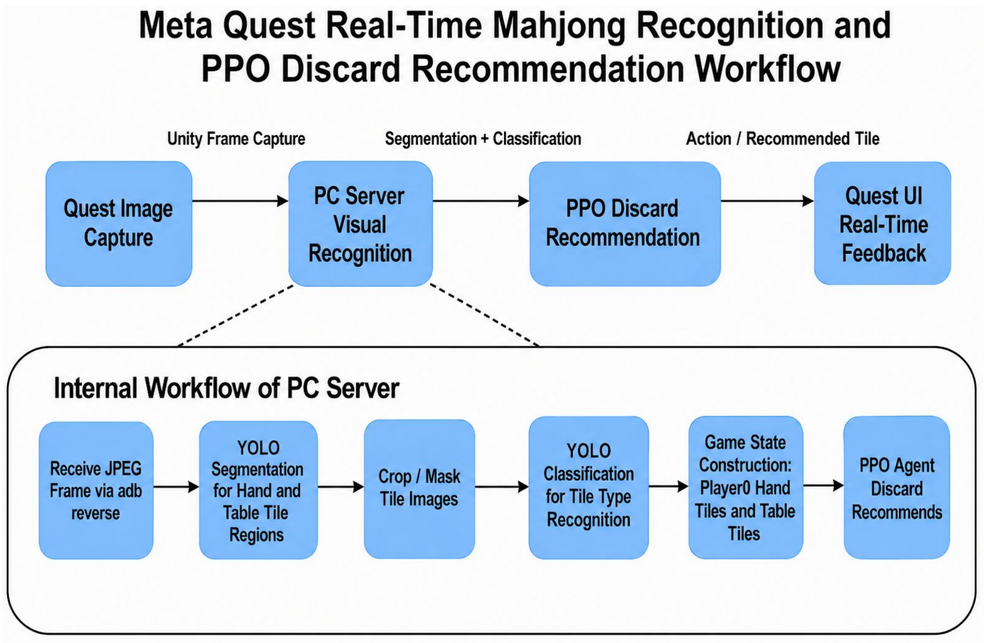
Design and Implementation of an AI Mahjong Assistance System Based on XR and Image Recognition

Yung-Chi Tan | Supervised by Hung-Jui Guo
Department of Computer Science and Information Engineering, National Cheng Kung University

Abstract

- **Traditional Mahjong** requires players to memorize complex rules, evaluate probabilities, and make **strategic decisions in real time**.
- **Novice players** often struggle to manage hand information and interpret the **evolving game state** during physical gameplay.
- **MR headsets** provide an opportunity to deliver **tactical recommendations** directly in the player's first-person view without interrupting immersion.
- Existing **Mahjong AIs** mainly operate in digital environments and cannot directly process **noisy visual information** from real-world Mahjong tables.
- Therefore, this work develops an **XR-based AI Mahjong assistance system** that combines **image recognition** and **reinforcement learning** to provide **real-time discard recommendations**.

System Architecture



Experimental & Simulation Method

- Collected Mahjong tile images and built a **YOLO (You Only Look Once)-based vision pipeline** for **Mahjong tile recognition**.
- Compared four **YOLOv8n instance segmentation** settings: **baseline**, **P2 Head**, **KD (Knowledge Distillation)**, and **P2 Head + KD** under the same training configuration.
- Used **segmentation** to locate Mahjong tile regions, then applied **classification** to identify tile types.
- Built two deployment benchmarks: **PC offload through ADB (Android Debug Bridge) reverse** and **Quest local inference.**
- Evaluated the **PPO decision model** through Mahjong game simulations, comparing **win rate**, **draw rate**, **reward**, **average steps**, and **invalid action rate**.

Sample videos

Quest pipeline | PC pipeline | Failure cases | Github link

Results & Discussion

Table 1. Baseline Selection: Model Scale and Input Size Trade-off

Route	Segmentation Setting	Image Size	Params	Mask mAP50-95 ↑	Quest Local Latency ↓	PC Offload Latency ↓
1	YOLOv8n-seg	96×96	3.26M	0.076	91.2 ms / 11.0 FPS	94.2 ms / 10.6 FPS
2	YOLOv8n-seg	320×320	3.26M	0.135	125.0 ms / 8.0 FPS	147.0 ms / 6.8 FPS
3	**YOLOv8n-seg**	**640×640**	**3.26M**	**0.411**	**281.7 ms / 3.5 FPS**	**243.6 ms / 4.1 FPS**
4	YOLOv8s-seg	640×640	11.79M	0.417	521.4 ms / 1.9 FPS	511.7 ms / 2.0 FPS
5	YOLOv8m-seg	640×640	27.24M	0.422	1037.4 ms / 1.0 FPS	964.2 ms / 1.0 FPS

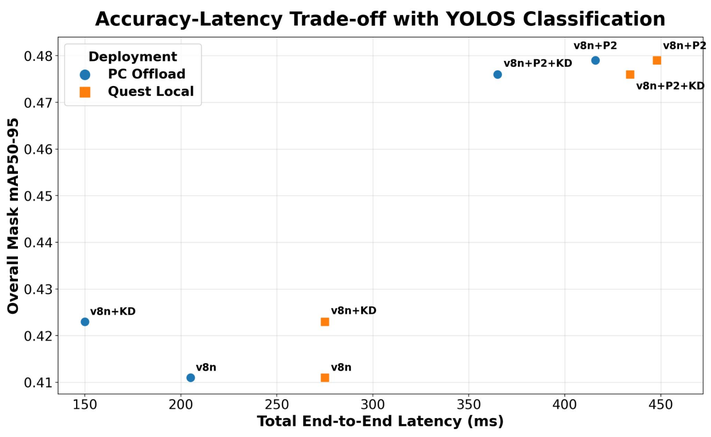
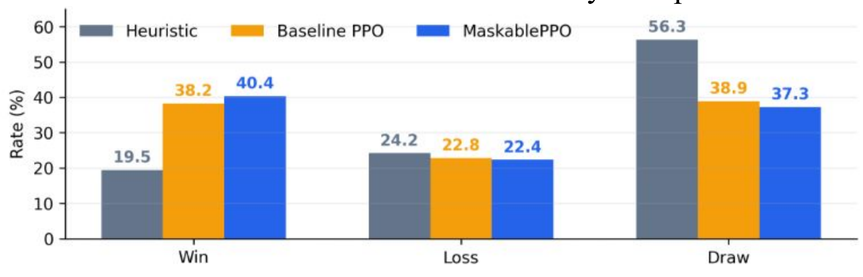


Table 2. Accuracy–Latency Trade-off (KD, P2 Head)

Segmentation Model	Mask mAP50-95 ↑	Table mAP50-95 ↑	PC Offload Latency ↓	Quest Local Latency ↓
YOLOv8n-seg	0.411	0.263	243.6 ms	281.7 ms
YOLOv8n-seg + KD	0.423	0.276	150.8 ms	275.8 ms
YOLOv8n-seg + P2	**0.479**	**0.343**	**416.2 ms**	**447.0 ms**
YOLOv8n-seg + P2 + KD	0.476	0.347	364.7 ms	433.8 ms

Chart 1. Performance and Latency Comparison



Project Results and Highlights

- Built an **end-to-end Mahjong assistance prototype** integrating **Quest image capture**, **PC server recognition**, **PPO (Proximal Policy Optimization) Agent decision-making**, and **Quest UI (User Interface) feedback**.
- Improved **Mahjong tile recognition** by addressing **small-object challenges** through **YOLO (You Only Look Once)-based segmentation**, **classification**, **knowledge distillation**, and **P2 head experiments**.
- Developed a **PPO-based decision module**, allowing the system to provide **discard suggestions** instead of only recognizing **tile information**.

Project Conclusion

- This project integrates **VR (Virtual Reality) devices**, **real-time image recognition**, **deep learning models**, and **reinforcement learning** into a complete **Mahjong assistance system**.
- The system demonstrates a complete workflow from **visual input** to **game-state interpretation** and **AI (Artificial Intelligence)-based decision feedback**.
- Based on the baseline comparison, **YOLOv8n-seg with 640 input size** was selected for its **accuracy–latency balance**, and the final model adopts a **P2 head** to prioritize **segmentation accuracy** for reliable Mahjong move recommendations.
- Future improvements include enhancing **image processing** for **occlusion** and **unstable recognition cases**.
- Further work can focus on improving **PPO (Proximal Policy Optimization) reward design** and **compressing models** so the system can eventually run independently on **Quest** without relying on a **PC (Personal Computer)**.
- The modular pipeline also provides a practical foundation for future **MR (Mixed Reality)-based tabletop game assistance systems**, showing the potential to extend this framework beyond Mahjong to other physical strategy games.