Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 12, No. 1, 2025

Design and Optimization of Urban Stray Animal Supervision Model based on YOLOv8 and Flink

Bo Rao, Junjie Feng, Chengxin Huang, Junming Chang *

Jianghan University College of Artificial Intelligence, Wuhan, Hubei, China
* **Corresponding author**: Junming Chang

Abstract: With the increasingly serious problem of stray animals in cities, traditional manual monitoring and management methods are unable to meet the real-time and efficient monitoring needs of stray animals. Therefore, this article designs a city stray animal supervision and testing model based on YOLOv8 and Flink. This model identifies stray animals and determines their species based on the video streams captured by the camera. It simultaneously uses the real-time processing framework Flink to sort and dynamically analyze data by time, thus providing accurate data for the management and protection of stray animals. Through training and parameter optimization of the model, we achieved high accuracy and real-time performance in the task of detecting stray animals. The experimental results show that the mAP value of this model in stray animal detection reaches 0.767, the F1 Score is 0.75, and the inference speed meets the requirements of real-time monitoring. The design and implementation of the model provide effective technical support for the management and protection of urban stray animals, and have high application value and practical significance.

Keywords: YOLOv8; Flink; Object Detection; Animal Identification.

1. Introduction

In the modern urban environment, the number of stray animals is increasing, and their activities not only affect the urban environmental health, such as public excretion, rummaging, etc., but also may carry germs and parasites, posing a threat to public health. In addition, the disorderly reproduction and activities of stray animals may disturb the balance of urban ecosystems[1]. To solve these problems, scholars at home and abroad have proposed a variety of management schemes, such as manual inspection and capture, electronic tag tracking and so on.

However, these methods generally have limitations: manual inspection cost is high and efficiency is low; Electronic tag tracking depends on individual wearing devices, and it is difficult to promote it on a large scale[2]. In recent years, the development of computer vision and deep learning technologies have provided new possibilities for stray animal monitoring[3]. Based on YOLOv8 target detection method and Flink streaming data processing framework, an efficient and intelligent stray animal supervision model was constructed in this study. In the detection stage, the model relies on the target detection algorithm to automatically identify and classify stray animals, and the data processing part adopts the high-throughput and low-latency streaming data computing framework to realize the real-time update and storage of the detection results. Through this technological integration, the model can not only accurately identify the types of stray animals, accurately distinguish individuals, and establish "identity files" for them[4,5], but also track their activity trajectories in real time, achieve fine management, and provide data support for the rescue, adoption and quantity control of stray animals. It also provides a new method for improving the efficiency of urban stray animal management.

2. Research and optimization of image recognition model based on YOLOv8 algorithm

2.1. Model Algorithm

The YOLOv8 algorithm, as the current new generation of highly efficient object detection model, still inherits the design philosophy of the YOLO series in its core concept, namely "You Only Look Once" (one-pass prediction). This means that when the model conducts object detection, it can complete the detection and classification of objects in an image through only one forward propagation. Compared with traditional multi-step detection algorithms, the characteristic of "one-pass prediction" greatly improves the detection efficiency, and at the same time maintains a good accuracy rate, which is particularly important for real-time or situations requiring a quick response[6]. In this study, we trained a model with the function of animal recognition based on this algorithm. We used the stray animal image dataset collected by the data collection layer for model training. After the model training was completed, we deployed the model on the server.

2.2. Algorithm Flow

In the algorithm, the CNN network divides the input picture into S×S grids, and each cell predicts the target, and then each cell cloud is processed by image positioning and positioning algorithm. Each cell defines a label of a multidimensional vector, and then observes the cell center position of the target border.

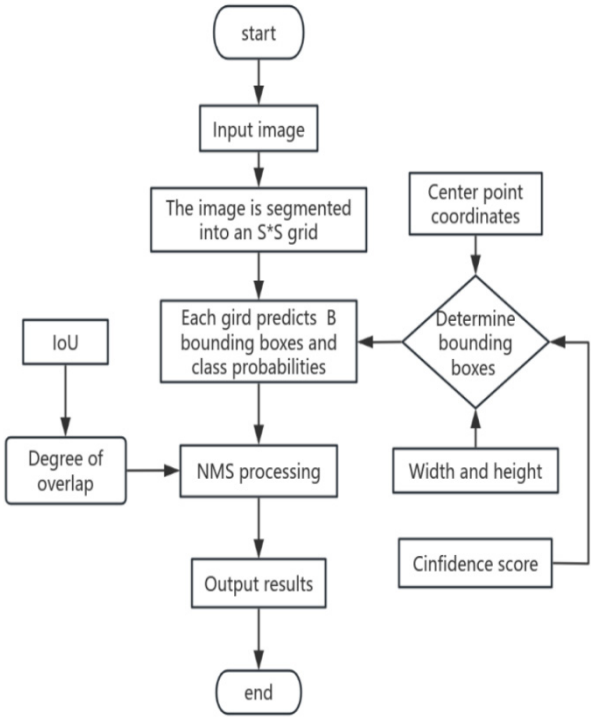

Figure 1. Algorithm Processing Data Flow Chart

Each grid will predict B bounding boxes and the confidence score of the bounding boxes.

The parameters of the border are as follows:

Coordinate of the center point: (x,y), offset from the upper left corner of the grid.

Width and height: (w,h), the ratio relative to the width and height of the image.

Confidence: confidence=Pr(Object)×IoU, which indicates the probability that there is an object in the grid and the accuracy of the bounding box.

For each bounding box, the model also predicts the probability of c categories, indicating the category of the target in the bounding box. The category probability and the confidence of the bounding box are calculated independently. For each cell, it also gives the probability values of predicting C categories, which represents the probability that the target of the bounding box predicted by the cell belongs to each category, but these probability values are actually the conditional probability under the confidence level of each bounding box, that is, PR (classic). At the same time, we can calculate the confidence of each bounding box category, which represents the possibility that the target in the bounding box belongs to each category and the quality of the bounding box matching the target.

Finally, the output is processed and Non-Maximum Suppression(NMS) is used.NMS ranks the bounding boxes according to their confidence, and removes other bounding boxes with high overlap with high confidence bounding boxes. Intersection over Union(IoU) is usually used to measure the degree of overlap, and the IoU threshold is usually set to 0.4 or 0.5. NMS processing can solve the problem that an object in a picture is detected many times. First, the frame with the greatest confidence is found out from all the detection frames, and then the IoU between it and the remaining frames is calculated one by one. If its value is greater than a certain threshold (the overlap is too high), then the frame is eliminated. Then repeat the above process for the remaining detection frames until all detection frames are processed[7,8].

3. Research on Data Processing Model Based on Flink

In this article, building a processing pipeline using Flink's DataStream API is at the core of data processing. This process comprehensively covers a series of steps from data collection and processing to storage, and effectively manages and updates the information of stray animals through interaction with the MySQL databas[7,8]. The specific process is shown in Figure 2.

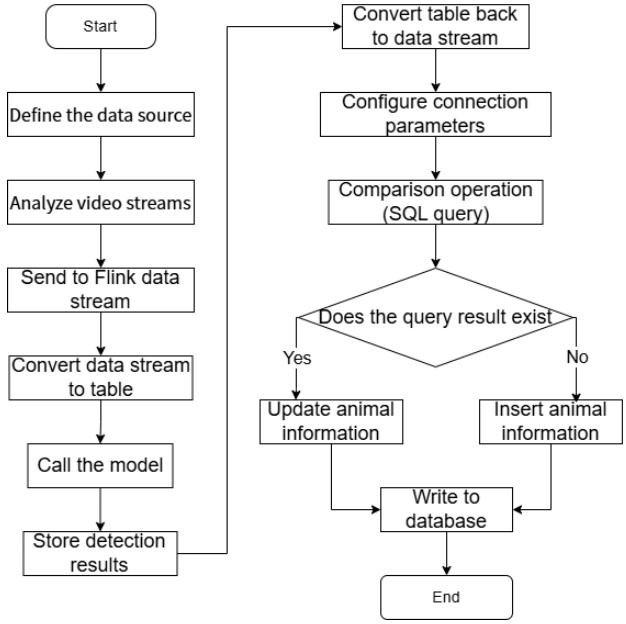

Figure 2. Flink Data Processing Flow Chart

3.1. Data Source Collection

The collection of data source is the starting point of the whole data processing process, which is responsible for receiving video streaming data from the camera and ensuring that the data can be efficiently entered into the streaming computing system. Under the framework of Flink, data sources are constructed through streaming data acquisition module to support long-term connection and stable data transmission of camera devices[9-13].

Based on the streaming data processing mechanism, we carry out structured transformation of the captured video stream to ensure the efficient execution of subsequent calculations. Before entering the processing pipeline, the data flow is optimized by time stamp marking, format conversion, frame rate control, etc., in order to meet the computational requirements of real-time target detection. At the same time, the data source management mechanism can also ensure that the collection of data streams can be quickly resumed in the case of sudden interruption or abnormal circumstances, and ensure the continuity and stability of stray animal monitoring.

3.2. Data and Video Stream Processing

After the video stream enters the Flink processing pipeline, it is necessary to complete the image frame analysis to meet the input requirements of target detection. Therefore, we use the data stream conversion module to structure the video stream, extract effective image information, and optimize the data format to adapt to the efficient calculation process. In the parsing process, the system performs data decoding, frame

extraction, format conversion and other operations, so that the video frame has a standardized structure suitable for subsequent detection. Aiming at the real-time requirement, the streaming frame processing strategy ensures the low delay of data conversion and provides stable high-quality image input for the target detection module.

3.3. Model Invocation and Detection

After completing the video frame parsing, enter the stray animal detection stage, which uses Python UDF (user-defined function) to call the object detection model. During the detection process, the video data is converted into a table structure to facilitate streaming computation using SQL like syntax and ensure that the data enters the detection module after structured processing.

The detection module extracts features and recognizes objects from input image frames, and combines spatial location, morphological features, and other information to determine the type, location, and confidence of stray animals. The recognition results are filtered and processed to remove low confidence targets, optimize bounding box information, and store the final results in a newly created data table. Subsequently, the data table was converted into a reflow data format to provide support for subsequent data storage, statistical analysis, and application display.

4. Results and Analysis

4.1. Environment Configuration

The experiment is based on Python 3.10 and Pytorch 1.13 frameworks, running on an AMD Ryzen 7 5800H @3.20GHz *16 core processor, and training using an NVIDIA RTX 3050 graphics card with 16GB of video memory.

4.2. Evaluation Indicators

In order to comprehensively evaluate the performance of the detection model, the following performance evaluation indicators are selected:

Accuracy: It measures the correctness of the overall prediction of the model and reflects its ability to distinguish between targets and non-targets. The formula is shown in Eq. (1).

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}. \quad (1)$$

Among them:

TP(True Positives):

The number of stray animals correctly detected.

TN(True Negatives):

The number of categories correctly identified as non-targets.

FP(False Positives):

The number of non-target animals that are wrongly detected as stray animals.

FN(False Negatives):

The actual number of stray animals that have not been identified.

- Precision: Measure the proportion of stray animals among the detected targets, and reflect the false alarm of the model. The formula is shown in Eq. (2).

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}. \quad (2)$$

- Recall:measures the proportion of all real stray animals successfully detected by the model, and reflects the missed detection of the model. The formula is shown in Eq. (3).

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}. \quad (3)$$

Mean Average Precision (mAP): Calculate the accuracy under multiple IoU thresholds respectively, and take the average value.

mAP 50: Average accuracy when IoU threshold is 0.5.

mAP 50-95: The IoU threshold is from 0.5 to 0.95, and the average accuracy is 0.05.

4.3. Model Adjustment and Optimization

During the training process, the performance of our model is improved by optimizing and adjusting hyperparameters (mean average precision (mAP), F1 score) to ensure accurate animal recognition. The specific parameters of the model are shown in Table 1.

Table 1. Model Parameters

Hyperparameter	Setting	Instructions
Learning rate (lr0)	0.01	The step size that determines the weight adjustment of the model is helpful for rapid convergence in the early stages of training
Decline in learning rate (lrf)	0.01	Controlling the rate of decrease in learning during the training process helps the model to make detailed adjustments in the later stages of training
momentum	0.937	Accelerate the learning of the model in the correct direction, reduce oscillations, and accelerate convergence speed
weight_decay	0.0005	Prevent overfitting by adding regularization terms to the loss function to reduce model complexity
warmup_epochs	3.0	Start training at a lower learning rate in the initial few cycles and gradually increase to the predetermined learning rate
batch	16	The number of samples input into the model during each iteration of training affects GPU memory usage and model performance

mAP comprehensively evaluated the accuracy and comprehensiveness of the model in identifying stray animals by calculating the average accuracy under different recall rates. In the scenario where this model is applied to stray animal detection, the mAP value reaches 0.767, indicating that this model has high accuracy and reliability in stray animal detection tasks.

The F1-Score is a harmonic average of accuracy and recall, which balances the two metrics, avoiding the situation of focusing on one and ignoring the other. The F1-Score of this

model reaches 0.75, which indicates that it can not only accurately identify stray animals (high accuracy), but also find most of the actual stray animals (high recall rate).

In order to further evaluate the model performance, this paper conducted a quantitative analysis of the training results, as shown in Figure 3.

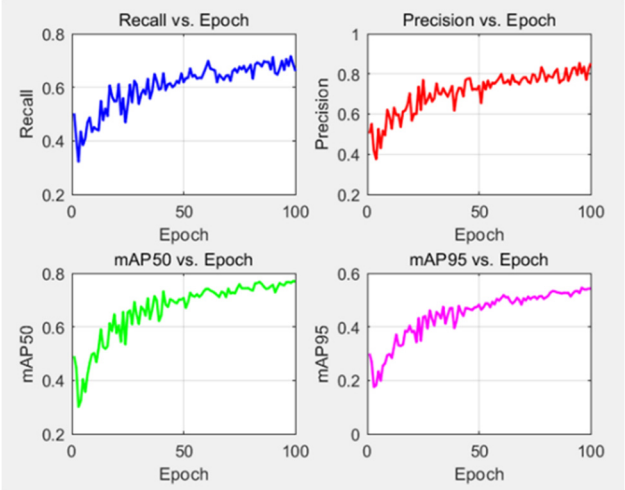

Figure 3. Model parameter diagram

It can be seen that the recall rate and accuracy rate reached 75% and 85% respectively, indicating that this model can capture the target well and reduce false positives in the detection of stray animals. In addition, on mAP@0.5 (mean accuracy @0.5), its value is close to 0.8, indicating that the model has higher detection accuracy under a low IoU threshold. More significantly, on the more stringent index mAP@0.95 (average accuracy @0.95), the model also performs significantly better, finally approaching 0.75, showing its advantages in high-precision target detection.

In general, the model showed high accuracy and stability in the task of stray animal detection, and the performance of various indicators proved its practicability in real-time monitoring function, providing reliable technical support for urban stray animal management.

5. Conclusion

This article successfully designed and implemented an urban stray animal monitoring model based on YOLOv8 and Flink, effectively filling many shortcomings of traditional stray animal management methods. By reasonably deploying cameras in areas where stray animals often appear, utilizing the efficient object detection capability of this model and Flink's powerful real-time data processing function, accurate identification and real-time information recording of stray animals have been achieved, and the data has been stored in an orderly manner in a MySQL database. In future research, further exploration can be conducted to optimize models and algorithms, improve their performance in all aspects, expand their application scenarios, and contribute more to the balance of urban ecology and public health security.

Acknowledgments

This paper is sponsored by the 2023 Student Research Project in Jianghan University (An Integrated Supervision Framework for Urban Stray Animals YOLOv8-Based Detection and Flink-Enabled Real-Time Analytics).

References

[1] Yan, N. (2020). The security hazards of urban stray animals and management suggestions. China Animal Quarantine, 37(07), 49-52.

[2] Chen, M. Y. (2016). On the ethical review and management of urban stray animals. Management Observation, (29), 75-77+80.

[3] Zhang, X. W., Wang, Y. M., Li, Z. J., et al. (2023). A review of animal identity recognition methods. Heilongjiang Animal Husbandry and Veterinary Medicine, (08), 34-42+134-135.

[4] Wu, G. B., & Pan, Y. Y. (2024). Design of a scenic spot population monitoring system based on YOLO and Flink. Electronic Technology, 53(10), 53-55.

[5] Zhang, L. J., Yao, G. P., & Wu, Z. Z. (2024). Design and implementation of a classroom attendance system based on facial recognition. Wireless Internet Technology, 21(22), 23-27.

[6] Jiang, X. X. (2024). Application research of facial recognition access control system in smart community. Television Technology, 48(10), 225-228.

[7] Chen, F. (2024). Design research of a data entry system based on MySQL database. Science and Technology Information, 22(20), 35-37.

[8] Liu, X., & Ji, Y. K. (2023). Design and implementation of an electronic medical record data processing model based on Flink. Wireless Internet Technology, 20(16), 52-56+66.

[9] Fan, X. H. (2020). An industrial big data storage and analysis system based on Hadoop. Science and Technology Innovation and Application, (23), 18-20+24.

[10] Mao, S. H., & Wang, W. D. (2024). A review of YOLO series object detection algorithms based on deep learning. Journal of Yan'an University (Natural Science Edition), 43(02), 88-95.

[11] Ma, P., Yang, Z. H., Wan, H., et al. (2023). A cotton aphid image recognition algorithm and software system design based on YOLOv8 network. Journal of Intelligent Agricultural Equipment (Chinese and English), 4(03), 42-49.

[12] Mao, S. H., & Wang, W. D. (2023). The impact of the depth and width of convolutional neural networks on the performance of cat and dog image recognition models. Henan Science, 41(07), 956-963.

[13] Shao, Y. H., Zhang, D., Chu, H. Y., et al. (2022). A review of YOLO object detection based on deep learning. Journal of Electronics and Information, 44(10)