2024 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
July 15-19, 2024. Boston, MA, USA

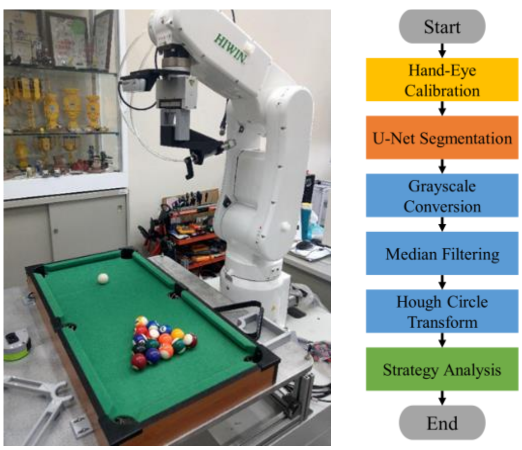
Evaluation of Pool Balls Detection Using Deep Learning Segmentation

Ya-Hsuan Wang[1], Ming-Tsung Lin[2,3,*], *Member, IEEE*, Yong-Zhong Li[1], Guan-Yi Lin[1] and You-Cheng Chen[1]
[1] Department of Electrical Engineering, National Formosa University, Yunlin 632301, Taiwan, ROC
[2] Department of Mechanical Design Engineering, National Formosa University, Yunlin 632301, Taiwan, ROC
[3]Smart Machine and Intelligent Manufacturing Research Center, Yunlin 632301, Taiwan, ROC
* Corresponding author: Ming-Tsung Lin (email: mtlin@nfu.edu.tw)

***Abstract*—This paper proposes a pool robot system (PRS) capable of detecting and locating pool balls using deep learning networks. U-Net and Mask R-CNN networks are used to extract ball features while solving issues such as glare, shadows, and noise in images. U-Net outperforms Mask R-CNN in terms of accuracy, detection time, and memory footprint, making it more suitable for real-time detection in edge computing. To validate the feasibility of PRS, experiments were conducted on a pool table. The trained model achieved a detection rate of 98.76%, with the robotic arm completing clearance in an average of 20 shots in seven games.**

I. Straight Pool Robot System

The pool robot system (PRS) is composed of a 6-axis robotic arm, a camera, a fixture with a pneumatic cylinder, 16 balls, a pool table and an industry PC illustrated in Fig. 1. The PRS is specifically designed for a pool game and assists the robotic arm to make a successful stroking.



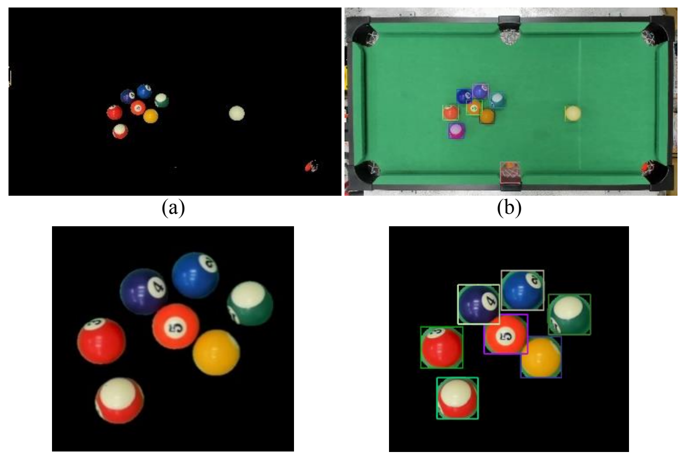
(a) (b)

Fig. 1. The pool robot system. (a) system architecture, (b) flowchart.

II. Pool Balls Detection

The proposed pool ball detection algorithm is implemented using Pytorch 2.0 on a PC with an Intel i9-9820X CPU and two NVIDIA GeForce RTX2080. The dataset of 100 images was collected from human players engaging in pool games. 70-15-15% of image dataset is used for training, validation and testing. In this setting, both U-Net and Mask R-CNN are trained. Table I gives a quantitative comparison of two object segmentation methods. It is observed that U-net outperforms Mask R-CNN in terms of mAP, mAR, and precision. Furthermore, its shorter inference time and lower memory usage make U-net more suitable for real-time object detection in edge computing.

The results of the pool ball detection are shown in Fig. 2. The deep learning networks help extract pool balls while effectively identifying them from images containing shadows and surface light reflections. Due to data augmentation and accurate object labeling, good performance is still achieved on an extremely limited dataset. After U-Net completes pool ball detection, the PRS determines their positions as well as the center and radius of the circles by applying a median filter and a Hough circle transform. This approach is different from Mask R-CNN, which directly determines ball position through bounding box shown in Fig. 2.(d). Mask R-CNN is designed for object localization and classification, and is suitable for various pool games such as 8-ball and 9-ball. In comparison, U-Net excels in real-time image segmentation, which is particularly suited for 14.1 game.



(a) (b)

(c) (d)

Fig. 2. The results of pool ball detection. (a) U-net, (b) Mask R-CNN, (c) U-Net segmentation, (d) bounding box detection of Mask R-CNN.

TABLE I. Evaluation of Deep Learning Segmentation

Model	U-net	Mask R-CNN
IoU	0.953	**0.956**
mAP	**0.998**	0.957
mAR	**0.975**	0.951
Precision	**0.973**	0.930
K-fold cross-validation	**0.950**	0.943
Inference time	**6.1ms**	110ms
Memory size	**121.3MB**	172.1MB

III. Conclusion

Deep-learning neural networks have been developed for pool ball detection to achieve high-accuracy real-time recognition. U-Net demonstrates superior performance in both accuracy and detection rate compared with Mask R-CNN. PRS can complete a stroke within 15 seconds, averages 20 strokes per game, and has a potting rate exceeding 80%. Future work will focus on optimizing deep neural network through pruning techniques to reduce both inference time and memory footprint of the model.

Acknowledgment

This work was supported by the Ministry of National Science and Technology Council of Taiwan, ROC under Contract No. NSTC 112-2221-E-150-017 and NSTC 112-2218-E-002-042.