Artificial Intelligence and Digital Technology

Article

Gesture Recognition-Based Sign Language Translation System

Mengsen Yao [1,*], Chen Zhou [1], Zhixiong Liu [1] and Anchi Zhang [1]

[1] School of Computer and Information Engineering, Bengbu University, Bengbu, Anhui, 233000, China
* Correspondence: Mengsen Yao, School of Computer and Information Engineering, Bengbu University, Bengbu, Anhui, 233000, China

Abstract: To address communication barriers between deaf-mute individuals and non-sign language users, a gesture-based sign language translation system was developed for the real-time translation of sign language into text or speech. The system utilizes the YOLOv9 model and transfer learning techniques, integrating deep learning and natural language processing (NLP) to achieve gesture recognition and translation. The system design encompasses data preprocessing, feature extraction, model training and optimization, and real-time translation processing modules, adopting an end-to-end architecture to optimize user experience. Experimental results demonstrate that the proposed system exhibits superior performance in sign language recognition accuracy, response speed, and translation quality.

Keywords: gesture recognition; sign language translation; deep learning; natural language processing; Yolov9 model

Published: 13 January 2026

Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Sign language is the most critical communication tool in the daily lives of the deaf community. Its unique mode of expression conveys not only linguistic information but also rich emotional and cultural connotations. However, due to the complexity and diversity of sign language gestures, as well as significant differences in grammar and expression habits compared to spoken languages, communication between deaf individuals and the hearing population has long faced significant barriers.

In recent years, the rapid development of Information Technology and Artificial Intelligence has provided new directions for sign language translation research. Nevertheless, existing technologies still face numerous limitations. Traditional statistical learning-based methods, such as SVM and HMM, show certain advantages in processing simple gestures but lack sufficient recognition accuracy and generalization capability when dealing with complex dynamic gesture sequences and semantic translation requirements [1].

With the rise of deep learning, CNN and RNN have been widely applied to sign language recognition tasks, demonstrating significant advantages in feature extraction and temporal modeling [2,3]. For instance, the YOLO series, known for efficient object detection capabilities, has been introduced to the field of gesture recognition. However, early YOLO versions exhibited insufficient robustness in complex backgrounds and low-light scenarios [4].

Furthermore, most existing sign language translation systems rely on Autoregressive Models. While their word-by-word translation mechanism ensures semantic accuracy, the inference speed is often slow, making it difficult to meet real-time interaction demands [5].

To address these issues, this study proposes a sign language translation system based on gesture recognition [6]. We employ an improved YOLOv9 model to achieve precise detection and recognition of sign language gestures. Additionally, we utilize a Non-

Autoregressive Model (NAR) based on the Transformer architecture to achieve efficient sign-to-text conversion [7]. YOLOv9 offers significant improvements in detection accuracy and efficiency. By integrating Multimodal Alignment Technology, we further enhance the correlation between gestures and semantics, addressing recognition challenges under complex backgrounds and dynamic lighting conditions [8]. To satisfy real-time requirements, the system introduces Hardware Acceleration and latency optimization strategies, utilizing image denoising and frame buffering mechanisms to significantly reduce recognition latency [9]. The ultimate goal is to develop an efficient, precise, and multi-language adaptable sign language translation system to support communication for the deaf community and promote the application of this technology in education, healthcare, and public services [10].

2. YOLOv9 Model

YOLOv9, proposed in February 2024, is a single-stage object detection model that retains the classic "Backbone → Neck → Head" pipeline. Its core innovations are Programmable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN).

As shown in Figure 1, the backbone network of YOLOv9 is based on the lightweight CSPDarknet × 0.75. Through five down-sampling operations, it generates five levels of feature maps, P_3 – P_7. The GELAN architecture is embedded into each residual block, preserving input information through reversible residual paths. This alleviates information bottlenecks and the vanishing gradient problem, while simultaneously enhancing the perception capability for targets of different scales.

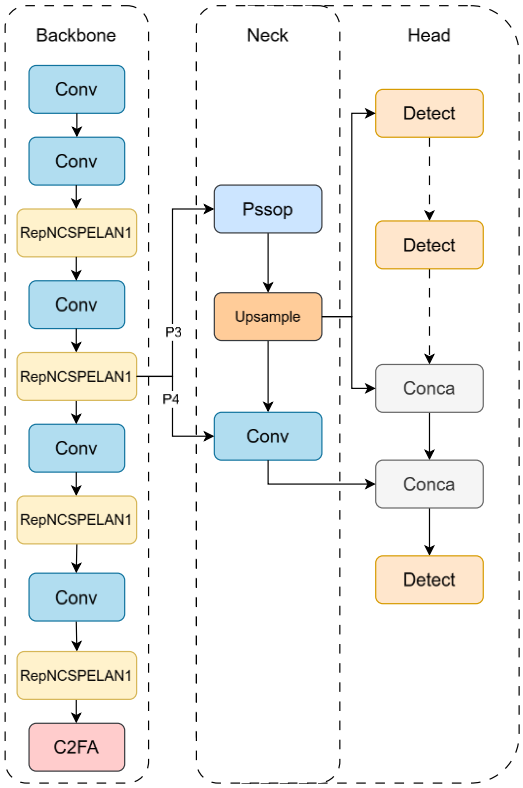

Figure 1. YOLOv9 model structure.

3. Improved YOLOv9 Model

Aiming at the problems of small target size, rich posture changes, and strong background interference in gesture recognition tasks, the original YOLOv9 model is prone to missed detection and false detection in complex environments, especially in scenes with overlapping fingers, rapid movement, or occlusion. Therefore, considering the balance between detection accuracy, model lightweight, and inference efficiency, the following improvements are made on the basis of YOLOv9 (see Figure 2).

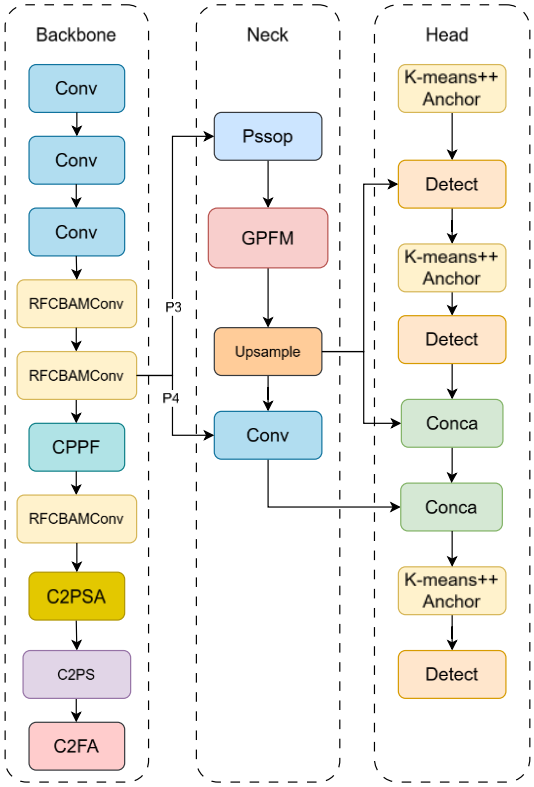

Figure 2. The architecture of the YOLOv9 model.

The backbone network of the original YOLOv9 model has insufficient ability to capture fine-grained features in dynamic gesture recognition tasks, especially in complex backgrounds and occlusion conditions, which easily leads to vague expression of key regions. To solve this problem, this paper introduces the RFCBAMConv module to replace the original C3k2. This module integrates a multi-scale convolution structure with a channel-spatial attention mechanism: first, it uses parallel convolution branches of 3 × 3 and 5 × 5 to extract multi-scale features, enhancing the modeling ability of different sizes and local structures; then, it introduces a combined channel attention and spatial attention mechanism (CBAM), which effectively highlights the response of key hand regions while maintaining the original feature channel information, and suppresses redundant background noise interference. On the whole, RFCBAMConv greatly improves the model's ability to perceive the details of gesture actions and enhances the robustness of targets in complex postures and rapid movement scenes without significantly increasing the computational overhead. After introducing RFCBAMConv, the model shows better detection accuracy and stability on multiple small-scale gesture test sets. The structure of the improved model is shown in Figure. 3.

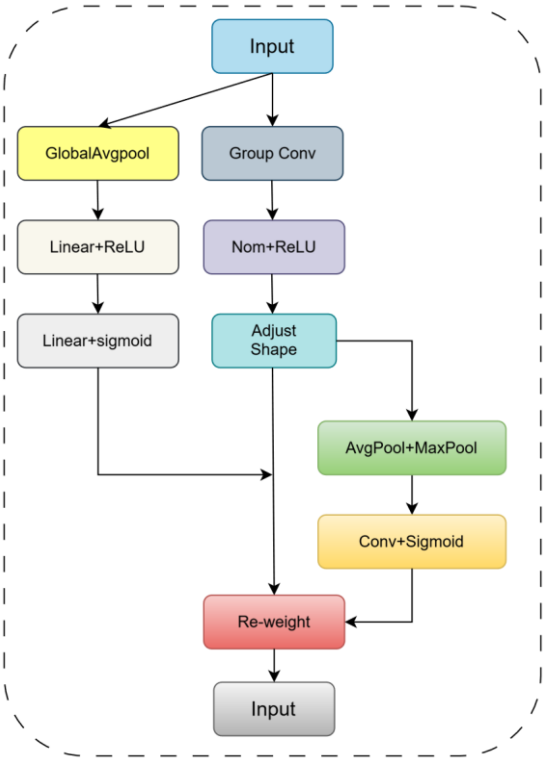

Figure 3. RFCBAMConv model structure.

3.1. Integration of the RFCBAMConv Module

During the design of the RFCBAMConv module, the focus was placed on improving feature discrimination and robustness under complex visual conditions. The design concept draws on the idea of jointly modeling channel-wise and spatial information to guide the network toward more informative regions, thereby enhancing its ability to capture fine-grained action features and suppress irrelevant background interference [11]. By embedding attention-based feature recalibration into the convolutional process, the module strengthens the representation of salient features while maintaining overall structural stability.

At the same time, the introduction of dilated convolution provides effective support for expanding the receptive field without increasing parameter complexity or reducing feature map resolution. This approach enables the model to capture contextual information at multiple scales and improves its adaptability to variations in target size and spatial distribution [12]. On this basis, a parallel branch structure that combines 3 × 3 and 5 × 5 convolution kernels is employed, allowing the module to integrate complementary local and broader contextual features. As a result, the RFCBAMConv module maintains strong feature representation capability even in scenarios with complex backgrounds and dynamic variations.

In the construction of the GPFM module, the design emphasizes multi-scale feature aggregation and efficient information fusion. The spatial pyramid-based idea is adopted as a reference, while traditional pooling operations are replaced with multi-scale convolution to alleviate the loss of spatial information that often occurs during downsampling [13]. This modification helps preserve fine-grained spatial details and enhances the continuity of feature representations across scales.

Furthermore, cross-layer feature fusion strategies are incorporated to improve the robustness and expressiveness of the model. The integration of features from different depths enables the network to effectively combine high-level semantic information with

low-level spatial details, thereby improving overall feature consistency and stability [14]. In addition, the effectiveness of multi-scale dilated convolution in capturing contextual information has been demonstrated in related tasks, providing methodological support for employing dilated convolutions with different dilation rates in the GPFM module to further expand the receptive field [15]. To enhance the effectiveness of feature integration, a learnable weighted fusion strategy is introduced, which allows the network to adaptively balance the contributions of features from different paths and scales, drawing on established insights into feature fusion mechanisms [16].

In summary, the RFCBAMConv and GPFM modules proposed in this study jointly enhance the model's ability to extract discriminative and robust features by integrating attention mechanisms, multi-scale convolution, and adaptive feature fusion strategies. Through these designs, the overall detection accuracy and robustness of the model in dynamic gesture recognition tasks are effectively improved, particularly under complex background conditions and varying spatial scales.

3.2. Improved Feature Fusion Module: GPFM

To address insufficient robustness caused by complex backgrounds and gesture deformation, we incorporate the Global Perception Feature Module (GPFM) based on receptive field expansion and adaptive feature fusion theory.

GPFM operates as follows (see Figure 4):

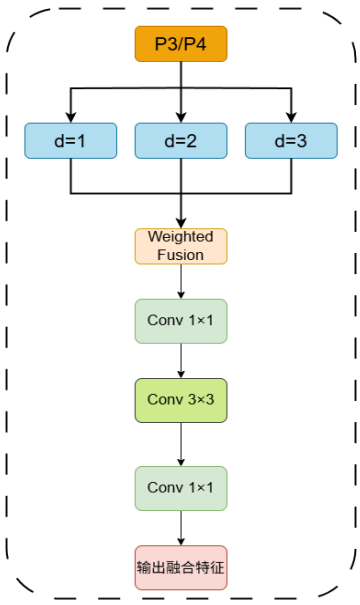

Figure 4. GPFM model structure.

Dilated Convolutions: Upon receiving multi-scale feature maps from the backbone, GPFM applies parallel dilated convolutions with three dilation rates (d = 1, 3, 5). This significantly expands the receptive field while maintaining resolution, balancing global semantics with local details.

Weighted Fusion: A learnable weighted fusion layer is introduced to adaptively weight high-frequency information from shallow layers and low-frequency semantics from deep layers.

Pyramid Convolution: Finally, a pyramid convolution sequence (1 × 1 3 × 3 1 × 1) enhances interaction in channel and spatial dimensions.

This design avoids spatial information loss caused by pooling and improves detection robustness. Experimental results on the EgoGesture dataset show that the model with GPFM improved mAP@0.5 by approximately 3.2% compared to the baseline.

4. Gesture Dataset Introduction

4.1. Dataset Selection and Creation

Current public gesture datasets are generally applied to dynamic gesture recognition, while datasets for static gesture recognition are scarce. Common examples include the American Sign Language (ASL) dataset and the NUS-II gesture dataset. However, due to the small sample size of NUS-II (only 2,750 images), it was not used in this experiment.

This experiment utilizes the SY Dataset, constructed by our team. To ensure data validity, the SY dataset comprises 3,612 items in the training set, 344 in the validation set, and 172 in the test set. Additionally, we collected datasets suitable for training and evaluating YOLOv8 to compare with our improved YOLOv9 model.

4.2. Visual Comparison of Detection Performance

To systematically evaluate the proposed algorithm, we conducted comprehensive comparative experiments using YOLOv8 as the baseline against the improved YOLOv9 algorithm. The tests covered gesture instances with varying complexity, occlusion conditions, and scales.

Visual comparison results (Figure 5) indicate that the improved YOLOv9 model outperforms the baseline in all test scenarios:

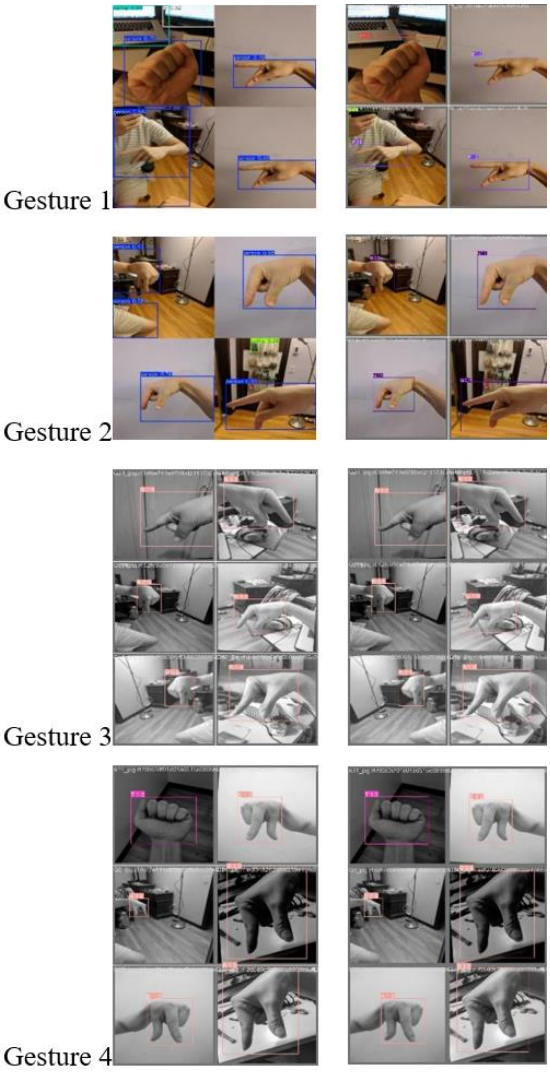

Figure 5. Gesture dataset.

Gesture Set 1: Enhanced capability to capture continuous gestures and edge-blurred targets.

Gesture Set 2: Superior anti-interference capability and feature discrimination in scenarios with occlusion and complex backgrounds.

Gesture Set 3: Improved sensitivity and localization precision for small-scale gestures.

Overall, the improved model exhibited higher recognition robustness and accuracy across different gesture scenarios, particularly showing distinct advantages in the representation of weak features, the recovery of occluded targets, and the discrimination of similar categories. The results indicate that the introduced improvement mechanisms effectively enhanced the model's ability to extract gesture semantic information, providing a more reliable technical foundation for sign language translation systems in complex real-world scenarios. YOLOv9

4.3. Dataset Characteristics

The dataset ensures experimental diversity through varying lighting conditions, background environments, presentation angles, and distances. Furthermore, class balance is maintained across training and test sets to avoid model bias.

5. Simulation Experiments and Result Analysis

5.1. Experimental Parameters and Environment

The experimental platform adopts the Windows 11 64-bit operating system, equipped with a 12th Gen Intel (R) Core (TM) i9-12900H (2.50 GHz) processor. The system memory is 16GB, and the graphics card is an Nvidia GeForce RTX 3060 with 6GB of video memory. The PyTorch framework version is 1.11.0+cu113, and the Python version is 3.8.20. The Batch size is set to 16, image size to 640×640, and epochs to 300. The network structure used in the experiment is modified based on the official YOLOv9 open-source code, loading its pre-trained model weights, and completing model training and validation under a unified hyperparameter configuration.

5.2. Detection Metrics

To detect targets more accurately, a comprehensive and precise evaluation metric system is required. Among them, mAP@0.5, precision, recall, parameter count, and GFLOPs are key metrics for measuring model performance and efficiency:

Mean Average Precision (mAP): As a comprehensive performance metric, mAP fuses the model's precision and recall. mAP@0.5 focuses on the average detection precision of all categories when the IoU threshold is 0.5; it evaluates model performance more strictly, where a higher mAP value represents superior detection effects. Precision: Starting from the perspective of detection results, precision refers to the proportion of the number of correctly detected targets to the total number of detected targets, reflecting the accuracy of the model's detection results. Recall: Based on real targets, recall measures the proportion of the number of targets the model can detect to the total number of real targets, embodying the model's ability to capture real targets. Parameters: Used to evaluate the scale and complexity of the model, obtained by accumulating the weight parameter values of each layer of the model. A smaller parameter count implies a more lightweight model, while a larger parameter count, though helpful for learning complex features, will occupy more storage and computational resources. GFLOPs: As a metric for measuring the model's computational complexity and execution efficiency, GFLOPs represents the number of floating-point operations the model executes per second; the value intuitively reflects the complexity and efficiency level of the model's operations.

5.3. Visualization of Experimental Data

Through multiple experiments, the results were organized into visual charts to facilitate analysis and comparison. Figure 6 shows the variation curves of the loss function

and evaluation metrics during the target detection model training process. According to the analysis of experimental results, it can be known that during the training process, the model's learning effects for tasks such as bounding box prediction and category classification gradually improved; performance on the validation set was stable with small errors. The comprehensive performance, such as detection precision and the ability to identify positive and negative samples, is constantly improving.

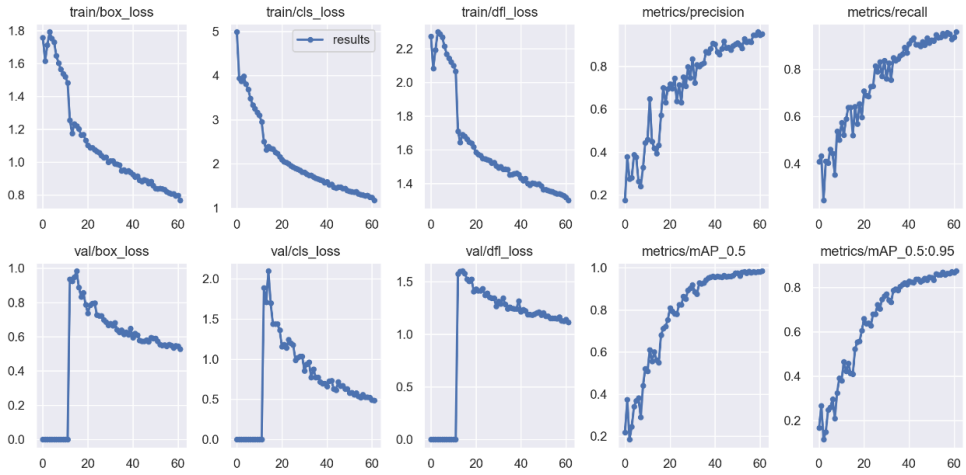

Figure 6. The change curves of loss functions and evaluation metrics during the training process of the object detection model.

The confusion matrix shown in Figure. 7 further reveals the model's performance in classification tasks. The color gradient of the matrix reflects the prediction proportion, where dark blue indicates a high prediction proportion and light color indicates a low prediction proportion. Observing the matrix, it can be seen that most dark blue squares are located on the diagonal.

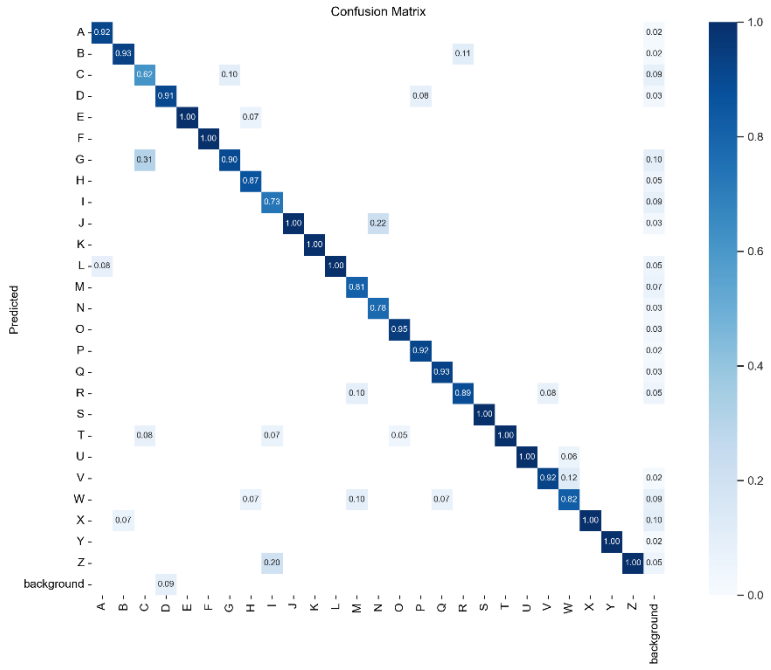

Figure 7. confusion matrix.

The analysis results of comprehensive indicators and confusion matrix show that the model can achieve high detection and classification accuracy in most categories, but there is still room for optimization in distinguishing a few specific categories. In the future, the performance of the model in confusing categories can be improved through further data augmentation, feature extraction optimization, or category weight adjustment.

5.4. Comparative Experiments

Table 1 compares the detection results of YOLOv5, YOLOv6, YOLOv7, YOLOv8, and the proposed model (Ours). According to the data comparison of the detection results, the YOLOv9 model integrated with multiple modules has made great progress in various aspects and still performs excellently in real-time detection. Therefore, in subsequent experiments, our team chose to integrate YOLOv9 with the RFCBAMConv module and the GPFM module to improve the detection effect of target precision detection.

Table 1. Comparative experiments of different models.

Model	mAP50-90	FPS
YOLOv5	75.7%	52
YOLOv6	77.9%	50
YOLOv7	78.2%	55
YOLOv8	81.6%	58
Ours	90.3%	62

To comprehensively verify the improvement effect of each improved module on model performance, ablation experiments are conducted on the proposed improved model on the self-built SY dataset. These experiments are based on the YOLOv9 model, and the improved methods are integrated step by step according to the modules to quantify the contribution of each module to detection accuracy and efficiency (see Table 2).

Table 2. Ablation experiments.

Baseline	RFCBAMConv	GPFM	mAP50-90	recall	precision
√			0.757	0.888	0.868
√	√		0.862	0.947	0.944
√	√	√	0.903	0.979	0.980

Experimental results indicate that the synergy of modules can significantly enhance model detection capability. First, after adding the RFCBAMConv module to the baseline YOLOv9 model, mAP@50-90 increased by 10.5%. This indicates that the RFCBAMConv module effectively mitigated the information loss problem of traditional convolution in multi-scale feature fusion, significantly enhancing sensitivity and adaptability to targets of different scales. Compared with the baseline, it eliminated extreme cases of zero recognition rate, proving that the RFCBAMConv module played an important role in feature representation equalization.

Secondly, examining the experimental data when adding the GPFM module to the baseline YOLOv9 model, mAP@0.5 increased by 4.9%. This indicates that adopting the GPFM module not only reduced the loss of detailed information but also improved the ability to capture multi-scale fine-grained features, contributing to the stable detection of small targets in complex scenes.

Finally, when the baseline YOLOv9 model is integrated with both the RFCBAMConv module and the GPFM module, a technical complementarity is formed. RFCBAMConv focuses on local feature refinement and attention focusing, while the GPFM module emphasizes global feature equalization. The combined use of both forms a complete feature optimization solution. Technically, it ensures that key gestures are rarely missed; furthermore, stable low-confidence thresholds can reduce false triggers. In terms of user

experience, it can bring a smoother real-time translation experience, reducing translation interruptions caused by detection failures (see Table 3).

Table 3. Comparative experiments of various model versions.

Model Version	mAP@0.5	Precision	Recall	mAP@0.5:0.95
YOLOv9 Baseline	92.7%	97.8%	96.9%	75.7%
+RFCBAMConv	94.1%	98.0%	97.2%	86.2%
+GPFM	97.6%	98.1%	97.4%	80.6%
Full Model	98.5%	98.2%	97.6%	90.3%

The loss function of the target detection model YOLOv9 consists of three parts: Localization Loss, Confidence Loss, and Classification Loss. Their expressions are as follows:

Localization Loss:

$$L_{loc} = \sum_{i=1}^{N} 1_{\{obj_i\}} \left((x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2 \right) \quad (0.1)$$

Confidence Loss:

$$L_{conf} = \sum_{i=1}^{N} 1_{\{obj_i\}} \left(\hat{C}_i - C_i \right)^2 + \lambda_{noobj} \sum_{i=1}^{N} 1_{\{noobj_i\}} \left(1 - \hat{C}_i \right)^2 \quad (0.2)$$

Classification Loss:

$$L_{cls} = \sum_{i=1}^{N} 1_{\{obj_i\}} \sum_{c=1}^{C} (p_{c,i} \, log(\hat{p}_{c,i})) \quad (0.3)$$

Gesture Recognition Model: Adopts the acoustic model CNN+CTC (Connectionist Temporal Classification) to recognize each frame in the sign language video. The Convolutional Neural Network (CNN) is used to extract gesture features, and CTC is used to avoid the need for forced alignment of gesture sequences.

In the process of system acceleration, the Speed-up Ratio (S) is defined to measure the performance improvement of hardware acceleration, with the formula as follows:

$$S = \frac{T_{baseline}}{T_{optimized}} \quad (0.4)$$

Hardware acceleration can ensure the real-time performance of gesture recognition and translation. Model optimization is used to reduce computational overhead and improve system response speed. Caching and latency optimization can reduce unnecessary computational delays during the recognition process, ensuring smooth real-time feedback.

6. Conclusion

This study addresses core challenges in sign language translation systems, such as insufficient environmental robustness, real-time bottlenecks, and lack of multimodal synergy, proposing a sign language translation system based on an improved YOLOv9 and Transformer-NAR architecture. By introducing the RFCBAMConv module to enhance gesture detail perception capability, designing the GPFM feature fusion module to optimize adaptability in complex scenes, and combining with a non-autoregressive translation model to achieve single-step parallel decoding.

The system solves problems such as missed gesture detection under dynamic lighting, high latency in autoregressive translation, and multimodal semantic fragmentation, providing a low-latency, high-precision cross-modal communication tool for the deaf-mute population. Future work will focus on transfer learning for sign language dialects, enhancement of neuro-symbolic reasoning, and optimization of cross-lingual generalization capabilities, further promoting the inclusive implementation of barrier-free technology in scenarios such as education and healthcare.

Funding: This work was supported by the Bengbu University General Research Project (Grant No. 2024ZR02); the Bengbu University Quality Engineering Project (Grant No. 2024RGTSKC2); the National Undergraduate Innovation and Entrepreneurship Training Program of China (Grant No. 202411305032, 202411305033, 202511305075); and the Anhui Provincial Undergraduate Innovation and Entrepreneurship Training Program (Grant No. S202511305078).

References

1. H. Bhavsar, and J. Trivedi, "Performance comparison of svm, cnn, hmm and neuro-fuzzy approach for indian sign language recognition," *Indian J Comput Sci Eng*, vol. 12, no. 4, pp. 1093-1101, 2021.
2. K. Myagila, D. G. Nyambo, and M. A. Dida, "Efficient spatio-temporal modeling for sign language recognition using CNN and RNN architectures," *Frontiers in Artificial Intelligence*, vol. 8, p. 1630743, 2025. doi: 10.3389/frai.2025.1630743
3. A. O. Tur, and H. Y. Keles, "Evaluation of hidden markov models using deep cnn features in isolated sign recognition," *Multimedia tools and applications*, vol. 80, no. 13, pp. 19137-19155, 2021. doi: 10.1007/s11042-021-10593-w
4. A. B. Aziz, N. Basnin, M. Farshid, M. Akhter, T. Mahmud, K. Andersson, and M. S. Kaiser, "Yolo-v4 based detection of varied hand gestures in heterogeneous settings," In *International Conference on Applied Intelligence and Informatics*, October, 2023, pp. 325-338. doi: 10.1007/978-3-031-68639-9_21
5. P. Yu, L. Zhang, B. Fu, and Y. Chen, "Efficient Sign Language Translation with a Curriculum-based Non-autoregressive Decoder," In *IJCAI*, August, 2023, pp. 5260-5268. doi: 10.24963/ijcai.2023/584
6. W. Jia, and C. Li, "SLR-YOLO: An improved YOLOv8 network for real-time sign language recognition," *Journal of Intelligent & Fuzzy Systems*, vol. 46, no. 1, pp. 1663-1680, 2024. doi: 10.3233/jifs-235132
7. F. Zhou, and T. Van de Cruys, "Non-autoregressive modeling for sign-gloss to texts translation," In *Proceedings of Machine Translation Summit XX: Volume 1*, June, 2025, pp. 220-230.
8. Y. Min, A. Hao, X. Chai, and X. Chen, "Visual alignment constraint for continuous sign language recognition," In *proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 11542-11551.
9. J. M. Blair, "Architectures for Real-Time Automatic Sign Language Recognition on Resource-Constrained Device," 2018.
10. R. San-Segundo, J. M. Montero, R. Cordoba, V. Sama, F. Fernández, L. F. D'Haro, and A. García, "Design, development and field evaluation of a Spanish into sign language translation system," *Pattern Analysis and Applications*, vol. 15, no. 2, pp. 203-224, 2012.
11. S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," In *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3-19.
12. F. Yu, and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.
13. K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904-1916, 2015. doi: 10.1109/tpami.2015.2389824
14. T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117-2125.
15. L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834-848, 2017.
16. S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8759-8768. doi: 10.1109/cvpr.2018.00913

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of SOAP and/or the editor(s). SOAP and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.