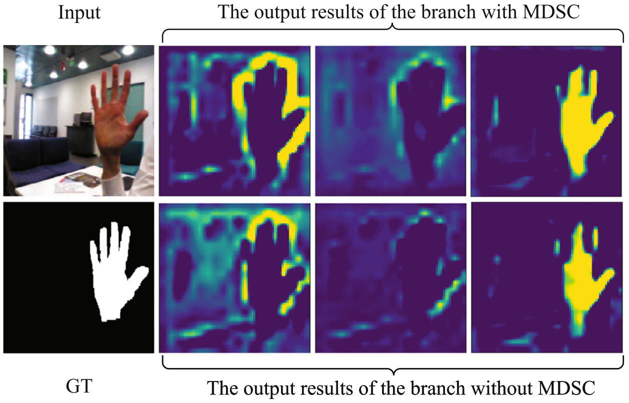
Input
The output results of the branch with MDSC
GT
The output results of the branch without MDSC

Fig. 9 Feature visualization of the MDSC module

We visualize the features of the MDSC module and Bag module. Figure 9 shows the feature visualization of the MDSC module. The top row from left to right shows the original input image and the visualization results of three random channels in the output feature map of the MDSC module. The bottom row from left to right shows the ground truth and the random three-channel output results of branches without the MDSC module. Obviously, after adding the MDSC module, the boundary features of the hand are more prominent, and the contour is more complete. The output of the boundary branch without this module does not have a clear division in the gesture boundary area.

Figure 10 shows the feature visualization of the Bag module, with the same layout as Fig. 9. The rightmost three images in the top row are the Bag module output, and the rightmost three images in the bottom row are the output feature maps of some branches without the Bag module. When not using the Bag module, we directly add the three branches together. From Fig. 10, it can be seen that under the guidance of the Bag module, the network model can clearly determine the hand position and gesture boundary, while the result

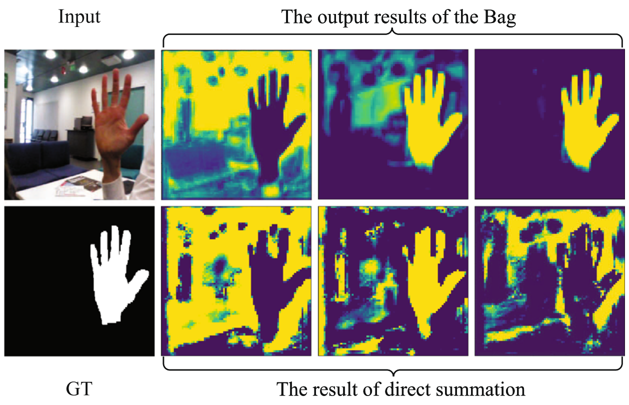

Fig. 10 Feature visualization of the Bag module

of directly adding the feature maps of the three branches together does not have smooth and complete gesture edges, and the overall judgment of the foreground and background is also fuzzy and confused. Obviously, boundary features in the Bag module can guide the integration of local detail features and semantic features correctly.

Comparison with other methods

To test our model, we propose experimental comparisons between BLSNet and HGRNet, DDRNet series, SegFormer, and PP-LiteSeg series in terms of PixAcc, mIOU, GFLOPs and parameter evaluation metrics.

Table 3 shows the statistical results of several segmentation methods on OUHANDS data with the criteria of PixAcc, mIOU, GFLOPs and parameters. From Table 3, we can see that our model achieves the highest accuracy compared to other segmentation models. Specifically, despite our model having a parameter size of 3.66 M, which is second only to HGRNet's 0.28 M, its mIOU is 12.21% higher than HGRNet. DDRNet-23-slim is the lightest and fastest model in the DDRNet series. Despite having the lowest GFLOPs at 1.85, DDRNet-23-slim's mIOU is only 80.14%. Meanwhile, DDRNet-39, having a parameter size of 32.65 M, has an mIOU of only 80.02%. SegFormer-0 is a lightweight model in the SegFormer series that uses self-attention mechanisms to coordinate contextual information. Its model parameter size is 3.71 M, but its mIOU is only 80.77%, which is much lower than our model's mIOU. PP-LiteSeg has been used frequently as a benchmark for lightweight semantic segmentation models, with the highest mIOU reaching only 86.23%, which is 3.19% lower than our model, and the difference in parameter size is significant. Thus, our model achieves a good balance between computational complexity and accuracy while having a lower number of parameters, indicating that our method can meet the edge device interaction requirements, such as human–robot interactions.

Figure 11 presents several segmentations using comparison methods on typical images. We can see that our model can still accurately distinguish the hand subject position even under significant changes in background lighting, while other models fail. In Sample 2, the hand position is clearly overexposed, and other models mistakenly classify other background positions as hands, but our model can still accurately segment gestures. We speculate that this is largely due to accurate boundary information extraction and guidance on the local feature branch and semantical branch. Contextual information is used to fill object areas outside the boundaries, and detailed features complete gesture edges.

In this section, HGR1 is used to test the proposed method. Table 4 shows the segmentation results on the HGR1 dataset. The BLSNet achieves the highest accuracy with an mIOU of 96.83%. The mIOU of the proposed method is 0.81% higher than that of PP-LiteSeg-B, obtaining the second-highest accuracy among the comparison methods. Furthermore, despite having a higher computational complexity of 4.73 GFLOPs compared to other methods, such as DDRNet-23-slim, SegFormer-b0, and HGRNet-seg, our method far surpasses them in terms of accuracy. The parameters are an indicator of model size, and our model has a parameter count of 3.66 M, which is only slightly higher than HGRNet-seg's 0.28 M. The segmentation results for several typical images in HGR1 are presented in Fig. 12, from which we can see that BLSNet performs significantly better than

Table 3 Performances of different approaches on the OUHANDS dataset

Method	PixAcc (%)	mIOU (%)	GFLOPs	Parameters (M)
HGRNet-seg [9]	94.64	77.21	3.07	**0.28**
DDRNet-23-slim [33]	95.25	80.14	**1.85**	5.73
DDRNet-23 [33]	95.32	80.95	7.26	20.29
DDRNet-39 [33]	95.10	80.02	14.25	32.65
SegFormer-b0 [42]	95.55	80.77	2.64	3.71
PP-LiteSeg-T [43]	96.60	85.66	7.15	13.52
PP-LiteSeg-B [43]	96.83	86.23	12.32	21.58
BLSNet	**97.65**	**89.42**	4.73	3.66

Bold values represent the maximum for PixAcc and mIOU, or the minimum for GFLOPs and Parameters

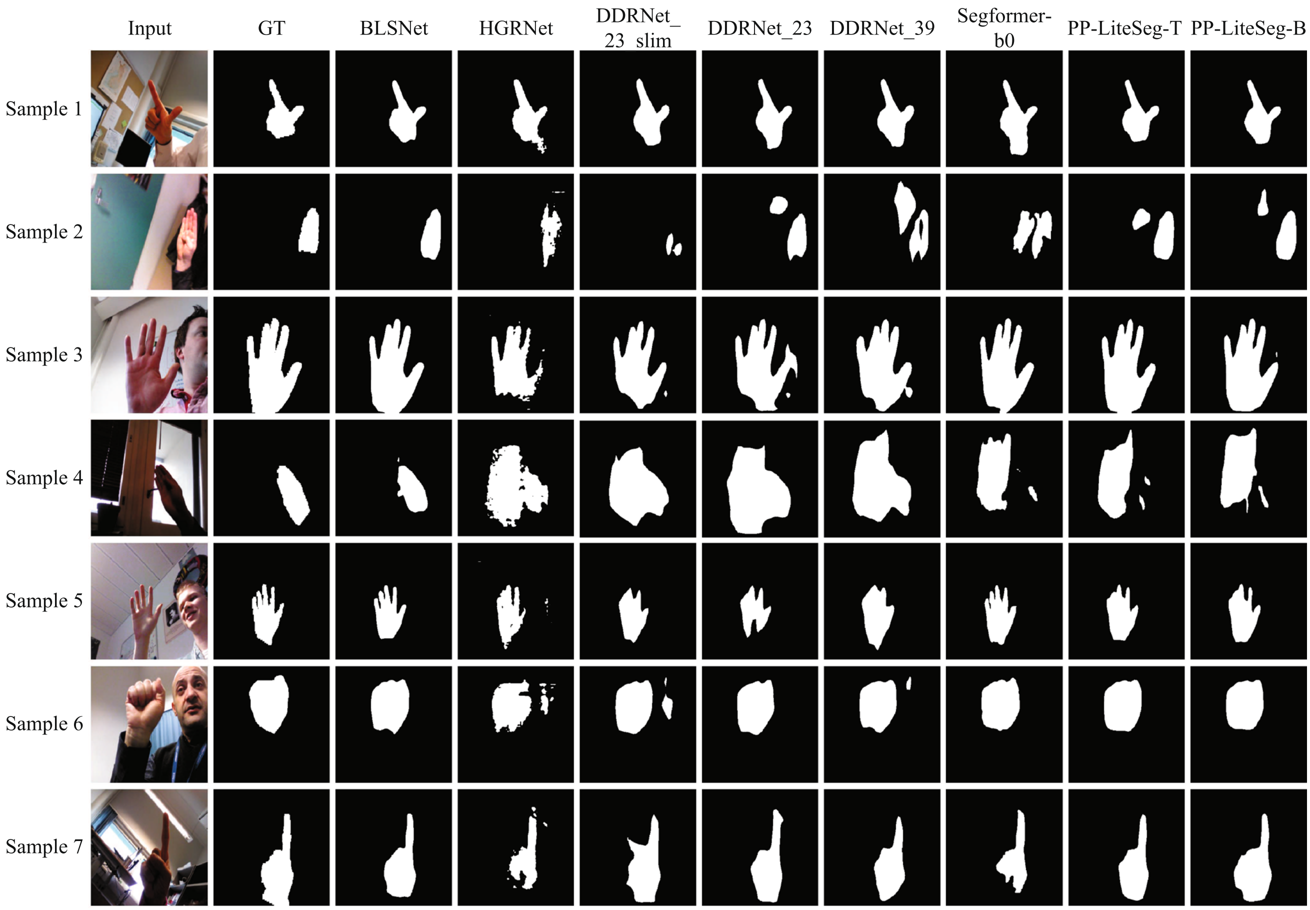

Fig. 11 An illustration of the segmentation performance of different methods on the OUHANDS dataset

Table 4 Performances of different approaches on the HGR1 dataset

Method	PixAcc (%)	mIOU (%)	GFLOPs	Parameters (M)
HGRNet-seg [9]	97.55	92.97	3.07	**0.28**
DDRNet-23-slim [33]	97.88	93.95	**1.85**	5.73
DDRNet-23 [33]	97.96	94.14	7.26	20.29
DDRNet-39 [33]	98.25	94.98	14.25	32.65
SegFormer-b0 [42]	98.57	95.92	2.64	3.71
PP-LiteSeg-T [43]	98.59	95.94	7.15	13.52
PP-LiteSeg-B [43]	98.62	96.02	12.32	21.58
BLSNet	**98.90**	**96.83**	4.73	3.66

Bold values represent the maximum for PixAcc and mIOU, or the minimum for GFLOPs and Parameters

other methods in restoring finger details, obtaining overall contours and removing the background.

Conclusion

Gesture segmentation in cluttered backgrounds poses a significant challenge, and traditional encoder–decoder networks are susceptible to information loss through repeated downsampling. The dual-branch architecture is inadequate in fusing detailed and contextual information about the gestures. Thus, based on the Bayesian framework, we propose BLSNet to segment the hand gesture images, comprising three branches devoted to extracting advanced boundaries, local information, and semantic information of the gestures. Extra semantic supervision is used to direct each branch's task. By establishing bridges between branches and activating a BW module between them, the feature representation and learning ability of each branch can be enhanced. We further exploit the MDSC module to improve the feature extraction ability of the boundary branch. Finally, the Bag module blends the semantic and local characteristics

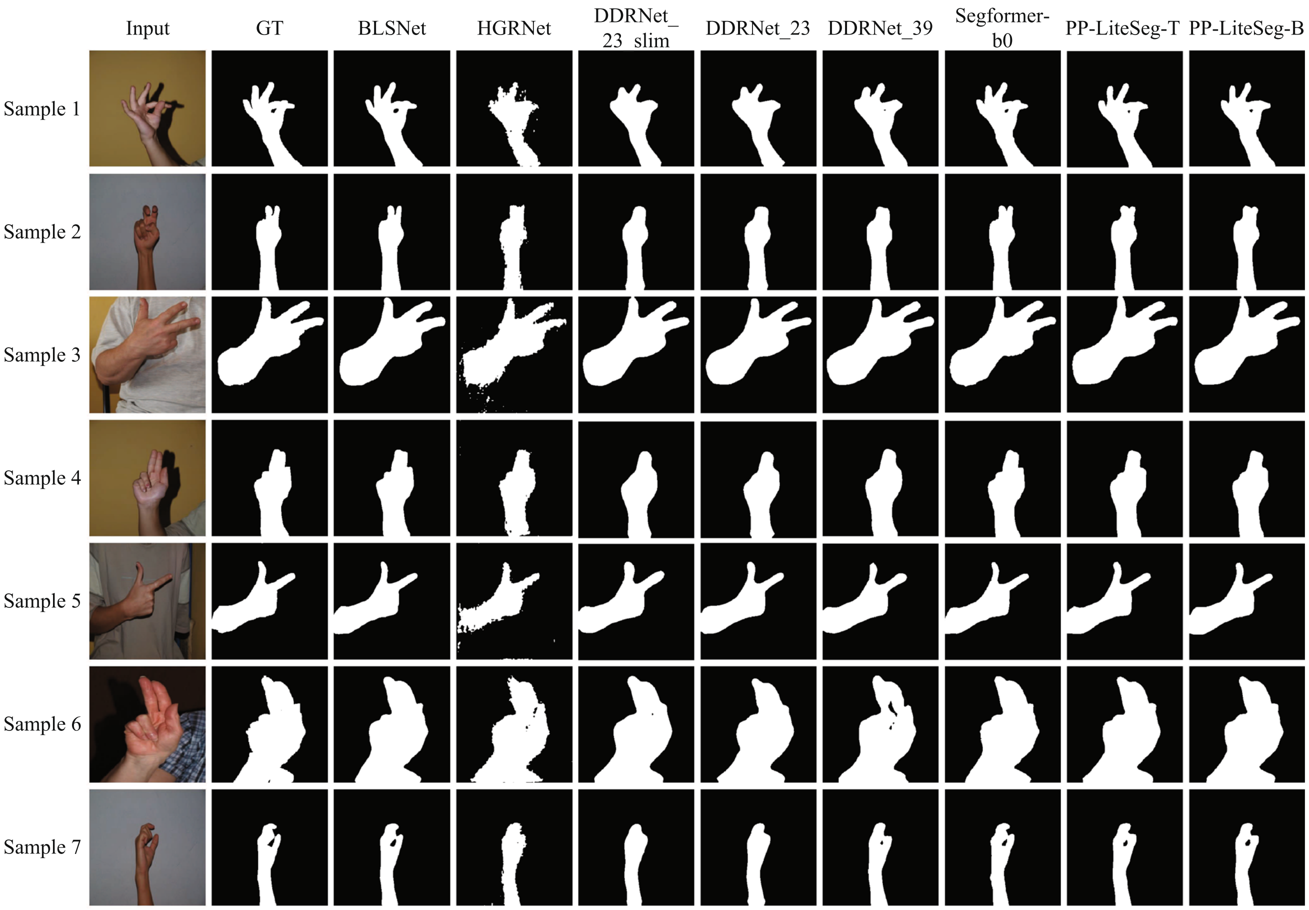

Fig. 12 An illustration of the segmentation performance of different methods on the HGR1 dataset

governed by the boundary information, yielding accurate gesture segmentation results. Experiments demonstrate that our network delivers higher accuracy than other lightweight networks while striking an optimal balance between computational complexity and accuracy.

Acknowledgements This work was supported in part by Scientific Research Foundation of Hebei University for Distinguished Young Scholars (Grant No. 521100221081), in part by Scientic Research Foundation of Colleges and Universities in Hebei Province (Grant No. QN2022107), in part by Science and Technology Program of Hebei Province (Grant No. 22370301D) and Central Government Guides Local Science and Technology Development Fund Projects (Grant No. 236Z0301G). Meanwhile, this work was supported by the computing power provided by the Hebei Artificial Intelligence Computing Center.

Data availability The data for this study can be obtained by request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

Declarations

Conflict of interest The authors have no conflict of interest to declare that are relevant to the content of this article.

References

1. Mišeikis J, Caroni P, Duchamp P, Gasser A, Marko R, Mišeikienė N, Zwilling F, Castelbajac C, Eicher L, Früh M, Früh H (2020) Lio-a personal robot assistant for human–robot interaction and care applications. IEEE Robot Autom Lett 5(4):5339–5346. https://doi.org/10.1109/LRA.2020.3007462
2. Qi J, Ma L, Cui Z, Yu Y (2023) Computer vision-based hand gesture recognition for human-robot interaction: a review. Complex Intell Syst 2023:1–26
3. Zhou D, Shi M, Chao F, Lin C-M, Yang L, Shang C, Zhou C (2018) Use of human gestures for controlling a mobile robot via adap-

tive CMAC network and fuzzy logic controller. Neurocomputing 282:218–231

4. Gao Q, Liu J, Ju Z (2020) Robust real-time hand detection and localization for space human-robot interaction based on deep learning. Neurocomputing 390:198–206
5. Dipietro L, Sabatini AM, Dario P (2008) A survey of glove-based systems and their applications. IEEE Trans Syst Man Cybern Part C (Appl Rev) 38(4):461–482
6. Rautaray SS, Agrawal A (2015) Vision based hand gesture recognition for human–computer interaction: a survey. Artif Intell Rev 43:1–54
7. Pisharady PK, Saerbeck M (2015) Recent methods and databases in vision-based hand gesture recognition: a review. Comput Vis Image Underst 141:152–165
8. Tang H, Liu H, Xiao W, Sebe N (2019) Fast and robust dynamic hand gesture recognition via key frames extraction and feature fusion. Neurocomputing 331:424–433
9. Dadashzadeh A, Targhi AT, Tahmasbi M, Mirmehdi M (2019) Hgr-net: a fusion network for hand gesture segmentation and recognition. IET Comput Vis 13(8):700–707
10. Luo Y, Cui G, Li D (2021) An improved gesture segmentation method for gesture recognition based on CNN and YCBCR. J Electr Comput Eng 2021:1–9
11. Aithal CN, Ishwarya P, Sneha S, Yashvardhan C, Kumar D, Suresh K (2021) Hand gesture recognition in complex background. In: International conference on cognition and recognition. Springer, London, pp 243–257
12. Zheng Y, Zheng P (2015) Hand segmentation based on improved gaussian mixture model. In: 2015 international conference on computer science and applications (CSA). IEEE, London, pp 168–171
13. Chuang Y, Chen L, Chen G (2014) Saliency-guided improvement for hand posture detection and recognition. Neurocomputing 133:404–415
14. Zhao Y, Song Z, Wu X (2012) Hand detection using multi-resolution hog features. In: 2012 IEEE international conference on robotics and biomimetics (ROBIO). IEEE, London, pp 1715–1720
15. Chen Q, Georganas ND, Petriu EM (2008) Hand gesture recognition using Haar-like features and a stochastic context-free grammar. IEEE Trans Instrum Meas 57(8):1562–1571
16. Dardas NH, Georganas ND (2011) Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques. IEEE Trans Instrum Meas 60(11):3592–3607
17. Mocanu C, Suciu G (2019) Automatic recognition of hand gestures. In: 2019 11th international conference on electronics, computers and artificial intelligence (ECAI). IEEE, London, pp 1–5
18. Zhang Q-Y, Zhang M-Y, Hu J-Q (2009) Hand gesture contour tracking based on skin color probability and state estimation model. J Multimed 4(6):1
19. Popov PA, Laganière R (2022) Long hands gesture recognition system: 2 step gesture recognition with machine learning and geometric shape analysis. Multimed Tools Appl 81(28):40311–40342
20. Kang B, Tan K-H, Jiang N, Tai H-S, Tretter D, Nguyen T (2017) Hand segmentation for hand-object interaction from depth map. In: 2017 IEEE global conference on signal and information processing (GlobalSIP). IEEE, London, pp 259–263
21. Shotton J, Fitzgibbon A, Cook M, Sharp T, Finocchio M, Moore R, Kipman A, Blake A (2011) Real-time human pose recognition in parts from single depth images. In: CVPR 2011. IEEE, London, pp 1297–1304
22. Al-Hammadi M, Muhammad G, Abdul W, Alsulaiman M, Bencherif MA, Alrayes TS, Mathkour H, Mekhtiche MA (2020) Deep learning-based approach for sign language gesture recognition with efficient hand gesture representation. IEEE Access 8:192527–192542
23. Wang S, Zhang S, Zhang X, Geng Q (2022) A two-branch hand gesture recognition approach combining Atrous convolution and attention mechanism. Vis Comput 2022:1–14
24. Benitez-Garcia G, Prudente-Tixteco L, Castro-Madrid LC, Toscano-Medina R, Olivares-Mercado J, Sanchez-Perez G, Villalba LJG (2021) Improving real-time hand gesture recognition with semantic segmentation. Sensors 21(2):356
25. Dang TL, Pham TH, Dang QM, Monet N (2023) A lightweight architecture for hand gesture recognition. Multimed Tools Appl 2023:1–19
26. Dayananda Kumar N, Suresh K, Dinesh R (2021) Depth based static hand gesture segmentation and recognition. In: International conference on cognition and recognition. Springer, London, pp 125–138
27. Das SK, Lahkar R, Antariksha K, Das A, Bora A, Ganguly A (2023) Lightweight encoder–decoder model for semantic segmentation of hand postures. In: Inventive computation and information technologies: proceedings of ICICIT 2022. Springer, London, pp 579–591
28. Zhao H, Qi X, Shen X, Shi J, Jia J (2018) Icnet for real-time semantic segmentation on high-resolution images. In: Proceedings of the European conference on computer vision (ECCV), pp 405–420
29. Li H, Xiong P, Fan H, Sun J (2019) Dfanet: deep feature aggregation for real-time semantic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9522–9531
30. Mehta S, Rastegari M, Shapiro L, Hajishirzi H (2019) Espnetv2: a light-weight, power efficient, and general purpose convolutional neural network. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9190–9200
31. Mehta S, Rastegari M, Caspi A, Shapiro L, Hajishirzi H (2018) Espnet: efficient spatial pyramid of dilated convolutions for semantic segmentation. In: Proceedings of the European conference on computer vision (ECCV), pp 552–568
32. 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
33. Hong Y, Pan H, Sun W, Jia Y (2021) Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes. arXiv Preprint arXiv:2101.06085
34. Du S, Lee J, Li H, Wang L, Zhai X (2019) Gradient descent finds global minima of deep neural networks. In: International conference on machine learning. PMLR, pp 1675–1685
35. Han K, Wang Y, Tian Q, Guo J, Xu C, Xu C (2020) Ghostnet: more features from cheap operations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1580–1589
36. Chen L-C, Papandreou G, Schroff F, Adam H (2017) Rethinking Atrous convolution for semantic image segmentation. arXiv Preprint arXiv:1706.05587
37. Xu J, Xiong Z, Bhattacharyya SP (2023) Pidnet: a real-time semantic segmentation network inspired by PID controllers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 19529–19539
38. 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
39. Yu C, Gao C, Wang J, Yu G, Shen C, Sang N (2021) Bisenet v2: bilateral network with guided aggregation for real-time semantic segmentation. Int J Comput Vis 129:3051–3068
40. Matilainen M, Sangi P, Holappa J, Silvén O (2016) Ouhands database for hand detection and pose recognition. In: 2016 6th international conference on image processing theory, tools and applications (IPTA). IEEE, pp 1–5

41. Baranyi P. Database for hand gesture recognition. https://sun.aei.polsl.pl/~mkawulok/gestures/
42. Xie E, Wang W, Yu Z, Anandkumar A, Alvarez JM, Luo P (2021) Segformer: simple and efficient design for semantic segmentation with transformers. Adv Neural Inf Process Syst 34:12077–12090
43. Peng J, Liu Y, Tang S, Hao Y, Chu L, Chen G, Wu Z, Chen Z, Yu Z, Du Y, et al (2022) Pp-liteseg: a superior real-time semantic segmentation model. arXiv Preprint arXiv:2204.02681

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