Design and Experiment of Real-Time Detection and Tracking System for Animal Behavior Based on Improved RetinaFace Neural Network

Jing Lyu[1,2*], Sibo Wang[1], Hongbo Jia[1,2], Zhenqiao Zhou[1], Min Li[1], Yan Wang[1], Yueyan Chen[1], Yuguo Tang[1,2]

[1]*Suzhou Institute of Biomedical Engineering Technology, Chinese Academy of Sciences, Suzhou 215163, China*

[2]*School of Biomedical Engineering (Suzhou), Division of life Sciences and Medicine, University of Science and Technology of China, Suzhou 215163, China*

**Corresponding Author.*

Abstract

To solve traditional animal behavior recognition method defects such as non-intelligence, heavy workload, and poor generalization performance, improved RetinaFace neural network animal training model and NVIDA TensorRt were used to accelerate and construct the model engine. GPU asynchronous flow was used to process multiple frames of images and improve model-based reasoning speed. Finally, the real-time target detection and tracking system was built for the animal behavior. Experiment results showed that: The mAP value of the neural network model was 82.38%. Average time of forward reasoning model after improvement was 29ms and was increased by 3.58 times. The complete system can recognize and track the water maze environment in a paid and stable way and realize the real-time analysis on key behavior information of mice (such as trajectory, displacement, and exploration time).

Keywords: *Animal behavior; Deep learning; Improved RetinaFace; Model acceleration*

I. INTRODUCTION

Around 600 million people around the world suffer from CNS diseases and mental sicknesses such as epilepsy, Parkinsonism, schizophrenia, depression, Alzheimer's Disease, stroke, and drug addiction. With population aging acceleration and influence of economy, society and environment, the number of patients suffering these diseases has been increasing yearly, which has posed a huge obstacle to social and economic development[1]. Neuroscientists and neurologists are always exploring new research tools and treatment ways for nervous system diseases and mental diseases including epilepsy for years[2]. To research the relationship between nervous system and mental diseases as well as behaviors, real-time tracking and monitoring on the animal behavior of the model under the special conditions are required such as animal activity form, sound production, body posture, recognizable changes in appearance, and mutual communication role caused by these changes[3-4].

At present, in the above field, new technology (deep learning) is used for relevant research, which has got certain achievements[5-6]. For example, in the mid-nineteenth century, a famous photography research of Eadweard Muybridge had provided new thought for data analysis via computer visual and AI. Song W[7] made use of meanshift to design and realize the experiment facility and automatic analysis software of the rat limb sports behavior, which had reduced labor strength of animal behavior experiment and provided objective data. Kai S[8] made use of YOLO V3 neural network training model, combined image processing techniques with deep learning technologies to detect mice, and completed sports behavior analysis on mice. Giancardo[9] et al. designed a set of sports characteristics, used the random forest classifier to recognize interaction role of mice, and realized tracking of center, nose and tail of at rat. Lorbach[10] et al. used Noldus to track mice to get positions of centers, noses and tales of mice, thus extracting sports characteristics such as speed, distance and direction. Lars Haalck[11] et al. made use of a global reasoning method to detect the sports of animals at the messy environment. Chen Zexin, et al.[12],

ISSN: 0010-8189

from Department of Computer Science and Engineering of Shanghai Jiao Tong University, designed an animal behavior software-AlphaTracker to complete description on positions and directions of mice. Above research had gotten certain achievements, but the tracking was extracted based on traditional goals in most research and deep learning was used for optimization in less research. However, the model was generally complex, so the understanding was difficult and generalization performance was poor, which further restricted the research on the animal behavior[13].

Based on this, in the paper, the author designed an animal behavior tracking and analysis system based on deep learning which improved RetinaFace and built the data set including 2600 rat sample pictures for network training. TensorRT was used to optimize the network model[14], and GPU asynchronous flow treatment ways were used to further improve execution speed, which had realized forward reasoning time of 29ms.

II. SYSTEM SCHEME DESIGN

As shown in Fig. 1, the animal behavior real-time detection system was divided into three parts: Model training, model deployment and user application program. After training of the model, best parameters were gotten, and model optimization and forward prediction were carried out during the model deployment. In the process of the optimization, some scattered operators were combined in the neural network, models were serialized, and prediction speed of the neural network was increased under the condition of not affecting the precision. In the user application program, rat behavior analysis and data storage were designed, which was convenient for users and had improved experiment efficiency.

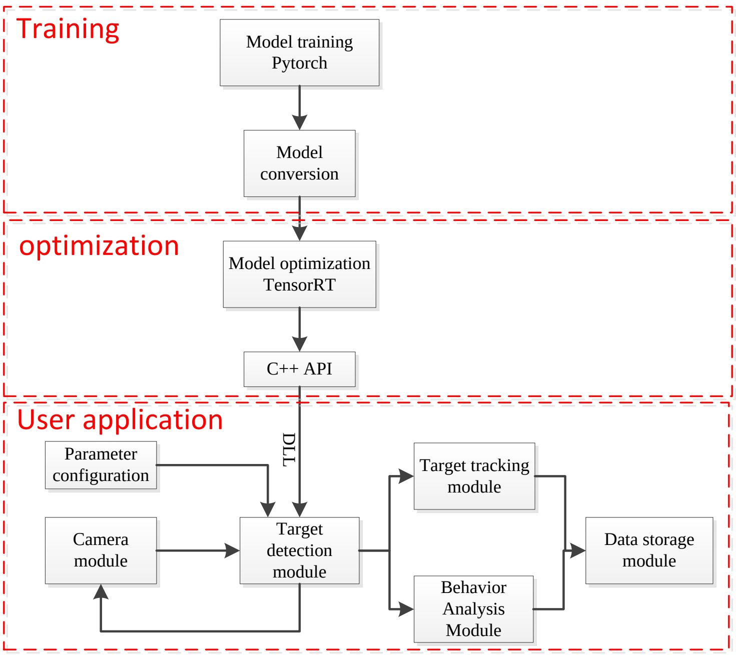


Fig.1 Design of animal behavior real-time detection system

III. TARGET DETECTION AND MODEL OPTIMIZATION

3.1 Model selection

Since the seventies, the face recognition has become one of the most research topics in the field of computer vision and biological recognition. In recent years, traditional methods, based on artificial design characteristics and traditional machine learning technique, have been replaced by Deep Neural Networks (DNN), and the model application and optimization have been evolved maturely. Seeing Small Faces from robust anchor's perspective

(SSF) was as described in the literature[15], which solved the problem – low Intersection over Union (IOU) for small faces and anchor, proposed new EMO score, showed anchor and faces to get high IOU ability, put forward some simple and effective policies to design anchor, achieved high IOU, and enhanced small face detection ability. In the literature[16], there was Dual Shot Face Detector (DSFD). DEFD inherited SSD structure, introduced an FEM (feature enhancement module), and made used of different level information, thus obtaining more identifiability and robustness. Meanwhile, proposing an improved anchor matching policy made anchor match with real faces, thus providing better initialization for regression. In the literature[17], there was a single-shotscale-aware network for real-time Face Detection (SFDet). The detection network of multiscale aware was used to deal with different sizes of faces, and scale compensation was introduced to re-set anchor matching policy and improve detection recall rate, thus improving face detection effect. In the literature[18], one single-stage field face location algorithm was called RetinaFace. In the method, light backbone network was used to realize real-time operation on the single CPU and complete multiscale face detection, face alignment, pixel-level face analysis and 3D analysis of key face density points. The comparison results of Average Precision (AP) in the WIDER FACE[19] was shown in Table 1. WIDER FACE is a data set based on face detection, and also the largest recognized face detection platform in the world. It is divided into three levels: easy, medium, and hard based on face detection difficulty in the picture.

Table I. Comparison of mAP values of face detection algorithms

Method	Detection level		
	Easy	Medium	Hard
SSF	0.949	0.933	0.961
DSFD	0.963	0.954	0.901
SFDet	0.954	0.945	0.888
RetinaFace	0.969	0.962	0.919

In the Table 1, RetinaFace had high detection precision in terms of face detection. Based on this, the paper author improved the face detection model in an innovative way to apply into the key body point detection of mice. The details were as follows. (1) Parameters of key face points in the recession of RetinaFace was modified to change 5 key network points to 3 points. The three points were separately used to show head information, body information and tail information of mice, and 3D analysis on key face points was removed, which improved training speed of the RetinaFace. (2) The data set of mice was recorded, and lots of manual annotation for data was used for training and test of model network.

3.2 Improved RetinaFace network structure

RetinaFace is a robust single-stage face detection algorithm, makes use of extra supervision and self-monitoring methods to position the pixel-level for different sizes of faces, which can achieve excellent detection effects. The network structure of the algorithm was shown in Fig. 2, which integrated feature pyramid network, context network and task association. RetinaFace used two backbone feature extraction networks in the practical training: MobilenetV1-0.25 and Resnet. Resnet can realize higher precision, and MobilenetV1-0.25 features with higher speed and can be run on CPU. In the paper, Resnet50 was selected as backbone network[20]. In the network, after 7*7 convolution (C1) for the input, four ResidualBlocks were included and were separately indicated as C2, C3, C4 and C5. Then C3~C5 output was taken as PEN (Feature Pyramid Networks) structure input. In the FPN structure layer, the system made use of 1*1 convolution to adjust number of channels for three effective features, and Upsampel and Add were integrated to complete FPN construction. Finally, three effective feature layers of P3, P4 and P5 were obtained. To further strengthen feature extraction, the paper added three SSH (Single Stage Headless) modules of parallel structure. The first parallel structure was 3*3 convolution. In the second parallel structure, two 3*3 convolutions replaced the 5*5 convolution. In the third parallel structure, 3*3 convolutions replaced the 7*7 convolution. Through the operation, reception field action scope was increased and robust context semantic segmentation was enhanced. Finally, tree effective feature layers of SSH1, SSH2, SSH3, and prediction results were obtained. (1) Classification prediction results of targets; (2) Recession prediction results of target boxes; (3) Recession prediction of key target points

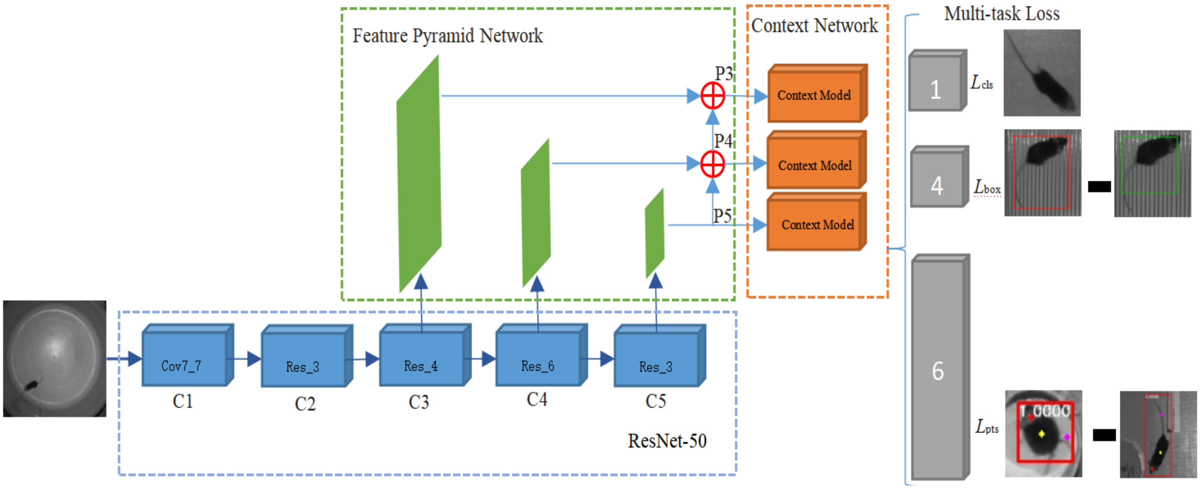
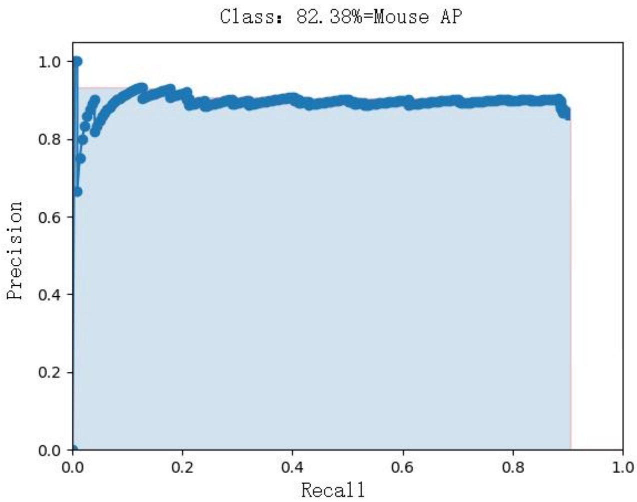


Fig. 2 Improved RetinaFace network structure

3.3 Model training

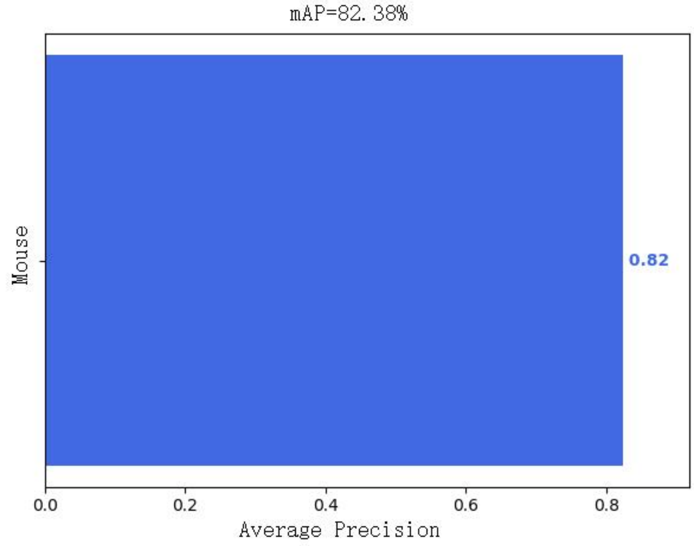
In the paper, selected deep learning framework was Pytorch and GPU acceleration tool was CUDA 11.1. Hardware configuration was as follows: CPU of Intel(R) Core(TM) i7-9700H@2.6GHz, GPU of NVDIA RTX 2060@8 GB, and RAM of 32 GB. For the backbone network, ResNet50 was selected for pre-training. For the pre-training way, Stochastic Gradient Descent (SGD) optimization model was used for network training with momentum of 0.9, weight decay of 0.0005 and batch of 50*2400. The learning rate started from 10-3, learning rate decay factor 1 was 40, and decay factor 2 was 50. When every learning rate decay was 1/10 of last epoch after first 40 epochs of the network update, whole training process ended after 50 epochs.

In the process of model training, there was loss of printing category, box, and point. The comprehensive loss is obtained by multiplying the category loss by the weight and adding the loss of boxes and points. The test set detection was used to draw PR curves and Mean Average Precision (mAP) curve to judge model training conditions. As shown in Fig.3, mAP value of neural network model constructed after training in the paper was 82.38%.



(a)

ISSN: 0010-8189



(b)

Fig.3 Training curve (a)PR curve; (b)mAP curve

3.4 Model acceleration

After getting higher-precision model, the problem – model acceleration was required to be solved, thus meeting high-speed tracking identification of mice. Therefore, in the paper, NVIDIA TensorRt was used for acceleration. Before acceleration, the model was required to transform the model to onnx model file (Open Neural Network Exchange) to support arbitrary neural network framework. Then, transformed onnx model was simplified to combine some unsupported operators. Finally, trtexec in TensorRt was used to re-build TensorRt for acceleration. Specific procedures are as follows: (1) Parse parameters in the model, including every weight and bias term in the neural network. (2) Create TensorRt engine, make use of create Engine provided by NVIDIA for Model serialization, and produce TensorRt engine. (3) In the process test, use doInference for the benchmark test, and obtain model optimization effect through testing forward reasoning time.

According to above procedures, in the paper, data of mice were tested 50 times. Before TensorRt acceleration, average forward reasoning time of the model was 104ms. After TensorRt optimization and acceleration, average forward reasoning time was reduced to 44mms, and was increased by 2.3 times.

Ⅳ.SOFTWARE DESIGN AND OPTIMIZATION

4.1 Animal behavior software design

As shown in Fig. 4, the system software design was divided into three parts: Model training, model deployment and user application program.

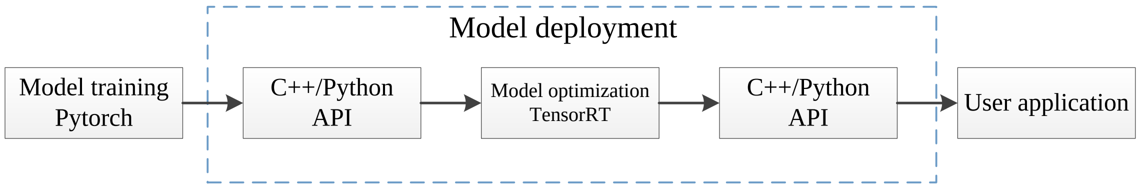


Fig. 4 Software design scheme of animal behavior system

The multi-language program model was used in the software to improve development efficiency. For example, in

ISSN: 0010-8189

the model deployment algorithm was used C++, and featured with fast speed and high execution efficiency, and TensorRt C++ API was used to complete acceleration and reasoning of the model. In the Interface UI, LabVIEW graphical programming language design was used, which greatly improved development efficiency. The communication between user interface and basic algorithm was realized through calling DLL (Dynamic Link Library). For details, refer to Fig. 5. Firstly, C/C++ was used to edit basic GPU codes, DLL was compiled and generated, LabVIEW function node was used to call DLL, corresponding name and data type were matched with parameter types defined in CUDA. After deployment, output results were showed.

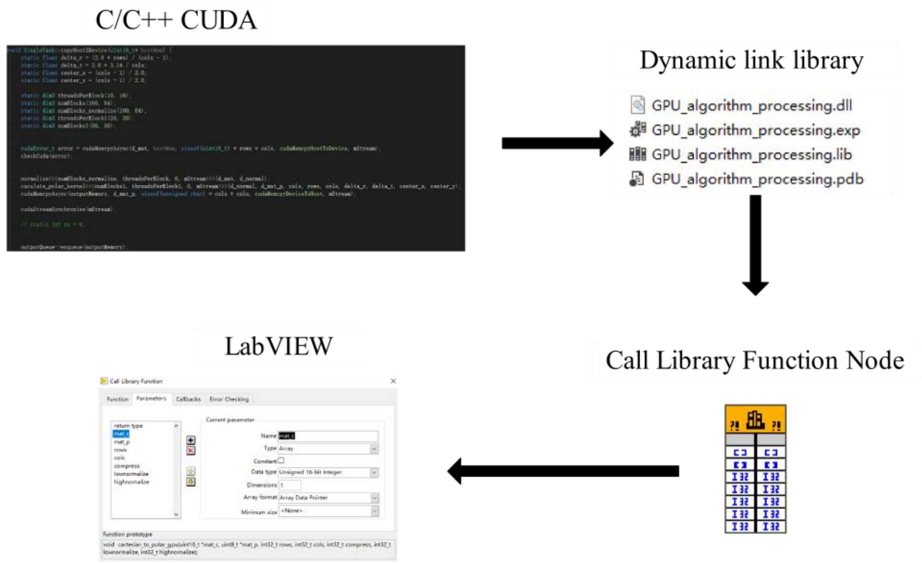


Fig.5 LabVIEW call DLL process

Based on above framework, develop rate target detection DLL: (1) MOUSE_AlgCreate (memory application) (2) MOUSE_Forword (target detection) (3) MOUSE_AlgFree (memory release) During program execution, MOUSE_AlgCreate was firstly called to load trained model file, memory space was applied for target detection program, and the initialization tasks of network model were completed. Then, MOUSE_Forword was executed. In the forward reasoning process of the target detection, upper left corner and lower right corner of the target box, target head, body and tail positions, quantity of targets, and confidence were output as results. After program execution, memory was released via MOUSE_AlgFree to prevent memory leakage.

After target detection and recognition, the smoothing algorithm was used to re-construct trajectory of mice. Meanwhile, head, body and tail information detected were used for online analysis such as holding time analysis, movement speed analysis and turning analysis.

4.2 GPU acceleration and test

In the system, after optimization on TensorRt, model reasoning speed was increased to 44ms from 104ms, and the corresponding frame rate was 22 frame/second. However, under some active states, real-time detection of mice was generally not less than 30 frame/second. Therefore, in the paper, parallel streaming features and programmability of the GPU were used, parallel treatment methods based on GPU image was put forward to solve the time-consuming problem caused by traditional CPU multithread processing. As shown in Fig. 6, GPU asynchronous flow model was used to create two data flows. One data flow was responsible for transmission and one data flow was responsible for processing. When data was copied in the CPU memory, first in-first out principle was used to avoid data loss. Meanwhile, to avoid frequent creation and expenses due to release of GPU memory, variable memory of the initialization equipment for the class's constructor and equipment constant were used, and the destructor was used to automatically release memory space. The system only needed once GPU memory matching. After the program ends, the system automatically released memory space, which greatly saved data transfer time. Through testing, after GPU second acceleration, real-time frame rate was 34.48 frame/second, and was increased by 11.75 frame/second after single TensorRt optimization. The model reasoning speed was increased to 29ms from 44ms.

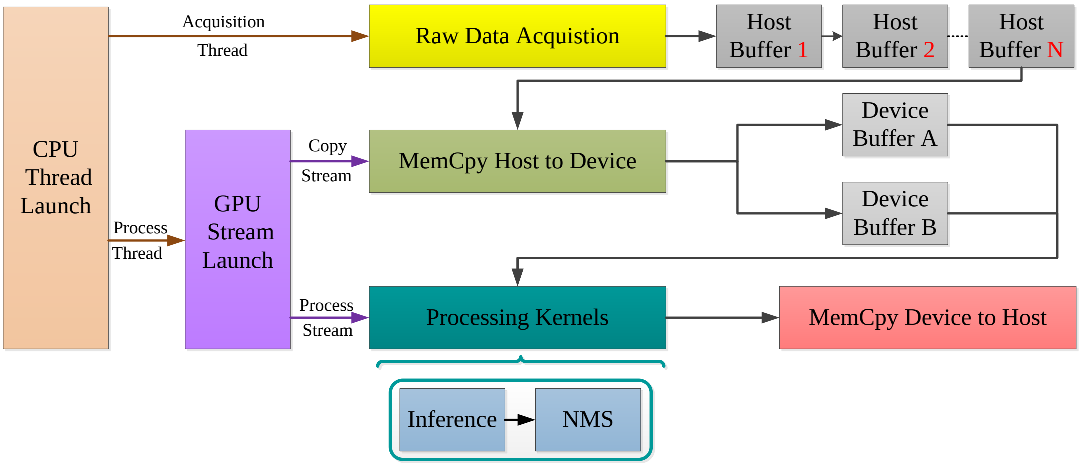


Fig.6 GPU asynchronous processing flow chart

As shown in Fig. 7, 1000-frame continuous image was used for testing, and the timeliness of non-optimization, TensorRt optimization and TensorRt+GPU optimization were tested. The results were shown below: Average reasoning speed of the model not optimized was about 104ms. After Level 1 optimization of TensorRt, reasoning time was 44ms; After Level 2 optimization of TensorRt+GPU, model reasoning time was reduced to 29ms, and was increased by 3.58 times.

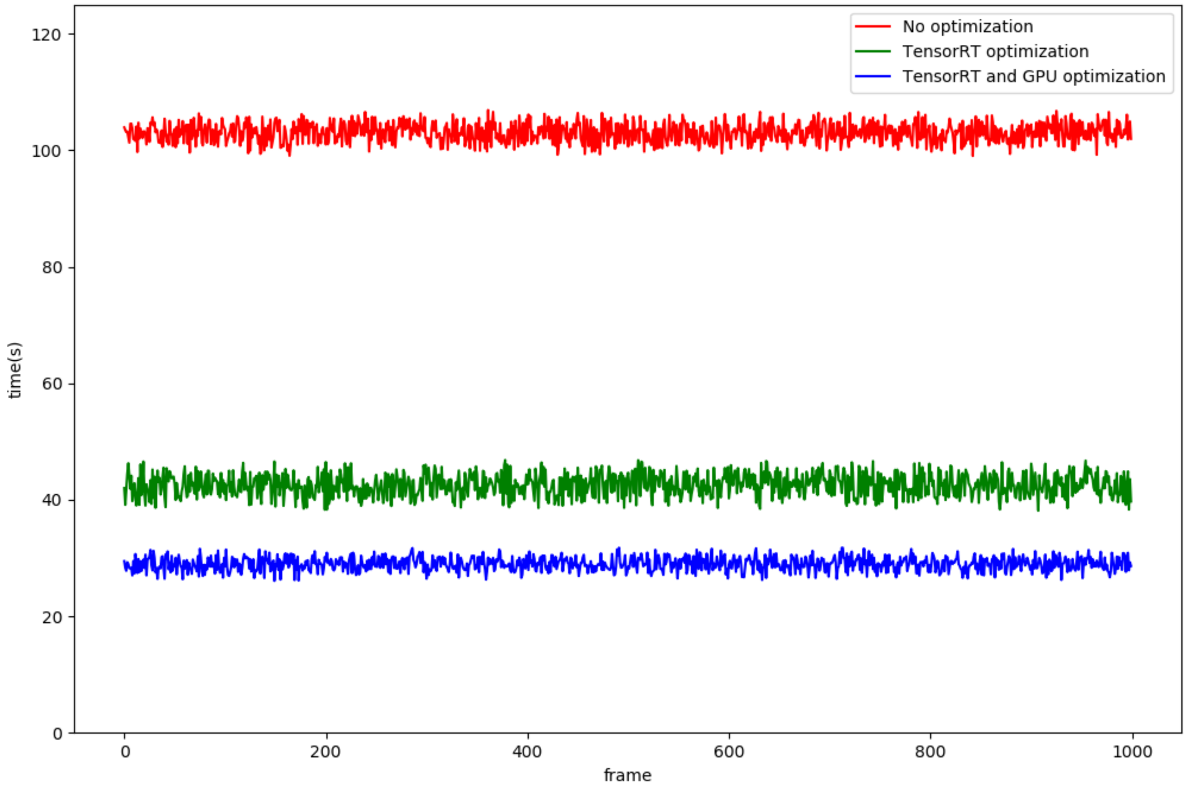


Fig.7 Timeliness comparison test

Ⅴ.EXPERIMENT RESULTS AND ANALYSIS

5.1 Experiment environment setup

To test the detect effect of the system, typical water maze experiment was selected for test. In the water maze, water escape was used as one way to stimulate learning and memory[21]. There was wide application in the aspects of aging evaluation, experimental lesion, and drug effect (especially in the rodents). The experiment process of "visual spatial navigation" for mice in the water maze environment was approved by Ethics Committee on Laboratory Animals for Suzhou Institute of Biomedical Engineering and Technology, Chinese Academy of Sciences.

As shown in Fig. 8, experiment water maze was composed of one large round pool with diameter of 120cm, height of 40 cm and water depth of 30cm. The pool temperature was kept at room temperature or a little higher than the temperature. Then titanium dioxide was added to form opaque water, a transparent round organic glass platform was placed in the pool, and fixed platform hided under the water (1.0 cm under the water). Platform and pool wall were white. Before the experiment, the pool was mixed until water became muddy. To prevent rate from clearly seeing platform under the water, square, triangle and rhombus black patterns were pasted on the pool wall. Therefore, mice can navigate in the maze through these visual prompts[22].

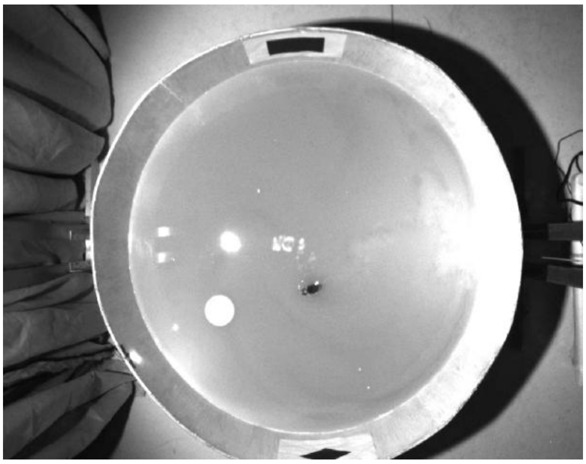


Fig.8 Physical drawing of water maze

The experiment was carried out in the totally dark place. In the experiment, three 8-week C57/B6 male mice were selected, and three mice were selected at random for water maze experiment. Through the animal behavior system software, water maze was divided into four quadrants. As shown in Fig. 9, the counterclockwise division from top left corner was first quadrant, second first quadrant, third first quadrant and fourth first quadrant. In the process of the experiment, the software monitored the movement of mice in a real-time way and analyzed trajectory information and time information for mice to explore the hidden platform. Based on different quadrants in the scene, the software analyzed displacement and time in every quadrant. Compared to 6-day experiment results, cognitive and learning abilities were evaluated[23].

ISSN: 0010-8189

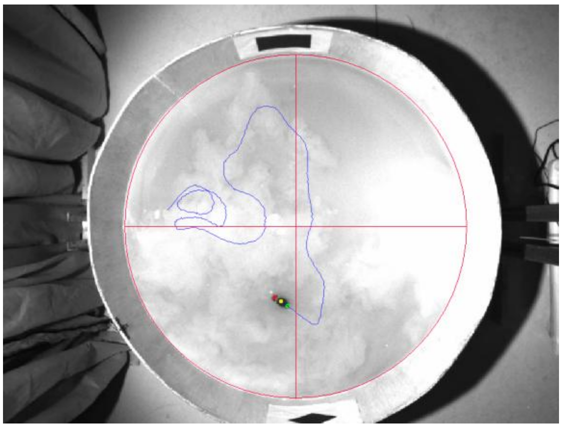


Fig.9 Quadrant division of water maze

5.2 Experiment result analysis

Through the animal behavior software, online monitoring and analysis on No. 2 mice for 5 consecutive days were carried out. After the experiment ended, software automatically generated and results were saved, as shown in Fig. 10. At the first day of the experiment, mice took long trajectory to seek for hidden platform. After training for 5 consecutive days, mice reduced trajectory to find the platform. At the fifth day of the experiment, mice used short time to find hidden platform. After training of many days for mice, through prompt information on the swimming pool wall, the mice learnt how to rapidly find the hidden platform.

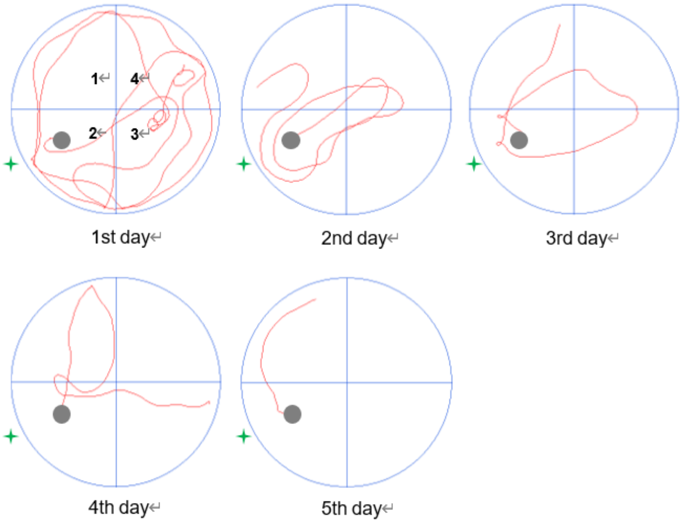


Fig.10 Trajectory analysis of experimental mice on day 1-5

Through further analysis, exploration time for 3 mice in the water maze was obtained, and average value of exploration time for 3 mice was calculated, as shown in Fig 11. It took long time for 3 mice to explore the platform. With the increase of training times and time, the time for mice to explore the hidden platform was reduced, which verified that the system was well applicable.

ISSN: 0010-8189

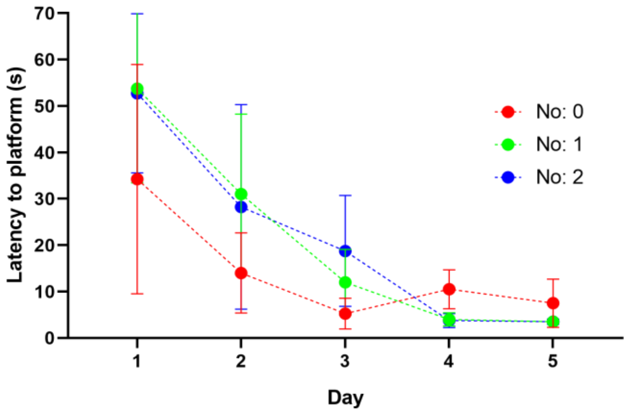


Fig.11 Analysis of exploration time in different mice

In addition, the time for 3 mice to explore the platform was averaged, as shown in Fig 12. At the fourth day and fifth day of the experiment, there was no change for platform exploration time of mice, which shown that the platform exploration time for mice was at stable state.

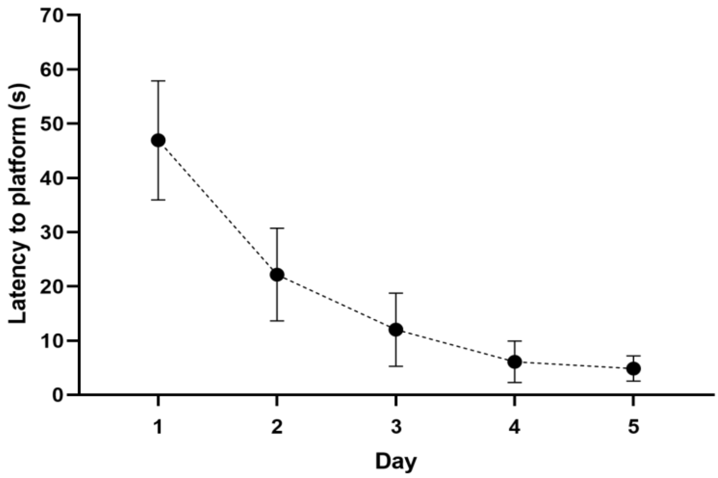


Fig.12 Exploration time analysis of mice with different training days

To further verify the spatial memory of mice, at the sixth day of the experiment, hidden platform in the water was removed, and 3 mice which were trained for 5 days were put in the fourth quadrant. Water maze exploration time for mice should be 30 s. Through system software analysis, experiment results of mice at the sixth day were obtained. As shown in Fig. 13, sports trajectory information of 3 mice were indicated respectively.

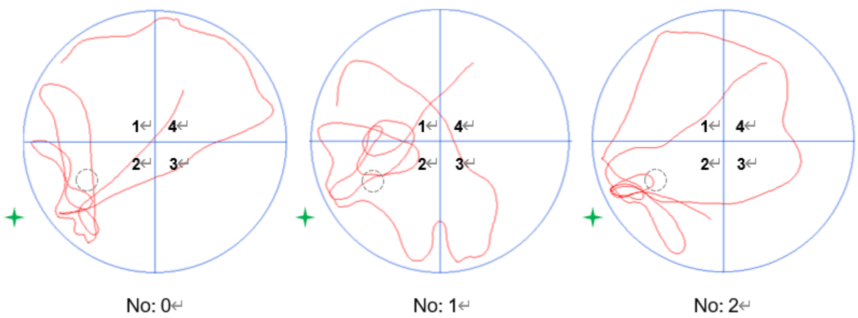


Fig.13 Trajectory analysis of experimental mice on day 6

According to above trajectory information, exploration time proportion of 3 mice in different quadrants, displacement of mice in different quadrants were calculated, as shown in Fig. 14 and Fig. 15. After mice well trained were removed from the hidden platform, exploration time and displacement of mice in the former platform in the field (second quadrant) were higher than other quadrants, which showed that mice had certain learning abilities for space exploration, and lay foundation for follow-up study on space memory navigation.

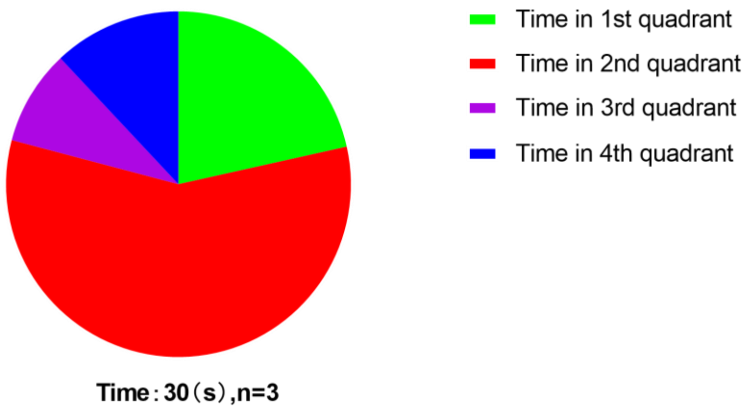


Fig.14 Time analysis of mice exploring in different quadrants

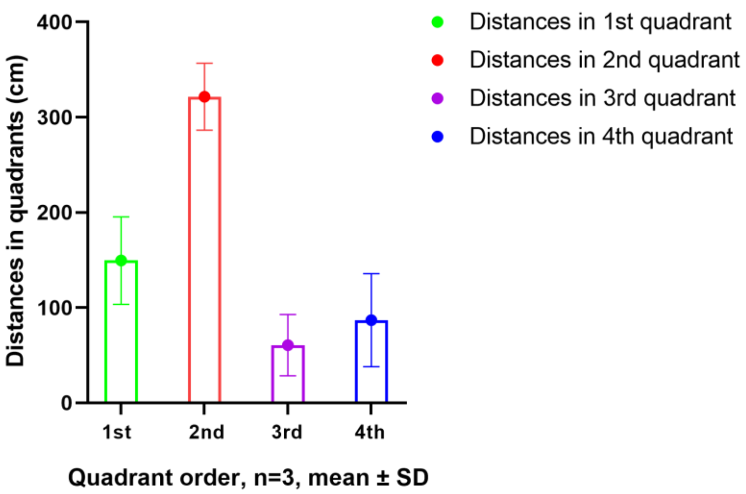


Fig.15 Analysis of displacement of mice in different quadrants (mean value; Mean ±standard deviation) (average value; average value ±standard deviation)

Ⅵ. CONCLUSION

To solve traditional animal behavior recognition method defects such as non-intelligence, heavy workload, and poor generalization performance, in the paper, RetinaFace neural network was introduced to the animal behavior analysis system, 2600 mice samples were made, and animal models were trained; NVIDA TensorRt was used to accelerate and re-construct the engine for the model and optimize through GPU multi-frame parallel calculation way and increase the frame by 11.75 frame/second. Experiment results showed that: The mAP value of the neural network model constructed was 82.38%. After many optimizations, average forward reasoning time of the single image was 29ms and was increased by 3.58 times; The complete system can recognize and track the water maze environment and realize the real-time analysis on key behavior information of mice (such as trajectory, displacement, and exploration time).

ACKNOWLEDGEMENTS

The authors are grateful to Dr. Yuguo Tang, and Dr. Hongbo Jia for very helpful discussions and comments; to Mr. Sibo Wang for help in composing and layout editing of the figures. This work was supported by Scientific Instrument Developing Project of Chinese Academy of Sciences, Grant No.GJJSTD20190003, the Scientific Instrument Developing Project of the Chinese Academy of Sciences, Grant NO.YJKYYQ20200052.

REFERENCES

[1] Mathilde A, Haas S S, Amirhossein M, et al. Personalized Estimates of Brain Structural Variability in Individuals With Early Psychosis. Schizophrenia Bulletin, 2021, 47(4):1029-1038.

[2] Lalousis P A,Wood S J,Schmaal L, et al. Heterogeneity and Classification of Recent Onset Psychosis and Depression: A Multimodal Machine Learning Approach. Schizophrenia Bulletin, 2021, 47(4):1130-1140.

[3] Patel R N, Khil V, Abdurahmonova L, et al. Mantis shrimp identify an object by its shape rather than its color during visual recognition.The Journal of experimental biology,2021,224(8).

[4] Wutke M, Schmitt A O, Traulsen I, et al. Investigation of Pig Activity Based on Video Data and Semi-Supervised NeuralNetworks. AgriEngineering,2020,2(4):

[5] Zexin CH, Ruihan Z, Yu Eva Z, et al. AlphaTracker: A Multi-Animal Tracking and Behavioral Analysis Tool. bioRxiv - Animal Behavior and Cognition,2020,doi: https://doi.org/10.1101/ 2020.12.04.405159.

[6] Xinyao W, Wenbo W, Yezhong T,et al. Apparatus and methods for mouse behavior recognition on foot contact features. Knowledge-Based Systems,2021(prepublish):

[7] Song W, Hengyi Z. Automatic analysis system of motion in rat based on vision. Transducer and Microsystem Technologies，2011,30(11):102-104.

[8] Kai S. 2019. “Automatic analysis system of mouse behavior”, Harbin Institute of Technology.

[9] Giancardo L, Sona D, Huang H, et al. Automatic visual tracking and social behavior analysis with multiple mice. Plos One, 2013, 8(9):1-14.

[10] Lorbach M, Kyriakou E I, Poppe R, et al. Learning to recognize rat social behavior: Novel dataset and cross-dataset application. Journal of Neuroscience Methods, 2018, 300(4):166-172.

[11] Lh A, Mm B, Bw C, et al. Towards image-based animal tracking in natural environments using a freely moving camera. Journal of Neuroscience Methods, 2020, 330(15):1-8.

[12] Zexin C, Ruihan Z, Yu Z, et al. Alpha Tracker: A Multi-Animal Tracking and Behavioral Analysis Tool. bioRxiv-Animal Behavior and Cognition ,2020, doi: https://doi.org/10.1101/ 2020.12.04.405159.

[13] Noldus L P, Spink A J, Tegelenbosch R A. EthoVision: a versatile video tracking system for automation of behavioral experiments. Behavior Research Methods, Instruments, & Computers, 2001, 33(3):398-414.

[14] Serdar Y. Nvidia's new TensorRT speeds machine learning predictions. InfoWorld.com, 2017.

[15] Zhu C, Tao R, Luu K, et al. Seeing Small Faces from Robust Anchor's Perspective//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2018.

[16] Li J, Wang Y, Wang C, et al. DSFD: Dual Shot Face Detector// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020.

[17] Zhang S, Wen L, Shi H, et al. Single-Shot Scale-Aware Network for Real-Time Face Detection. International Journal of Computer Vision, 2019, 127:537-559.

[18] Deng J, Guo J, Ververas E, et al. RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild// 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020.

[19] Yang S, Luo P, Loy C C, et al. WIDER FACE: A Face Detection Benchmark// IEEE Conference on Computer Vision & Pattern Recognition. IEEE, 2016:5525-5533.

[20] Akiba T, Suzuki S, Fukuda K. Extremely Large Minibatch SGD: Training ResNet-50 on ImageNet in 15 Minutes. arXiv:1711.04325v1 [cs.DC] 12 Nov 2017.

[21] Zorin A, Chernyuk D, Vlasova O, et al. Software for analyzing the behavioural test "Morris Water Maze". E3S Web of Conferences, 2020, 203(1):01029.

ISSN: 0010-8189

[22] Young M E, Hoane M R.Mixed effects modeling of Morris water maze data revisited: Bayesian censored regression. Learning & Behavior, 2021:1-14.

[23] Mariana B N, Daniella B V, Marcia J G, et al. Effects of fish oil supplementation on spatial memory in rats with pilocarpine-induced epilepsy assessed using the Morris Water Maze test. Epileptic Disorders, 2021, 23(3).

ISSN: 0010-8189