A CNN-Based Algorithm with an Optimized Attention Mechanism for Sign Language Gesture Recognition

Kai Yang, Zhiwei Yang, Li Liu(✉), Yuqi Liu, Xinyu Zhang, Naihe Wang, and Shengwei Zhang

Nanjing Normal University of Special Education, Nanjing 210038, China
lililiu@qq.com

Abstract. Sign language is the main method for people with hearing impairment to communicate with others and obtain information from the outside world. It is also an important tool to help them integrate into society. Continuous sign language recognition is a challenging task. Most current models need to pay more attention to the ability to model lengthy sequences as a whole, resulting in low accuracy in the recognition and translation of longer sign language videos. This paper proposes a sign language recognition network based on a target detection network model. First, an optimized attention module is introduced in the backbone network of YOLOv4-tiny, which optimizes channel attention and spatial attention and replaces the original feature vectors with weighted feature vectors for residual fusion. Thus, it can enhance feature representation and reduce the influence of other background sounds; In addition, to reduce the time-consuming object detection, three identical MobileNet modules are used to replace the three CSP-Block modules in the YOLOv4-tiny network to simplify the network structure. The experimental results show that the enhanced network model has improved the average precision mean, precision rate, and recall rate, respectively, effectively improving the detection accuracy of the sign language recognition network.

Keywords: Sign language · MobileNet · YOLO · gesture recognition

1 Introduction

As a silent communication tool, gesture has become an important way of communication in life. Especially on special occasions, such as communication between deaf people, vacuum conditions, etc., the convenience and significance of gestures are more evident. In recent years, the increasing advancement of the Internet of Things (IoT) and artificial intelligence (AI) technology has facilitated the utilization of human-computer interaction in numerous scenarios [1]. Consequently, gesture recognition technology has emerged as a prominent research area in both academic and industrial circles [2, 3]. People can interact with devices more conveniently using gesture recognition technology, and its related technologies and applications have tremendous potential in the context of the

B. Wang et al. (Eds.): ICMTEL 2023, LNICST 535, pp. 84–94, 2024.
https://doi.org/10.1007/978-3-031-50580-5_8

Internet of Everything. For example, gestures can be used in smart homes to control devices such as TVs, air conditioners, and refrigerators without additional controllers, significantly enhancing the user experience and operational efficiency.

There is a large quantity of research on gesture recognition algorithms. Methods based on hardware devices are more mature and widely used, with high recognition accuracy and fast recognition speed, such as wearable data gloves, Kinect, and Leap motion [4, 5], but their equipment is expensive and human-machine interaction is not smooth; methods based on machine vision, although they reduce the dependence of gesture recognition algorithms on hardware, their network models complicate and reduces the speed of gesture recognition.

Recently, deep learning has made significant progress in target detection and image classification. State-of-the-art algorithms like YOLO, SSD, RCNN, and Faster R-CNN [6–8] have achieved high accuracy rates in detecting and classifying targets. In the realm of gesture recognition, Jin et al. [9] proposed a method based on an enhanced residual network and dynamic learning rate adjustment to enhance the accuracy, robustness, and convergence speed of the recognition process. Redmon et al. [10] introduced a multi-scale convolutional feature fusion method for SSD gesture recognition, which integrated different convolutional layers to improve recognition accuracy for small and medium-sized target gestures. Ren et al. [11] presented a deep learning model for gesture recognition based on an improved YOLOv3 network combined with a Bayesian classifier to address the challenges of data vulnerability and enhance network invariance. Furthermore, Guo et al. [12] proposed a gesture interaction algorithm based on an enhanced YOLOv4 network, which can recognize gestures in complex scenarios in real-time while addressing problems such as false detections, missed detections, and a limited amount of gesture data available for recognition using the YOLOv4 algorithm.

Although the gesture recognition method based on deep learning can achieve high recognition accuracy [13], the deepening of the network layers poses a great challenge in terms of hardware cost, computation, and difficulty of training and running the neural network model for storage on storage embedded devices. Therefore, we propose the DRL-PP algorithm to improve the YOLOv4-tiny algorithm to analyze sign language images and propose the DRL-PP algorithm. This paper proposes a gesture recognition method that employs a lightweight convolutional neural network to reduce the model size while maintaining high accuracy, thus rendering the model more suitable for deployment on resource-limited mobile or embedded devices.

The article is structured as follows: Sect. 2 discusses the related work, Sect. 3 provides a detailed description of the YOLOv4-tiny network utilized in this study, Sect. 4 outlines the DRL-PP algorithm, Sect. 5 presents the experiment results, and finally, Sect. 6 concludes the paper.

2 Related Work

The prevalent approaches to sign language recognition can be categorized into two main types: sensor-based methods and computer vision-based methods, depending on the media employed.

1) Sensor-based approach. Sensors include data gloves, arm bands, smart devices, etc. Wen et al. [14] proposed a frictional electric smart glove-based sign language recognition method configured with a total of 15 frictional electric sensors to divide the sentences collected by the smart glove into word units using segmentation and reorganization new sentences created by word unit reorganization with a correct average rate of 86.67%. Ahmed et al. [15] introduced an innovative real-time sign recognition system that utilizes wearable sensory mittens consisting of 17 sensors and 65 channels. The experiment involved 75 signs performed by five Malaysian sign language (MSL) participants, comprising MSL numbers, letters, and words. The recognition accuracy was 99%, 96%, and 93.4% for numbers, letters, and words. Although the methods using sensors for sign recognition are highly flexible, they require deaf people to wear the requisite sensory equipment, which is an additional burden for deaf people [16].

2) Computer vision-based approach. Boukdir et al. [6] presented a method for recognizing Arabic sign language using a deep learning architecture. Their approach involves the use of a 2D convolutional recurring neural network (2DCRNN) model to extract features, along with a recurrent network pattern that detects relationships between frames. Additionally, a 3D convolutional neural network (3DCNN) model is used to learn spatial-temporal features from video blocks by quadruple. The cross-validation technique was employed to evaluate the performance of the proposed method, which yielded a horizontal accuracy of 92% for 2DCRNN and 99% for 3DCNN. In a separate study, Guo et al. [17] proposed a hierarchical long short-term memory (LSTM) network for sign language translation, which addresses the fact that traditional Hidden Markov models and linkage-time classification may not be able to resolve the confusion during recognition with the sentence confusing word order difficulties corresponding to the visual content in the sentence. Yu et al. [18] introduced an end-to-end sign language converter that integrates recognition and translation tasks into a unified architecture, implemented via connectionist temporal classification (CTC). This joint approach does not require temporal information, solves two interdependent sequence learning problems, and performs better on the PHOENIX14T dataset. Ren et al. [19] proposed a machine learning method to classify cancers, as pattern recognition for medical images is widely used in computer-aided cases. Wang et al. [20] used the PSO-guided self-tuning CNN to diagnosis COVID-19. Deep learning-based method can help classify medical images and efficiently improve the accuracy of diagnosis. Computer vision-based sign language recognition is suitable for real-life applications because of its uncomplicated interaction and low device dependency with guaranteed accuracy. Inspired by the above work, this paper proposes a gesture recognition method with a lightweight convolutional neural network.

3 Network Structure

The YOLO detection model utilizes the CSPdarknet53-tiny as its backbone network, and its primary structure is presented in Fig. 1. To achieve multi-scale sensing, the (104,104,64) feature map is upsampled and combined with the (52,52,128) feature map to obtain a detection path with a larger sensory field, which is directly output by the backbone network, along with the detection path that has the minimum sensory field.

These two detection paths work collaboratively to accomplish the detection task and ensure effective multi-scale sensing.

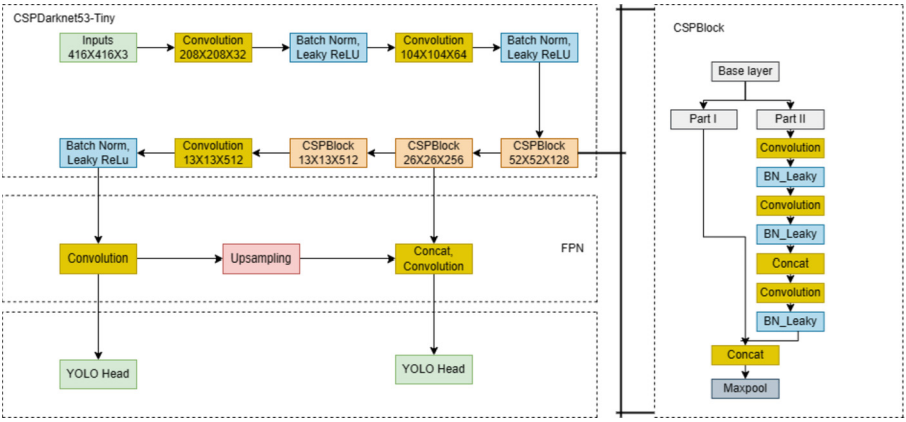

Fig. 1. YOLO model.

Although the existing Yolo-based network obtains good detection performance in a narrow space, it also has the following shortcomings: (1) The backbone network is too lightweight, and the contour evolution of the feature map is insufficient during the layer-by-layer transmission process to learn more occlusion target features effectively during the training process. (2) The neck's traditional feature fusion network (FPN) needs to be more complex, efficient in fusing between feature maps at different scales, and easy to lose edge detail information. (3) The traditional algorithm has limitations in the post-processing phase and is prone to erroneously deleting the overlapping prediction frames, leading to missed detection. A gesture recognition model based on MobileNet [14] and an attention mechanism are proposed to resolve the above problems.

4 DRL-PP Algorithm Description

In this paper, to improve the detection model's focus on the gesture region, we incorporate an attention module inspired by [15] into the backbone network. The attention module includes both channel attention and spatial attention, which can be formulated as:

$$F' = M_c(F) \otimes F \tag{1}$$

$$F'' = M_s(F') \otimes F \tag{2}$$

The channel attention and spatial attention are implemented as element-wise multiplication operations represented by the symbol ⊗. In this context, F represents the input feature map, F' represents the refined feature map, and F" represents the final refined output.

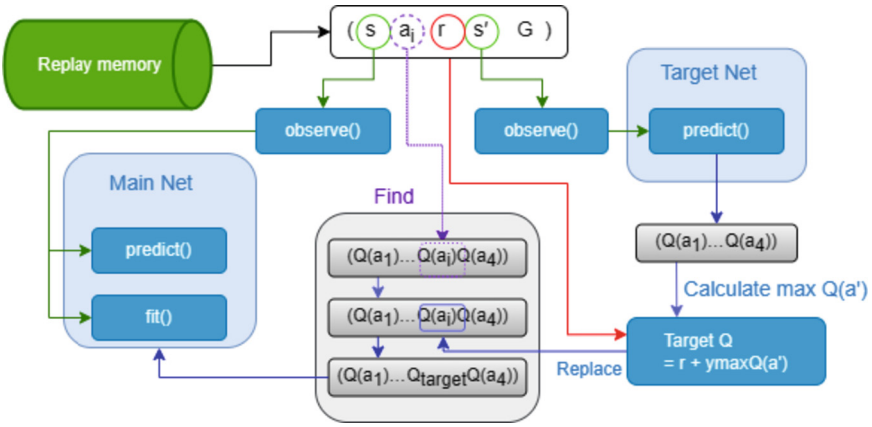

Fig. 2. Attention module.

The Channel Attention mechanism, as illustrated in Fig. 2, leverages the inter-channel relationships of feature maps to focus on the most significant portion of all channels. In particular, each convolution kernel can be viewed as a feature detector, producing a feature map that represents one object feature per channel. The first step of channel attention is to reduce the dimensionality of the feature map by passing it through both a max pooling layer and an average pooling layer, which produce two feature descriptors: one that emphasizes important features of the object and another that computes the range of the object efficiently. Next, these descriptors are fed into a shared network comprising an input layer, an output layer, and three hidden layers. As the descriptors pass through the shared network, their output feature vectors are added element-wise. Finally, the sigmoid function is applied to activate the feature vectors, generating the channel attention map.

Spatial attention, as shown in the figure, is another component of the attention module that utilizes the spatial relationship between features to generate a spatial attention map. When an image is fed into a convolutional neural network, each pixel in the image is involved in the computation. Similar to channel attention, spatial attention aims to emphasize the regions in the image that are most relevant to the object.

First, the channel attention map and the refined feature map obtained from the feature map are passed through the max pooling layer and the average pooling layer, respectively, to generate two feature descriptors. These descriptors are then concatenated and passed through two convolutional layers to accentuate the relevant regions of the descriptors. Finally, the sigmoid function is applied to activate the vector and obtain the spatial attention map.

With the channel and spatial attention modules, the weights of the feature map are optimized, resulting in a final feature map that contains more information on the relevant gesture features.

Assuming that the average pooling and maximum pooling operations are denoted by F_{avg} and F_{max}, respectively, the operation Att_{avg} is effective in removing global background information of the object while preserving the salient features of the gesture.

Let x_n denote the weight of the nth convolutional kernel. Then, the operations Att_{avg} and Att_{max}. Can be expressed as follows.

$$Att_{avg} = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{i=0}^{\infty} \gamma^i R_{t+i+1} \tag{3}$$

$$Att\max = E\left[R_{t+1} | s = s_t\right] \tag{4}$$

After the shared network, the channel attention module outputs a vector that recalibrates the importance of each channel in the feature map. This recalibration is expressed as:

$$output_{channel} = \sigma\left(output_{avg} \times output_{max}\right) \tag{5}$$

The resulting features are obtained through matrix multiplication with the original feature map, $W = [\omega_1, \omega_2, ..., \omega_n]$, which can be expressed as

$$W = (x_n, output_{channel}) = x_n \times output_{channel} \tag{6}$$

To feed W into the spatial attention module, the feature vectors undergo the average and maximum pooling layers, respectively. Then, the resulting features are concatenated along the channel dimension to obtain $C_{conv} \in R_1 \times 2C$. To obtain the feature weight information, a convolution operation is needed to let $F_{5\times5}$ denote a convolution operation with two input channels, only one output channel, and kernel size of 5×5.

$$a_i = softmax(S_i) = \frac{exp(S_i)}{\sum_{j=1}^{N} exp(S_j)} \tag{7}$$

The final output of the attention module is denoted as $\text{output}_{\text{cbsp}}\text{outputcbsp} + X$, which re-weights the importance of different elements in the original input vector. By doing so, the model is able to selectively amplify the features that contain gesture information and suppress the irrelevant or weak features. This mechanism helps the model to focus on the most salient regions in the input image and improve the accuracy of gesture recognition.

The YOLO technique uses the CSPBlock module as a residual module to enhance accuracy, which, however, increases network complexity and slows down the object detection process. To expedite gesture recognition detection, we incorporate the Mobilenet module in place of the three CSPBlock modules used in YOLO. In this study, we use Mobilenet-V1 as the backbone extraction network in DRL-PP. The fundamental concept of the Mobilenet-V1 model is Depthwise Separable Convolution. While the conventional convolution kernel convolves three channels simultaneously to obtain one number, the depthwise separable convolution first convolves three channels with three convolutions to obtain three numbers, then passes a $1 \times 1 \times 3$ convolution kernel to obtain the final number. When more and more feature attributes are extracted, the depth-separable convolution saves more parameters. Thus, the final model is shown in Fig. 3.

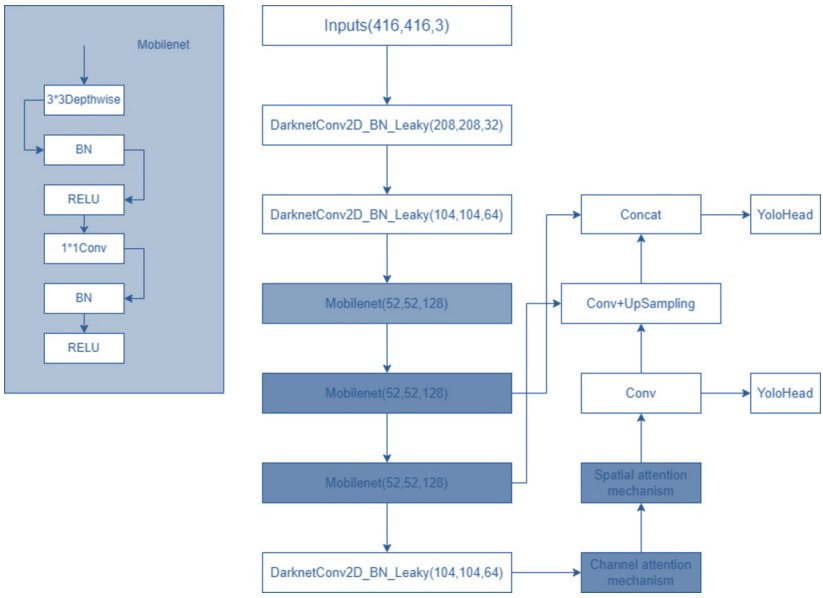

Fig. 3. Mobile Net-based gesture recognition model under occlusion conditions.

5 Evaluations

5.1 Experiment Settings

In the experiment, we use Rob Flow's American Sign Language Letters Dataset [7] for training. The dataset comprises 1728 images with an image size of 608 × 608, which contains 720 real sign language images and images expanded by data warping and oversampling. The dataset was split into a training set and a test set with a 9:1 ratio. The training set consisted of 1,555 images, while the test set contained 173 images.

The network hyperparameters were set as follows: during model training, the Adam optimizer was used to tune the parameters, with a category confidence threshold of 0.5 for the target. The initial learning rate was set to 0.001, and a weight decay coefficient of 0.0005 was employed to avoid overfitting. The batch size was set to 16, and the model was trained for 300 epochs.

The investigations are conducted on the Ubuntu operating system, using PyCharm software for programming, CUDA 10.2, cuDNN 7.6.5, as the compilation language. CPU is Intel(R) Pentium(R) G3260 @3.30 GHz, GPU is NVIDIA GTX 3090, and 1 TB hard disk.

All experiments in this paper were trained under Linux using the PyTorch framework, with a training configuration of 16 batches of 64 training samples per iteration—the number of iterations corresponding to the loss function on the homemade gesture dataset. The model attained convergence around 10000 iterations with a loss value of around 0.1.

5.2 Experiment Parameters

In the field of target detection, the performance of target detection algorithms is commonly evaluated using precision, recall, and mean average precision (*mAP*) [20]. Precision rate P, which is the ratio of true positive samples to the total samples predicted as positive by the model, can be calculated as shown in Eq. (8).

$$P = \frac{TP}{TP + FP} \times 100\% \tag{8}$$

The notation TP represents the count of positive samples correctly classified by the model, while FP refers to the number of negative samples wrongly classified as positive. The recall R measures the proportion of true positive samples correctly classified by the model among all the positive samples in the test set, which can be calculated using Eq. (9).

$$R = \frac{TP}{TP + FN} \times 100\% \tag{9}$$

where *FN* refers to positive samples erroneously identified as negative. If an algorithm performs relatively well, it should have the following performance: the value of the precision rate remains at a high level while the recall rate increases. A comprehensive parameter is also generally required to test the algorithm performance of the network, such as the *mAP* value, which is calculated as shown in Eq. (10).

$$mAP = \frac{1}{C} \sum_{K=1}^{N} P(K) \Delta R(K) \tag{10}$$

Here, N represents the total number of samples in the test set, while C is the number of categories involved in the detection task. *P(K)* signifies the precision rate achieved by the model when it identifies K samples simultaneously, and *R(K)* denotes the change in the recall rate when the number of samples identified by the model changes from K - 1 to K. These two measures are used to calculate the mean average precision (mAP), which provides an overall assessment of the model's performance.

5.3 Experiment Analysis

To evaluate the impact of various modules on detection performance, only modifications were made to the YOLO backbone extraction network, and the resulting detection results are presented in Table 1.

Tests were conducted on YOLO. The results demonstrate that Yolo with Mobilenet-V1 improves the algorithm runtime. Although the Mobilenet-V1 algorithm diminishes accuracy, the degradation is only 1% to 2%. The improvement is evident in the algorithm training time loss, which is improved by one-third. The loss function images of the YOLO algorithm for the two structures are shown in Fig. 4.

Similarly, to compare the effect of the attention mechanism on detection results, YOLO + CSPBlock, DRL-PP and attention module were used as comparison models, and the results are shown in Fig. 5.

Table 1. Impact of the upper sampling table module.

Methods	Accuracy(%)	Time(s)
Yolo + CSPBlock	92.7	0.8
DRL-PP	89.5	0.5

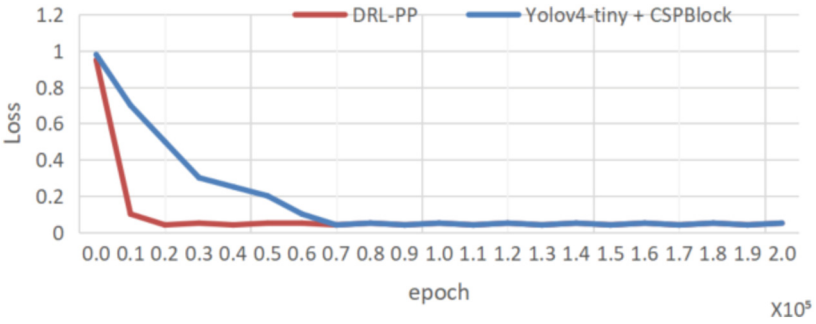

Fig. 4. Yolo-tiny loss function plot.

As depicted in Fig. 6, among the four methods, the DRL-PP approach achieved superior performance. By employing MobileNet as the backbone model, the accuracy, recall, and average precision of the algorithm were improved by 1.2%, 6.5%, and 2.1%, respectively, compared to the original model. The enhanced attention module proposed in the paper has a recall advantage, suggesting that the model pays more attention to the region containing detailed gesture information. To avoid missed detections, the confidence threshold (0.35) is appropriately lowered during inference to ensure that more finger region boundaries can be detected.

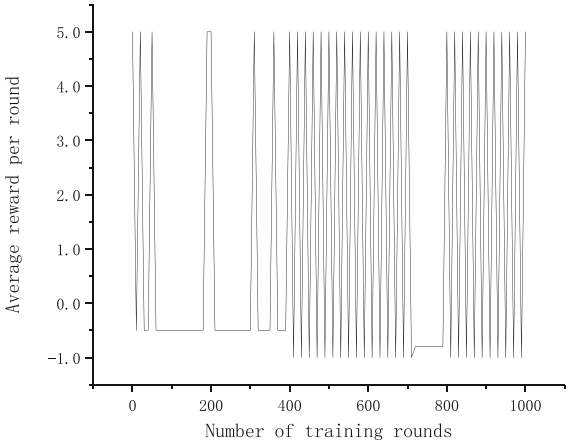

Fig. 5. Training process.

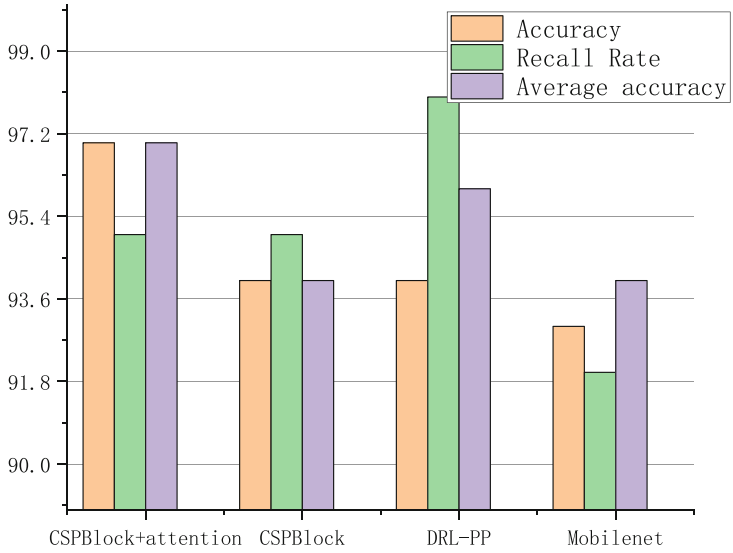

Fig. 6. Impact of attention module (%).

6 Conclusion

This paper proposes a solution to the challenge of communication between non-disabled individuals and those who are deaf or hard of hearing, through a lightweight convolutional neural network-based gesture recognition detection algorithm called DRL-PP. To improve gesture recognition accuracy, the proposed algorithm combines two approaches. First, MobileNet is used as the backbone extraction network of YOLO to reduce computation and parameters. Second, a self-attentive mechanism is incorporated into the YOLO network to capture richer contextual information. The combination of these approaches compensates for accuracy loss due to the model's lightweight. Experiment results demonstrate that the DRL-PP algorithm outperforms other methods in sign language gesture recognition, effectively addressing social isolation among deaf individuals and bridging communication gaps between normal and deaf individuals.

Acknowledgements. This work was supported by Universities'Philosophy and Social Science Researches Project in Jiangsu Province. (No. 2020SJA0631 & No. 2019SJA0544) & Educational Reform Research Project(No.2018XJJG28) from Nanjing Normal University of Special Education.

References

1. Redmon, J., Divvala, S., Girshick, R., et al.: You only look once: Unified, real-time object detection. In: IEEE CVPR2016 Conference on Computer Vision and Pattern Recognition, pp. 779–788. IEEE Computer Society Press, Washington DC (2016)

2. Wang, P., Huang, H., Wang, M., et al.: YOLOv5s-FCG: an improved YOLOv5 method for inspecting riders' helmet wearing. J. Phys: Conf. Ser. **2024**, 012059 (2021)
3. Woo, S., Park, J., Lee, J.Y., et al.: CBAM: convolutional block attention module. In: Proceedings of the 15th European Conference on Computer Vision, Munich, 3–19 (2018)
4. Zhu, R., Huang, X., Huang, X., Li, D., Yang, Q.: An on-site-based opportunistic routing protocol for scalable and energy-efficient underwater acoustic sensor networks. Appl. Sci. **12**(23), 12482 (2022)
5. Berman, M., Triki, A.R., Blaschiko, M.B.: The Lovasz-Softmax Loss: a tractable surrogate for optimizing the intersection-over-union measure in neural networks. In: Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4413–4421 (2018)
6. Boukdir, A., Benaddy, M., Ellahyani, A., et al.: Isolated video-based Arabic sign language recognition using convolutional and recursive neural networks. Arab. J. Sci. Eng. **47**, 2187–2199 (2022)
7. Oz, C., Leu, M.c.: American Sign Language word recognition with a sensory glove using artificial neural networks. Eng. Appl. Artif. Intell. **24**(7), 1204–1213 (2011)
8. Camgoz, N.c., Koller, O., Hadfield, S., et al.: Sign language transformers: joint end-to-end sign language recognition and translation. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10020–10030 (2020)
9. Jin, X., Lan, C.L., Zeng, W.J., et al.: Style normalization and restitution for generalizable person re-identification. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3140–3149. IEEE, Seattle, WA, USA (2020)
10. Redmon, J., Farhadi, A.: YOLOv3; an incremental improvement. arXiv: 1804.02767 (2018)
11. Ren, S., He, K., Girshick, R., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2017)
12. Guo, X.J., Sui, H.D.: Application of improved YOLOv3 in foreign object debris target detection on airfield pavement. Comput. Eng. Appl. **57**(8), 249–255 (2021)
13. Chao, H.Q., He, Y.W., Zhang, J.P., et al.: Gait set: regarding gait as a set for cross-view gait recognition. Proceedings of the AAAI Conference on Artificial Intelligence **33**, 8126–8133 (2019)
14. Zheng, H.L., Wu, Y.J., Deng, L., et al.: Going deeper with directly-trained larger spiking neural networks. Proceedings of the AAAI Conference on Artificial Intelligence **35**(12), 11062–11070 (2021)
15. Guo, D., Zhou, W.G., Wang, M., et al.: Hierarchical LSTM for sign language translation. In: Proceedings of the 32 ND AAAI Conference on Artificial Intelligence, the 30th Innovative Applications of Artificial Intelligence Conference and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence, pp. 6845–6852 (2018)
16. Yu, S.Q., Tan, D.L., Tan, T.N.: A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition. In: 18th International Conference on Pattern Recognition (ICPR'06), pp. 44–444. IEEE, Hong Kong, China (2006)
17. Camgoz, N.C., Hadfield, S., Koller, O., et al.: Neural sign language translation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7784–7793. IEEE Computer Society, Piscataway, NJ (2018)
18. Zhang, S.J., Zhang, Q.: Sign language recognition based on global-local attention. J. Vis. Commun. Image Represent. **80**(7), 103280 (2021)
19. Ren, Z., Zhang, Y., Wang, S.: A hybrid framework for lung cancer classification. Electronics **11**(10), 1614 (2022). May
20. Wang, W., Pei, Y., Wang, S.H., Gorrz, J.M., Zhang, Y.D.: PSTCNN: Explainable COVID-19 diagnosis using PSO-guided self-tuning CNN. Biocell