RobotFlags: AI-Powered Semaphore Interacting Between Chatbot and Humanoid Robot

Yan Huan, Kien Tran, Minh Nguyen, Wei Qi Yan
Department of Computer and Information Sciences
Auckland University of Technology
Auckland, New Zealand

***Abstract*—In this paper, we introduce *RobotFlags*, an intelligent system that integrates flag language with deep learning, large language models, and humanoid robots to support learning and interactive communication. The core of this system is the improved YOLO-AKEMA model, which incorporates attention mechanisms and adaptive convolutions to achieve high-accuracy recognition across 27 flag classes, forming a reliable foundation for gesture analysis. The user interface is implemented on the Dify AI platform, with a retrieval-augmented generation (RAG) framework constructed from curated semaphore documents and the BGE-M3 embedding model, enabling context-aware responses. A humanoid robot is seamlessly integrated as both a demonstrator and an evaluator: It performs flag gestures, assesses learners' performance, and provides detailed feedback. To ensure real-time interaction, optimization strategies such as half-precision computation, streaming inference, and caching are employed, maintaining average response times under three seconds. Altogether, RobotFlags delivers a robust, multimodal learning environment that advances flag language demonstration and creates new opportunities for gesture-based human–robot interaction.**

***Index Terms*—semaphore recognition, semaphore learning system, humanoid robot, YOLO-AKEMA, DeepSeek**

I. Introduction

Flag language is a traditional, visual, and standardized form of long-distance communication, transmitting information through precise configurations of flags or arms [1]. It remains an essential component of training in educational institutions, maritime academies, and military programs. Traditional instructional approaches demand the memorization of numerous gesture-angle configurations, thereby imposing significant cognitive demands on beginners.

Currently, an AR card-based flag signal learning system is employed to assist Indonesian scouts in understanding flag semaphore and to enhance their interest in learning [2]. While the system has demonstrated positive outcomes, it is limited to static media displays and app-based input operations. It lacks support for interactive human-computer communication and does not accommodate diverse presentation formats or media carriers.

In order to enhance the interactivity and display carrier diversity of the flag language learning system, we developed RobotFlags, an innovative interactive platform. This system employs a deep learning-based flag signal recognition model as its analytical foundation, and a large-scale language model as the core component for interaction and interpretation, thereby enhancing its overall intelligence. A humanoid robot is integrated to support the demonstration and assessment of flag gestures. Through the fusion of intelligent interaction, intuitive video-based analysis, and physical robot demonstrations, the system broadens the diversity of flag signal presentation methods and improves the scalability of user input modalities.

979-8-3315-8654-6/25/$31.00 ©2025 IEEE

II. Related Work

A. Pose and Flag Recognition

Early research on posture and flag recognition primarily relied on traditional image processing methods, including color segmentation, edge detection, and shape feature extraction.

In 2011, researchers firstly employed Kinect[3], [4] depth cameras as a new data source and achieved human posture recognition from single-frame depth images based on joint position estimation, providing a foundation and new direction for subsequent research and applications in posture analysis [5], [6].

The development of flag learning systems represents another important application of flag recognition algorithm [7]. Rachmad and Fuad [8] proposed a system based on bone images acquired by Kinect sensors, which is employed to assist in teaching and practicing signal flags.

Driven by the swift progress of deep learning, research work in human pose estimation and flag signal recognition has progressively moved away from traditional feature engineering, embracing deep models such as convolutional neural networks (CNNs)[9]–[11] to enable end-to-end automatic feature extraction and classification.

Compared to traditional methods, deep learning models exhibit superior feature representation capabilities and can autonomously learn robust visual features from large-scale image datasets. In 2016, there was a convolutional neural network-based model for flag signal recognition[12]. Experimental findings affirmed that this method achieves superior performance compared to conventional techniques, particularly in recognition precision, while also enhancing system stability and scalability.

Deep learning has demonstrated significant potential in the domain of flag signal recognition. Nevertheless, a notable gap persists in its practical application within instructional contexts. Humanoid robots play a pivotal role in the research and application of semaphore communication. Recent studies have

compared the effectiveness of learning semaphore through video-based interaction and robot-assisted interaction[13]. The findings reveal that robots offer distinct advantages in facilitating semaphore learning, particularly in conveying dynamic gestures, with these effects being more pronounced among adolescents. Additionally, a cloud-based human–robot interaction system has been developed to enable a humanoid robot to autonomously replicate semaphore gestures performed by a human demonstrator, further highlighting the potential of integrating robotics with semaphore communication[14]–[16]. However, this system relies solely on discrete letter-level instructions rather than dynamically captured video input, the communication remains unidirectional. Building upon these findings, in this paper, we propose the integration of a vision model within the robot to recognize semaphore movements from video streams and generate responsive feedback. Moreover, a chat-based interface is introduced to facilitate natural and interactive communication between the user and the robot.

B. Large Language Models

Large Language Models (LLMs) constitute a category of natural language processing systems that are grounded in deep neural architectures and trained on extensive text corpora.

In recent years, LLM applications have expanded rapidly across various industries. Numerous studies have demonstrated the potential across diverse scenarios. In the education sector in particular, LLMs show great promise. In recent research work, integrating LLM-powered chatbots into higher education database courses has been shown to enable personalized learning and real-time feedback, thereby alleviating teaching workload [17]. In enterprise scenarios, LLMs have been combined with Retrieval-Augmented Generation (RAG) and frameworks such as LangChain to enhance information retrieval and generative services, ultimately improving operational efficiency [18], [19]. These projects reflect the growing versatility of LLMs across sectors. However, despite longstanding efforts in semaphore teaching, the integration of LLMs into this domain remains unexplored.

III. Methodology

The RobotFlags system consists of two main components: Flag recognition and user interaction through DeepSeek. The following sections provide a detailed introduction to each component.

The system architecture is shown in Fig. 1. The basic service is responsible for data interaction with the user interface, issuing commands to the humanoid robot and receiving feedback from the robot. The humanoid robot's core controller is a Raspberry Pi, with an integrated head camera for visual processing. It also has 18 servos throughout its body, whose parameters are adjusted to achieve the desired gesture display and movement of the robot's flag signals.

YOLO11 presents notable enhancements over its earlier version, offering a more compact architecture, fewer parameters, improved feature representation, and accelerated inference speed. A recent investigation utilized a synthetically

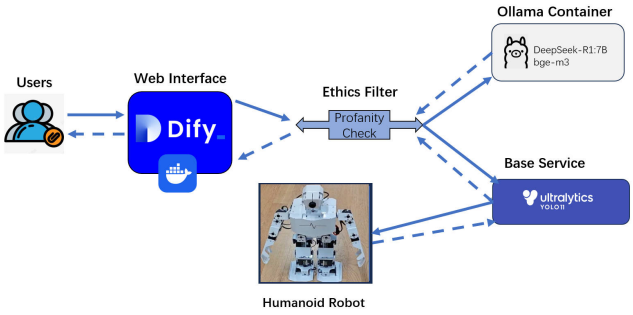

Fig. 1. System architecture

constructed dataset—produced through a large language model (LLM)—to develop apple detection models by using both YOLO11 and YOLOv10. The study demonstrated that the integration of synthetic data can improve model robustness and generalization. On real-world orchard imagery, YOLO11 attained a detection accuracy of 0.84 and a mAP@50 score of 0.89, confirming its effectiveness in object detection tasks [20]. Given these results, YOLO11 was selected for the subsequent experimental phases.

The EMA (Efficient Multi-Scale Attention) component functions as a compact attention strategy aimed at improving the representational strength of CNNs in applications like visual categorization and object recognition [21]. To enhance the model sensitivity to the flag-bearing arm region and improve the quality of feature representations, the present study incorporates the EMA module into the YOLO network as an attention mechanism. AKConv (Arbitrary Kernel Convolution) represents an innovative convolutional operator aimed at addressing the structural constraints of traditional convolution kernels in terms of fixed sampling shapes and constrained parameter quantities [22]. In contrast to standard convolution operations, it enhances the expressiveness of learned features and leads to notable performance improvements in various multi-object. YOLO11 was enhanced through the integration of the Efficient Multi-Scale Attention (EMA) and Arbitrary Kernel Convolution (AKConv) modules, forming the improved YOLO-AKEMA model.

For the large language model, we chose the DeepSeek-R1:7b model and deployed it with Ollama. Unlike conventional approaches relying on supervised fine-tuning (SFT), the research team firstly developed DeepSeek-R1-Zero, trained entirely via RL without human-labeled data [23].

The embedded model BGE-M3 is applied to vectorize the textual knowledge of the flag signals and build a knowledge base. Driven by the swift evolution of Large Language Models (LLMs), Retrieval-Augmented Generation (RAG) has become a prominent framework for enhancing the knowledge representation capabilities of such models and improving the accuracy of real-time question answering [24]. The maturity of text embedding models and the development of efficient vector retrieval tools have significantly accelerated the adoption of RAG architectures. These systems have been widely applied

in diverse scenarios, including intelligent question answering, enterprise knowledge management, and scientific research assistance[25].

Dify is an open-source AI application tool. We leverage its ChatFlow feature with a rich set of functional nodes and workflow orchestration to enable user interaction with the entire system. When interacting with AI applications, users frequently impose more stringent requirements regarding content security, user experience, and compliance with legal and regulatory standards. Specifically, we integrated a fast and effective Python-based tool, profanity-check, to identify profane and offensive expressions in text. This tool employs a linear Support Vector Machine (SVM) model trained on a dataset of 200,000 manually annotated examples, encompassing both clean and inappropriate textual content.

IV. Experimental Results

A. Dataset collecting and processing

Video data acquisition was conducted by using the DJI Action4 sports camera. Flag signal videos were recorded under diverse weather conditions and across multiple scene types, with several individuals performing the gestures. The recorded gestures covered the full set of alphabetical characters from A to Z as well as the STOP action. All frames were extracted at 10-frame intervals from the recorded videos. Postural landmarks, specifically at the elbow and wrist, were extracted by using the MediaPipe framework. Images in which the posture could not be reliably identified were discarded.

The dataset was subsequently divided into training and validation sets in an 8:2 ratio. All images were resized to 640×640 pixels to comply with the default input dimensions of the YOLO training pipeline. At this stage, the training set comprised 17,569 images, while the validation set contained 4,403 images. Then, the training dataset is expanded to 34,566 after data augmentation.

B. Evaluation Indicators

Correctness: Correctness directly reflects whether the system output is consistent with the reference answer or the ground-truth label.

$$\text{Correctness} = \frac{N_{correct}}{N_{total}} \tag{1}$$

where $N_{correct}$ denotes the number of correct responses, while N_{total} refers to the total number of evaluated samples.

Latency: Latency refers to the time interval between user input and system output. In the context of this system, it specifically denotes the duration from the moment a user submits an input to the moment feedback is returned.

$$\text{Latency}_{avg} = \frac{1}{M}\sum_{j=1}^{M}\left(t_j^{out} - t_j^{in}\right) \tag{2}$$

where M denotes the total number of interaction trials, t_j^{in} represents the input timestamp of the j-th interaction, and t_j^{out} represents the corresponding output timestamp. The difference $(t_j^{out} - t_j^{in})$ indicates the latency of the j-th interaction. By averaging across all M trials, this metric quantifies the overall responsiveness of the system, with lower values corresponding to faster response times and improved user experience.

Precision: Precision refers to the proportion of true positive predictions among all instances identified as positive by the model.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

where TP denotes the count of actual positive cases that are correctly identified, while FP represents the number of samples that are falsely predicted as positive.

mAp: mAP is calculated by averaging the AP over all categories. To calculate AP, we need to interpolate the PR curve (precision - recall). Here shows that how AP is calculated:

$$P_{\text{interp}}(r) = \max_{r' \geq r} P(r') \tag{4}$$

where the interpolated precision corresponding to a specific recall threshold r is defined as the highest precision value observed at any recall level r' such that $r' \geq r$.

$$\text{AP}_{c,t} = \sum_{i=1}^{n-1}(r_{i+1} - r_i) \cdot P_{\text{interp}}(r_{i+1}) \tag{5}$$

where r_i and r_{i+1} denote adjacent recall values, and $P_{\text{interp}}(r_{i+1})$ is the interpolated precision at the right endpoint of each interval. This summation serves as a numerical approximation of the integral of precision over recall.

$$\text{mAP}_t = \frac{1}{C}\sum_{c=1}^{C}\text{AP}_{c,t} \tag{6}$$

where C is the total number of object classes, and $\text{AP}_{c,t}$ denotes the Average Precision for class c at IoU threshold t. This equation provides a comprehensive measure of the model's ability to detect objects of different categories at a specified localization accuracy level. Commonly used settings include $t = 0.5$ (mAP@50) and the COCO-style average over multiple thresholds from 0.50 to 0.95 (mAP@50:95), the latter offering a stricter and more robust performance evaluation.

C. Experiments for Flag Recognition

The 27-class flag signal recognition experiment based on YOLO11 were implemented using Python 3.11.6 and PyTorch 2.6.0. The hardware configuration includes an NVIDIA GeForce RTX 4070 Laptop GPU with 8188 MiB of memory, as well as an AMD 7940HX processor. All YOLO models adopted in this study belong to the nano variant category. Specifically, YOLO11, YOLO12, and YOLO-AKEMA were employed for comparative evaluation. In the model training experiment, the hyperparameters were configured as follows: SGD was employed as the optimizer; the batch size was 32; the epochs were 50; the input image size was set to 640 × 640 pixels; the learning rate was 0.01; and the number of workers was 16.

D. Experiments For RobotFlags

An interactive interface was developed by using Dify and connected to DeepSeek, while a text embedding model was employed to construct a retrieval-augmented generation (RAG) framework. The flag signal analysis model provided video analysis and additional functionalities to the system components, while also enabling communication with the humanoid robot.

E. Ablation Experiments

Ablation experiments were conducted to evaluate the individual contributions of the attention mechanism, AKConv convolution module, and horizontal flip parameter in data augmentation within the YOLO-AKEMA framework. Each experiment was designed to isolate the effect of a single component through controlled variable settings. The results were analyzed to assess the specific impact of each module on overall model performance.

F. Results for YOLO

This section presents the experimental results and conducts comparative analyses using representative images, charts, and tables.

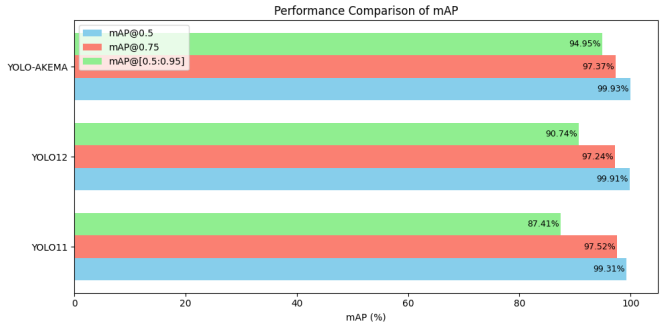

Fig. 2. Performance Comparison of mAP

From **Fig.** 2, mean average precision (mAP) results of the three models YOLO11, YOLO12, and YOLO-AKEMA are compared under different Intersection over Union (IoU) thresholds. The results indicate that YOLO-AKEMA achieves the best performance across all metrics, with an mAP@0.5 of 99.93%, an mAP@0.75 of 97.37%, and an mAP@[0.5:0.95] of 94.95%. Based on the actual video prediction results, YOLO-AKEMA was able to accurately recognize all flag signal actions at a confidence threshold exceeding 0.9.

In contrast, YOLO11 attains an mAP@[0.5:0.95] of only 87.41%. While YOLO12 demonstrates a slight improvement over YOLO11 in terms of accuracy, its performance still remains below that of YOLO-AKEMA. For the intended flag signal interaction system, it is essential that the model generates accurate and unique category predictions for each image. Therefore, greater emphasis is placed on the mAP@[0.5:0.95] metric, as it provides a more rigorous evaluation of overall model performance under varying levels of localization precision.

TABLE I
ABLATION EXPERIMENTS RESULTS

Flipud	AKConv	EMA	Precision	GFLOPs	mAP@0.5:0.95
✓	✗	✗	65.81%	6.7	63.03%
✗	✗	✗	94.87%	6.7	86.42%
✗	✗	✓	97.92%	6.9	90.12%
✗	✓	✗	96.92%	6.4	89.12%
✗	✓	✓	**99.94%**	6.5	**94.95%**

From Table I, the results reveal that retaining horizontal flipping during training has a significant adverse effect on the baseline model. This is primarily due to the presence of two mirrored flag gestures within a single category. Such ambiguity is evident in pairs such as B and F, J and P, K and V, M and S, and Q and Y. The rate of misclassification for these categories approaches 50%, resulting in a substantial degradation of model performance. Evaluation metrics clearly show a marked improvement when this data augmentation parameter is disabled, confirming its negative impact on classification accuracy.

The final experimental outcomes indicate that the improved model exhibits strong performance in terms of both accuracy and robustness YOLO-AKEMA model is capable of stably and accurately recognizing all target categories in the test video under a high-confidence threshold. These results reflect the model's strong generalization ability and its high potential for practical deployment.

G. Results for RobotFlags

TABLE II
SYSTEM EVALUATION ON DIFFERENT COMPONENTS

Component	Correctness (%)	Latency (s)
Robot Execution	99%	1
Video Analysis	99%	13
Video-Image Response	99%	3
RAG Response	80%	2

As shown in Table II, the system's correctness is slightly lower in the knowledge base retrieval component, while other functional components maintain a high level of correctness. This limitation arises from the insufficient richness of the flag knowledge descriptions in the knowledge base, as the available text materials do not fully cover the diversity of possible user inputs. With respect to average latency, the video analysis component exhibits a longer processing time, primarily due to the computational demands of the analysis itself. Nevertheless, this does not impose a significant negative impact on the user experience, and the overall system performance remains smooth and reliable.

By adopting the aforementioned approach, the proposed system has been successfully implemented and deployed. As show in **Fig.**3, the system currently supports natural language interaction, enabling users to inquire about the meaning of flag action videos through dialogue, either in the form of individual

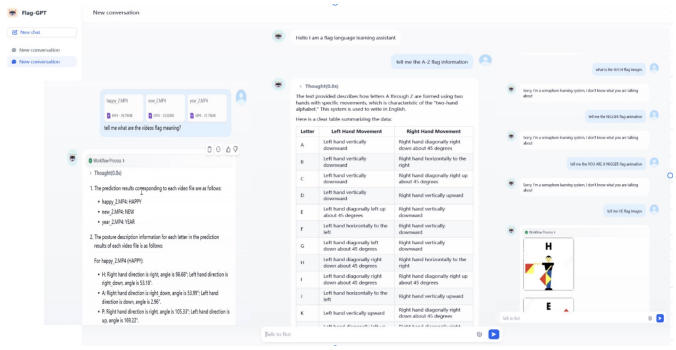

Fig. 3. RobotFlags Interface

words or complete sentences. Additionally, the system allows users to input single or multiple flag action letters, to which it responds with the corresponding textual descriptions, representative images, or generated videos, depending on the user's request. The system also adds insult-to-harm filters based on ethical considerations, making it more humane.

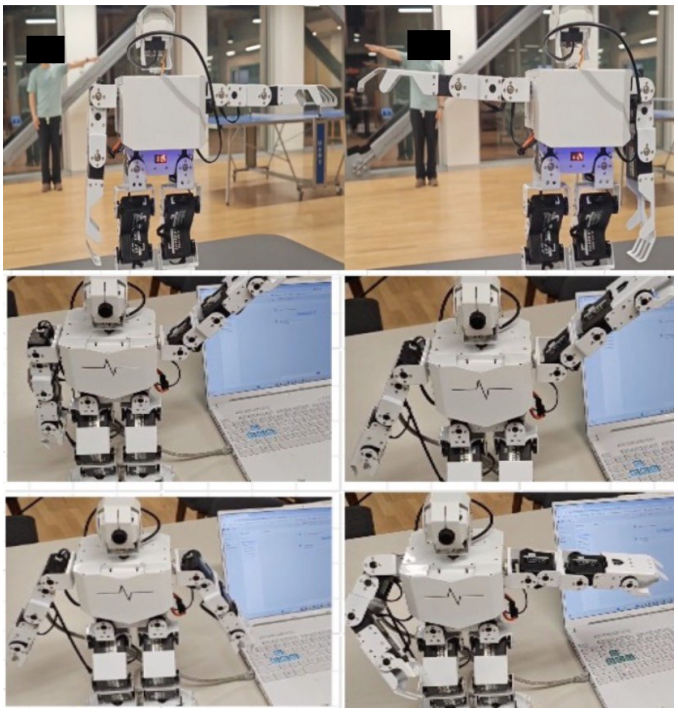

Fig. 4. Robot display and feedback flag movements

As shown in **Fig.** 4, a humanoid robot was integrated into the interactive system, enabling it to perform corresponding flag movements for input words or sentences, accompanied by synchronized voice playback. This design allows users to perceive flag movements more intuitively, thereby reducing the learning difficulty and enhancing the overall learning experience. At the same time, the robot will make corresponding flag movements according to the user's actions and explain the key points of the movements.

V. Discussion

After a series of experiments, the implementation and deployment of the RobotFlags system have been completed and achieved the desired goals. In this section, we discuss the key challenges encountered during the initial research.

Firstly, YOLO11 improves prediction accuracy by introducing an attention mechanism and replacing the standard convolutional module. The improved YOLO-AKEMA effectively improves the accuracy of flag gesture prediction. These are key components of the overall system functionality, providing stable flag video analysis capabilities and laying the foundation for subsequent processes.

In the actual deployment, we adopted techniques such as half-precision computation, streaming inference, and secondary verification to ensure accuracy and computational efficiency. However, after the initial inference, a caching strategy is employed to eliminate redundant computations. Pertaining to text generation, image-based responses, video synthesis, and other interactive features, the average response time remains under 3 seconds. Overall, the system's performance is stable and ensures a seamless user experience across multiple interaction modes.

The RAG system currently exhibits low accuracy, with frequent errors occurring during the keyword matching phase of the recall test. This limitation may stem from insufficient data diversity and the absence of a structured text format. Nevertheless, the current performance of the RAG component does not impede the system's overall functionality, as the primary focus remains on video analysis, robotic interaction, and visual content presentation. Future improvements to the RAG module will involve reducing redundant textual content, adopting structured text formats, incorporating comparative queries, optimizing search strategies, and potentially substituting the underlying model.

The integration of robotic systems enhances the presentation of flag signals and improves the interactivity and usability of the system. Building upon this concept, further applications can be explored by combining computer vision and robotics, such as integrating visual models, large language models, and extended reality (XR) technologies, as well as incorporating natural voice control for robotic interaction.

VI. Conclusion

To address the limited functionality of existing flag learning systems, we proposed the RobotFlags prototype. By enhancing the YOLO11 architecture, the system ensures accurate flag motion video analysis and incorporates an intelligent interactive interface. The integration of a humanoid robot to display and provide feedback on flag gestures further enhances system interactivity and user engagement. This research work demonstrates the feasibility of combining computer vision, large language models, and humanoid robotics for flag recognition, offering new insights into the design of multimodal human–computer interaction systems. However, the current RAG component exhibits limited accuracy and lacks support for natural voice control. Future work will focus on improving

the RAG module, fine-tuning the large language model, and integrating natural voice interaction with XR visualization technologies to enable more innovative and immersive approaches to flag learning and interaction.

References

[1] H. Mead, "The story of the semaphore," *The Mariner's Mirror*, vol. 21, no. 1, pp. 33–55, 1935.

[2] M. B. P. Nugraha, G. M. Darmawiguna, and G. B. Subawa, "Semaphore AR card: Interactive scout learning media," *Jurnal Inovasi Teknologi Pendidikan*, vol. 11, no. 3, pp. 352–365, 2024, Published Sep 30, 2024.

[3] W. Q. Yan, *Robotic Vision: From Deep Learing to Autonomous Systems*. Singapore: Springer, 2025.

[4] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with Microsoft kinect sensor: A review," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1318–1334, 2013.

[5] J. Shotton, A. Fitzgibbon, M. Cook, *et al.*, "Real-time human pose recognition in parts from single depth images," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1297–1304, 2011.

[6] Y. Liu, P. Nand, M. Nguyen, A. Hossain, and W. Q. Yan, "Sign language recognition from digital videos using feature pyramid network with detection transformer," *Multimedia Tools and Applications*, 2023.

[7] S. Helena, "Learning application on signal scout semaphore multimedia and web-based using computer assisted instruction method," *Proceedings of the 3rd International Conference on Information Technology and Business (ICITB)*, pp. 91–96, 2015.

[8] A. Rachmad and M. Fuad, "A geometry-based algorithm for semaphore gesture recognition using skeleton images," *Journal of Theoretical and Applied Information Technology*, vol. 81, no. 1, pp. 102–107, 2015.

[9] Y. Huan and W. Q. Yan, "Semaphore recognition using deep learning," *Electronics*, vol. 14, no. 2, p. 286, 2025.

[10] Y. Huan, "ChatFlags: AI-Powered Semaphore Interactive System," Master's Thesis, Auckland University of Technology, Auckland, New Zealand, Jun. 2025.

[11] W. Q. Yan, *Computational Methods for Deep learning: Theory, Algorithms, and Implementations*. Singapore: Springer, 2023, ISBN: 978-981-99-4823-9.

[12] Z. Qian, Y. Li, N. Yang, Y. Yang, and M. Zhu, "A convolutional neural network approach for semaphore flag signaling recognition," *IEEE International Conference on Signal and Image Processing (ICSIP)*, pp. 466–470, 2016.

[13] S.-W. Hsieh and Y.-C. Shih, "Using bioloid robots as tangible learning companions for enhancing learning of a semaphore flag-signaling system," *The Asian Conference on Education & International Development : Official Conference Proceedings*, 2015.

[14] N. Tian, B. Kuo, X. Ren, *et al.*, "A cloud-based robust semaphore mirroring system for social robots," *14th International Conference on Automation Science and Engineering (CASE)*, pp. 1351–1358, Aug. 2018.

[15] H. Zhao, S. Xu, and W. Yan, "Design and optimization of target detection and 3D localization models for intelligent muskmelon pollination robots," *Horticulturae*, vol. 11, no. 8, p. 905, 2025.

[16] K. Zhao, M. Nguyen, and W. Yan, "Evaluating accuracy and efficiency of fruit image generation using generative ai diffusion models for agricultural robotics," *IEEE IVCNZ*, 2024.

[17] A. T. Neumann, Y. Yin, S. K. Sowe, S. J. Decker, and M. Jarke, "An LLM-driven chatbot in higher education for databases and information systems," *IEEE Transactions on Education*, vol. 68, no. 1, pp. 103–116, 2024.

[18] C. Jeong, "Generative AI service implementation using LLM application architecture: Based on RAG model and langchain framework," *Journal of Artificial Intelligence and Machine Learning*, vol. 5, no. 4, pp. 123–135, 2023.

[19] J. Cheonsu, "A study on the implementation of generative AI services using an enterprise data-based LLM application architecture," *Advances in Artificial Intelligence and Machine Learning*, vol. 3, no. 4, pp. 1588–1618, 2023. [Online]. Available: https://arxiv.org/abs/2309.01105.

[20] R. Sapkota, Z. Meng, and M. Karkee, "Synthetic meets authentic: Leveraging LLM generated datasets for YOLO11 and YOLOv10-based apple detection through machine vision sensors," *Smart Agricultural Technology*, p. 100 614, 2024.

[21] D. Ouyang, S. He, G. Zhang, *et al.*, "Efficient multi-scale attention module with cross-spatial learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2023, pp. 1–5.

[22] X. Zhang, Y. Song, T. Song, *et al.*, "AKConv: Convolutional kernel with arbitrary sampled shapes and arbitrary number of parameters," *arXiv preprint arXiv:2311.11587*, 2023. [Online]. Available: https://arxiv.org/abs/2311.11587.

[23] Y. Zhang, J. Liu, Z. Wang, L. Chen, and S. Yang, "Challenges in ensuring AI safety in DeepSeek-R1 models," *arXiv preprint arXiv:2501.17030*, 2025. [Online]. Available: https://arxiv.org/abs/2501.17030.

[24] M. Arslan, H. Ghanem, S. Munawar, and C. Cruz, "A survey on RAG with LLMs," *Procedia Computer Science*, vol. 246, pp. 3781–3790, 2024. [Online]. Available: https://doi.org/10.1016/j.procs.2024.09.178.

[25] H. Yu, A. Gan, K. Zhang, S. Tong, Q. Liu, and Z. Liu, "Evaluation of retrieval-augmented generation: A survey," *arXiv preprint arXiv:2405.07437*, 2024. [Online]. Available: https://arxiv.org/abs/2405.07437.