Automatic sheep counting method and experimental study based on mask R-CNN

Yikun Fan,[1,2] Meian Li,[1,2] Ting Li,[1,2] Hao Lian,[1,2] HuiLin Jiang,[1,2] Wenqian Yang,[1,2] Peng Zhou[1,2]

[1]Computer and Information Engineering College, Inner Mongolia Agricultural University, Hohhot, Inner Mongolia Autonomous Region, China; [2]Inner Mongolia Autonomous Region Key Laboratory of Big Data Research and Application of Agriculture and Animal Husbandry, Hohhot, Inner Mongolia Autonomous Region, China

Abstract

With the rapid growth of global meat demand and the expansion of livestock farming, accurate and efficient sheep counting has become crucial in modern livestock management. However, traditional manual counting methods are inefficient and lack accuracy, while automatic counting systems based on RFID and GPS, though more precise, are costly and challenging to scale. To address this issue, this paper proposes an automatic sheep counting method based on Mask R-CNN, aiming to enhance the accuracy and robustness of object detection using instance segmentation technology from deep learning. Mask R-CNN not only provides pixel-level precise segmentation for each sheep but also resolves misdetections and missed detections in occluded and densely populated scenes by optimizing bounding box and mask thresholds. The study was conducted using sheep image data from an actual livestock farm in Inner Mongolia, China, and tested the model under various environmental factors, such as lighting, background complexity, and sheep density. The experimental results indicate that the optimized Mask R-CNN model performs exceptionally well in diverse scenarios, achieving a counting accuracy of 96.1% and a generalization accuracy of 88.8%, significantly outperforming traditional detection models like YOLOv5M (80.3%) and SSD (83.7%). This research not only demonstrates the excellent performance of Mask R-CNN in sheep counting tasks but also provides a low-cost, efficient automated management solution for modern livestock farms.

Correspondence: Meian Li, Computer and Information Engineering College, Inner Mongolia Agricultural University, Hohhot, China.
E-mail: limeian1973@126.com

Key words: sheep counting; Mask R-CNN; instance segmentation; object detection.

Contributions: YF, ML, TL, contribution to conceptualization; HL, PZ, contribution to experiment software; WY, contribution to validation; HJ, contribution to data curation; TL, manuscript original drafting; YF, manuscript review and editing; ML, project administration, funding acquisition. All authors read and approved the final version of the manuscript and agreed to be accountable for all aspects of the work.

Conflict of interest: the author declares that there are no known competitive economic interests or personal relationships that affect the work reported in this article.

Availability of data and materials: all data generated or analyzed during this study are included in this published article.

Acknowledgements: this research was funded by the Inner Mongolia Natural Science Foundation Project "Research on Feature Point Space Invariance Method for Multi target Individual Identity Recognition of Livestock" (grant number: 2023LHMS06012) and the Basic Research Business Fee Project of Inner Mongolia Autonomous Region Directly Affiliated Universities "Research and Demonstration of Key Technologies for Multi scale Yellow River Ice Situation Automatic Detection Based on AIoT" (grant number: BR231407).

Received: 19 October 2024.
Accepted: 5 September 2025.

©Copyright: the Author(s), 2025
Licensee PAGEPress, Italy
Journal of Agricultural Engineering 2025; LVI:1614
doi:10.4081/jae.2025.1614

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Publisher's note: all claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article or claim that may be made by its manufacturer is not guaranteed or endorsed by the publisher.

Introduction

With the continuous growth of the global population and changes in dietary structure, rising demand for meat consumption has driven the rapid development of animal husbandry, especially the sheep industry. According to FAO statistics in 2022, China's mutton production accounted for 60% of the global total, with Inner Mongolia as the main production base, contributing 21% of the national output (https://www.thepaper.cn/newsDetail_forward_26574633). The intensification of modern livestock farming has raised the need for efficient management technologies, where accurate sheep counting is crucial for resource allocation, disease control, and operational efficiency. Traditional manual counting is no longer suitable for large-scale farming due to its time consumption, labor intensity, and susceptibility to human error. Tag-based automated counting methods such as RFID and GPS improve accuracy (Bridge *et al*., 2019; Wang, 2019), but are costly, complex to maintain, may cause stress to animals, and are difficult for small and medium-sized farms to install and use in weak network environments, limiting their adoption. Meanwhile, computer vision has gained attention in livestock management. Traditional machine learning approaches, relying on feature extraction and segmentation algorithms (e.g., region growing, morphological processing), can achieve livestock counting (de Lima Weber *et al*., 2023), but require favorable optical conditions and high computational cost. With advances in deep learning, methods like object

detection (YOLO, Faster R-CNN), multi-object tracking (YOLO + DeepSORT), and image segmentation (Mask R-CNN) have been applied to livestock counting (Bharati and Pramanik, 2020). For example, YOLOv3 achieved 100% accuracy in cow counting, but struggles with occlusion in dense flocks (Wang and Xu, 2019); YOLO + DeepSORT reached 93% accuracy but demands high detection performance (Zhang *et al*., 2024); Mask R-CNN, with pixel-level segmentation, achieved 96% accuracy in drone images of cattle (Xu *et al*., 2024). However, most studies focus on lab settings, lacking validation in real farming environments, and current models still need improved generalization in complex scenarios (Dac *et al*., 2022). Therefore, this study applies Mask R-CNN to sheep counting, optimizes parameters to improve accuracy, and validates its effectiveness using real farm data from Inner Mongolia, aiming to provide a low-cost, efficient, and automated solution for modern livestock management.

The main contributions of this study are:

i) A Mask R-CNN-based sheep counting method, optimized for dense scenes and occlusion challenges.
ii) A dual-threshold strategy to improve counting accuracy by filtering noisy masks.
iii) Validation using real farming data and comprehensive comparison with YOLOv5, SSD, and DeepLabv3+.
iv) Practical recommendations for low-cost deployment, supporting intelligent management in small and medium-sized farms.

Materials and Methods

This study addresses the needs of numerous small and medium-sized farms, as well as individual herders in Inner Mongolia, who require cost-effective and computationally efficient counting methods. In response to these demands and the high costs and computational requirements of existing methods, this research proposes a sheep counting model based on image segmentation within deep learning.

Considering the target sheep herds in real farming environments, the study initially compares deep learning image segmentation methods, represented by models such as FCN (Long *et al*., 2015), Unet (Ronneberger *et al*., 2015), DeepLabv3+ (Chen *et al*., 2018), Mask R-CNN (He *et al*., 2017), and PANet (Liu *et al*., 2018). After comparing these models, Mask R-CNN is selected as the base network. The model generates segmented images, sets counting rules based on the mask generation logic of Mask R-CNN, and iteratively counts the masks that meet these criteria to achieve sheep counting. The overall technical workflow is illustrated in Figure 1.

Evaluation metrics

This study investigates a sheep image segmentation algorithm based on Mask R-CNN, introducing the concept of a confusion matrix and evaluating the model using various research metrics. These include mean average precision (mAP) for both object detection and image segmentation tasks, parameter count, inference speed per frame, and model size.

The confusion matrix assesses the classification model's performance by comparing the model's predictions with the actual labels, providing four key metrics: true positive (TP), representing correctly predicted positive cases; false positive (FP), indicating negative cases incorrectly predicted as positive; true negative (TN), showing correctly predicted negative cases; and false negative (FN), which represents positive cases incorrectly predicted as negative. These metrics enable the calculation of accuracy, precision, recall, and F1-Score, providing a comprehensive evaluation of the classification model's performance.

Here are some evaluation metrics derived from the confusion matrix, along with their calculation methods:

Recall: indicates the number of correctly predicted positive samples in the dataset, calculated using the formula in Eq. (1).

$$Recall = \frac{TP}{TP+FN} \quad \text{(Eq. 1)}$$

Precision: indicates the number of true positive samples among the predicted positive samples, calculated using the formula in Eq. (2).

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

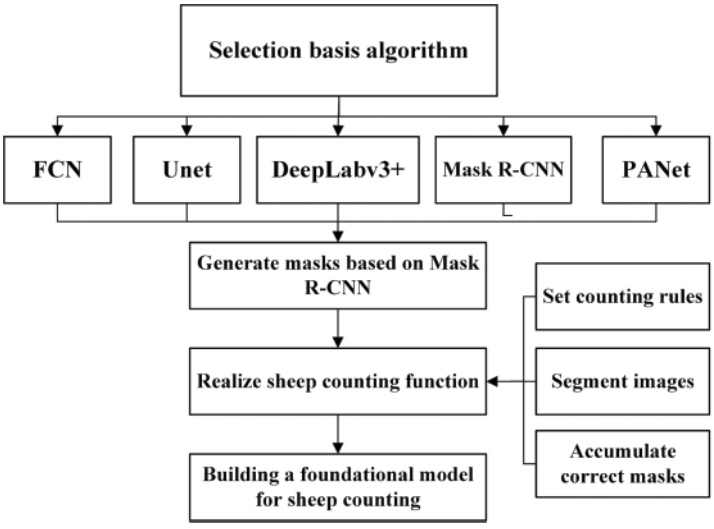

Figure 1. Technical roadmap.

In the sheep counting task, the evaluation of the model requires a comprehensive consideration of both metrics.

In object detection and instance segmentation, mAP evaluates multi-class performance. When there is only one class, mAP equals AP. As this study focuses solely on sheep, mAP = AP. Nonetheless, mAP is used to align with standard practice, enable comparisons, and support future multi-class extensions.

i) Calculate the intersection over union (IoU) values between predicted boxes and ground truth boxes, determining the counts of TP, FP, and FN based on a threshold.
ii) Sort the predicted boxes by their confidence scores and calculate precision and recall at different confidence levels based on the counts of TP, FP, and FN.
iii) Plot the Precision-Recall (P-R) curve using the multiple sets of precision and recall values obtained in step 2. The area under the curve represents AP, calculated as shown in Eq. (3).

$$AP = \int precision\ d(recall) \quad \text{(Eq. 3)}$$

iv) Take the average of all AP values to obtain mAP, calculated as shown in Equation, where N represents the total number of classes:

$$mAP = \frac{\sum_{n=0}^{n} AP_n}{N} \quad \text{(Eq. 4)}$$

Additionally, this study considers the following metrics: the number of parameters (Params), the computational complexity (Macc), the inference speed per frame, and the model size (MS). The number of parameters indicates the count of trainable parameters that need to be optimized, while computational complexity reflects the number of multiply-accumulate operations in the model. The inference speed per frame indicates the time required to infer a single image, and model size indicates the storage space occupied by the model.

Counting accuracy: the calculation method for counting accuracy is shown in Eq. (5). Count_Accuracy evaluates the accuracy of sheep counting, where total_projects refers to the number of sheep predicted by the model, annotated_count refers to the number of sheep in the ground truth annotations, and total_annotated refers to the total number of annotated data.

$$Count_{Accuracy} = \frac{\left(1 - ABS\left(\sum_{i=1}^{i=n} \frac{total_{projects} - annotated_{count}}{annotated_{count}}\right)\right)}{total_{annotated}} \times 100\% \quad \text{(Eq. 4)}$$

Dataset construction

To accurately evaluate the sheep counting method based on Mask R-CNN model, video data was acquired from real agricultural scenes in Baita village, Hohhot City, Inner Mongolia. The dataset was collected at different time periods, covering various lighting conditions, different farming densities, and complex scene environments. We used a HIKVISION MINI PTZ camera (Figure 2) mounted at a height of 3 m above the ground to capture four 1920×1080 videos from different top-down angles, to capture four video segments from different overhead angles. Next, we extracted image frames from the videos and annotated each frame with bounding boxes and instance segmentation masks for the sheep (Figure 3). The final dataset consists of 3,600 annotated images with sheep bounding boxes and instance segmentation masks, covering various lighting conditions, densities, and sheep behaviors (walking, standing, lying down). The dataset was divided into 80% training (2,880 images) and 20% testing (720 images). To improve the model's generalization ability, data augmentation techniques such as random rotations (±30°), flips, and scaling (0.8-1.2×) were applied to enhance diversity and robustness. The data were collected from a farm with 67 sheep, but due to the use of top-down camera angles, not all sheep were captured in a single image.

Constructing a sheep counting benchmark model based on Mask R-CNN

Model selection

This study employs the accumulation of segmentation masks of sheep categories to achieve sheep counting. To this end, a suitable base algorithm must be selected to generate accurate segmentation images. We selected several popular deep learning image segmentation algorithms and conducted experiments based on the

Figure 2. MINI PTZ Camera.

Figure 3. Sheep annotation schematic.

VOC public dataset to evaluate their performance across different metrics. These criteria will help us comprehensively assess the model's accuracy and resource consumption, enabling the selection of the most suitable algorithm for practical needs. The experimental results are detailed in Table 1.

According to the experimental results, PANet achieved the highest mAP at 49.5%, followed by Mask R-CNN at 47.4%, with DeepLabv3+, FCN, and U-Net at 43.8%, 41.6%, and 40.1%, respectively. The ~2% advantage of PANet over Mask R-CNN aligns with its design for enhanced multi-level segmentation, and this gap is not huge; therefore, Mask R-CNN remains a great choice.

In terms of model parameters (Params), FCN had the most parameters at 73.8M, indicating a complex model structure, but its performance did not show a corresponding advantage. In comparison, Mask R-CNN had a moderate parameter count of 23.5M, making it suitable for applications with certain resource consumption constraints. PANet and DeepLabv3+ had parameter counts of 29.4M and 39.6M, respectively, showing improvements in accuracy compared to Mask R-CNN, but also increased resource consumption. U-Net had the fewest parameters at only 16.9M, making it suitable for resource-constrained environments.

Regarding the inclusion of object detection functionality, both Mask R-CNN and PANet integrated this feature, enhancing their accuracy in handling occlusion and overlap issues in complex scenes. In contrast, FCN, DeepLabv3+, and U-Net did not integrate object detection capabilities, which may limit their segmentation effectiveness in practical farming scenarios.

In terms of model size (MS), DeepLabv3+ was the largest among the four models, with a size of 320MB, necessitating consideration of hardware resource limitations during deployment. The model sizes of FCN, PANet, and Mask R-CNN were relatively moderate, at 282MB, 268MB, and 178MB, respectively. U-Net had the smallest model size at just 121MB, making it suitable for resource-constrained applications.

Considering that the proposed method will ultimately be applied to sheep counting tasks in actual farming scenarios, a trade-off between segmentation performance and resource consumption is necessary. Based on the above test results, FCN, DeepLabv3+, and U-Net models were first eliminated. In the comparison between Mask R-CNN and PANet, PANet had a 2.15% higher accuracy than Mask R-CNN, but its model size was 33.6% larger than Mask R-CNN, and its parameter count increased by 19.9%. Therefore, considering the application requirements and resource consumption, Mask R-CNN was ultimately chosen as the base segmentation model for sheep counting.

Sheep counting network structure and optimization based on Mask R-CNN

The specific structure of the Mask R-CNN model is as follows (Figure 4).

Feature extraction network: Mask R-CNN uses ResNet50 as the backbone network, combined with a Feature Pyramid Network (FPN) to obtain multi-scale feature maps at different layers. The FPN improves object detection and segmentation performance by integrating high-resolution and low-resolution features through a top-down pyramid structure.

Region proposal network (RPN): Mask R-CNN first generates

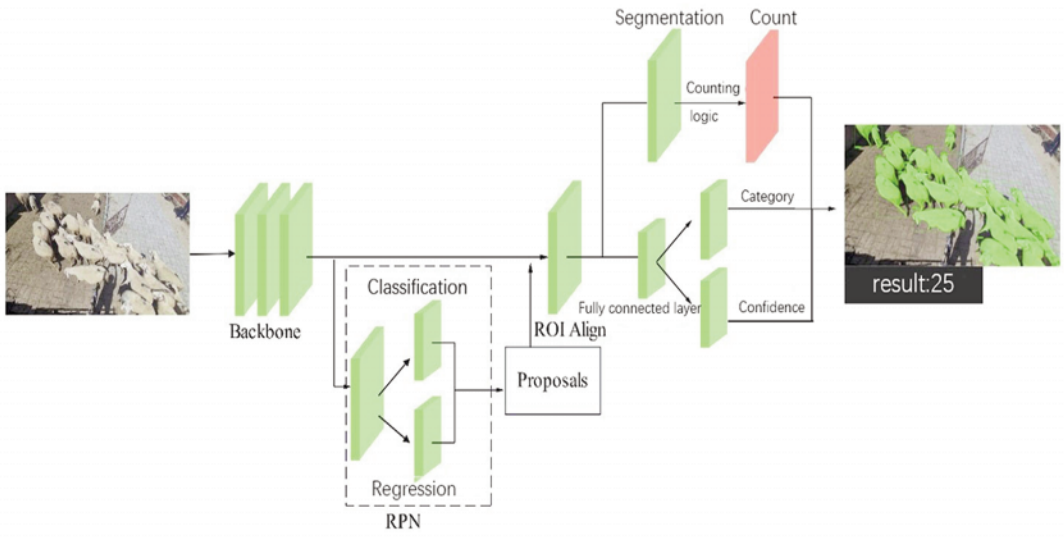

Figure 4. Structure diagram of counting network based on MaskR-CNN.

Table 1. Comparison of deep learning image segmentation experiments.

Algorithm	mAP (%)	Params/M	Use of object detection	MS/ M
FCN	41.6	73.8	No	282
Unet	40.1	16.9	No	121
DeepLabv3+	43.8	39.6	No	320
Mask R-CNN	47.4	23.5	Yes	178
PANet	49.5	29.4	Yes	268

candidate regions (Regions of Interest, RoI) through the RPN, which then classifies each candidate region and performs bounding box regression. The RPN utilizes a sliding window approach to generate multiple anchor boxes and filters out the regions most likely to contain objects.

Segmentation branch: unlike Faster R-CNN, Mask R-CNN adds a segmentation branch after bounding box prediction, generating pixel-level masks for the objects based on the candidate regions. The segmentation branch uses convolutional layers to map each RoI to a fixed-size mask output.

Model optimization: We enhance the Mask R-CNN by adjusting two key parameters—box_thresh (b_t) and mask_thresh (m_t) Use b_t and m_t for all subsequent cases.).

In Mask R-CNN, masks can more precisely outline the contours of objects, and effectively utilizing these masks can significantly enhance counting accuracy. To explore how to optimize counting precision using masks, we conducted experiments in this direction. During the training phase, the input to the mask branch consists of positive sample candidate regions generated by the RPN, as shown in Figure 5. By using the candidate regions provided by the RPN as training inputs, the model can adapt to various object positions and shapes in real applications, similar to the random cropping methods used in data augmentation. This approach improves the model's generalization ability and adaptability in actual farming scenarios by diversifying the training samples.

During the prediction phase of Mask R-CNN, the input to the mask branch comes from the detection results in the Faster R-CNN framework, which differs from the training phase. As shown in Figure 6, the input bounding boxes at this stage undergo Non-Maximum Suppression (NMS) to filter out a smaller number of precise candidate regions. NMS effectively eliminates overlapping or inaccurate candidate boxes, allowing the mask branch to only process high-quality bounding boxes, thus reducing computational load and achieving efficient inference. This method enhances the model's inference speed while ensuring segmentation accuracy, meeting the requirements for precision.

Through analysis, adjusting the two thresholds during Mask R-CNN inference process can affect the resulting masks. First, the bounding boxes input to the mask branch during inference come from the bounding boxes obtained after non-maximum suppression (NMS) in Faster R-CNN. By adjusting the NMS threshold, b_t, different quantities of bounding boxes can be generated, subsequently affecting the number of masks produced. As shown in the red box in Figure 7a, when b_t is lowered, more lower-scoring objects are included as candidate boxes.

In the mask branch, Mask R-CNN maps the predicted values of each pixel in the output mask to a range between 0 and 1 using the Sigmoid activation function. Following this principle, this

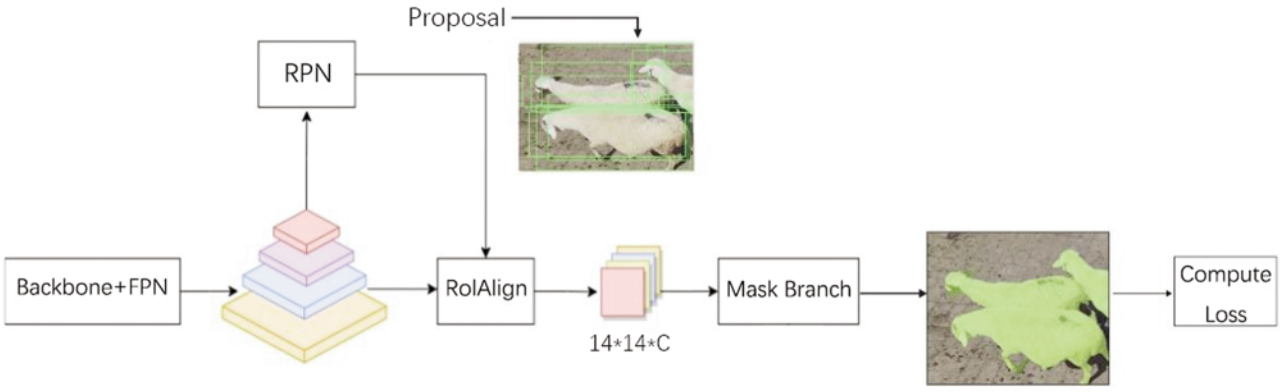

Figure 5. Mask R-CNN training process.

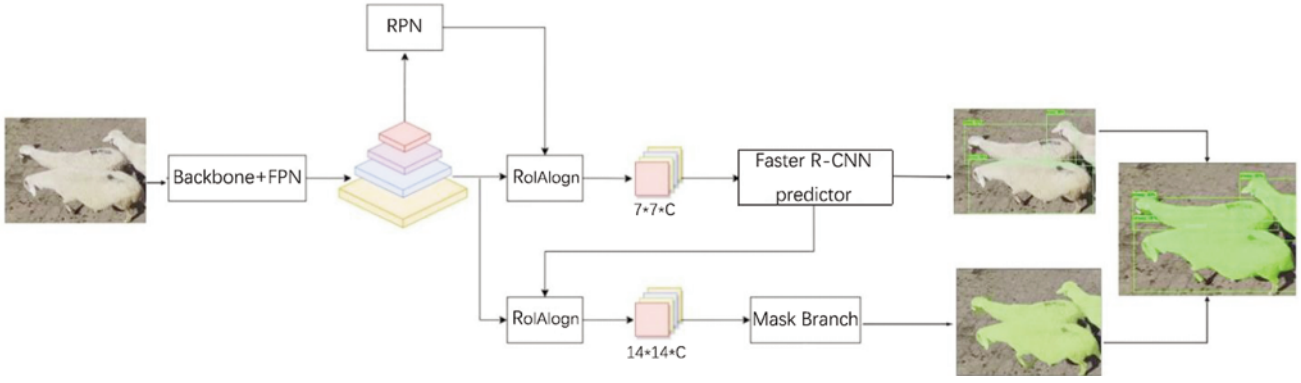

Figure 6. Mask R-CNN prediction process.

study sets m_t to represent the probability of a pixel belonging to a sheep. By adjusting m_t the area of the mask for the objects in the image changes; when m_t is increased, certain pixels with lower probability values are discarded and not included in the mask generation. The effect of raising m_t is shown in Figure 7b.

The threshold b_t controls which target regions can generate segmentation masks, while the threshold m_t regulates the size of the generated masks. By following these principles, a counting model can be designed to enhance counting accuracy.

When counting objects using segmentation masks, the thresholds b_t and m_t can be adjusted to optimize counting accuracy; b_t controls which target regions generate segmentation masks, while m_t determines the coverage of the masks. First, by setting b_t, it is determined whether the incoming targets in the segmentation phase are sheep. Targets greater than b_t are identified as sheep, while those below are considered background. In the segmentation phase, masks are generated for the regions confirmed as sheep. By default, Mask R-CNN sets m_t to 0.5. In this study, the result without m_t is defined as "Image 1," and the result with m_t as "Image 2." If the mask area difference between the two exceeds 80%, the mask is considered noisy and excluded from the count. This method effectively excludes noise points, ensuring that stable targets are counted, thereby enhancing counting accuracy and reliability. Finally, by accumulating the masks of sheep that meet the counting criteria, the total number of sheep can be obtained. The counting process is illustrated in Figure 8.

Results and Discussion

Threshold analysis and effect on counting accuracy

Table 2 shows the changes in counting accuracy under various thresholds. As the thresholds increase, the counting accuracy generally shows a trend of rising first and then falling.

From the trend in Table 2, it can be seen that when b_t is set to 0.6 and m_t to 0.7, the highest counting result reaches 96.09%, while other combinations yield relatively lower accuracy. This is because, at a b_t of 0.7 and an m_t of 0.9, the segmented instances generate corresponding masks that accurately identify the sheep, resulting in correct counting. By selecting appropriate values for these two thresholds, the method effectively avoids occlusion and missed detection problems in dense sheep scenes. In contrast, with other combinations, the segmented instances do not generate the corresponding masks precisely, leading to missed detections or false positives, whereby parts that should be identified as sheep or those that are not sheep are misidentified, thus reducing the counting accuracy. To further investigate the relationship between counting accuracy and the two thresholds, this study conducted experiments on the impact of a single threshold on counting accuracy. The results are shown in Figure 9b, illustrating how the counting accuracy changes as the b_t value varies from 0.6 to 0.98. The results indicate that b_t does indeed influence the performance of Mask R-CNN in sheep counting tasks. Setting a lower b_t allows the model to classify more areas as sheep, but it may also introduce false positives, such as mistakenly identifying similar objects or background interference as sheep. Conversely, a higher b_t means the model will only select areas with higher confidence, which can lead to missed detections of some actual sheep. In this experiment, the best performance was observed at a threshold of 0.9, achieving a counting accuracy of 95.50%. However, to combine with the m_t threshold for a more comprehensive identification of sheep, this study opted to set the b_t at 0.7. This value better controls the area of generated masks, ultimately improving overall counting accuracy. Figure 9a shows the variation in counting accuracy as the m_t value changes from 0.6 to 0.9. Adjusting the m_t can further enhance counting accuracy. When the m_t is low, the area of the generated mask increases, meaning that even pixels with lower probability values are included in the mask. However, if the threshold is set too high, some pixels that should generate a mask may be overlooked, leading to a reduction in mask area and failing to meet the required ratio for counting, resulting in their exclusion. Therefore, this experiment selected a m_t value of 0.9 to control the optimal range for mask generation, ensuring the most accurate sheep pixels are included in the count. This, in combination with the b_t value, ultimately achieves the best counting accuracy.

Table 2. Counting accuracy of box_thresh and mask_thresh combinations.

Box_thresh	Mask_thresh	Counting accuracy (%)
0.5	0.7	89.8
0.6	0.8	92.5
0.7	0.9	96.1
0.8	0.9	93.2
0.9	0.95	90.3
0.6	0.7	88.1
0.5	0.8	85.8
0.7	0.8	94.2
0.8	0.7	91.0
0.9	0.8	88.7

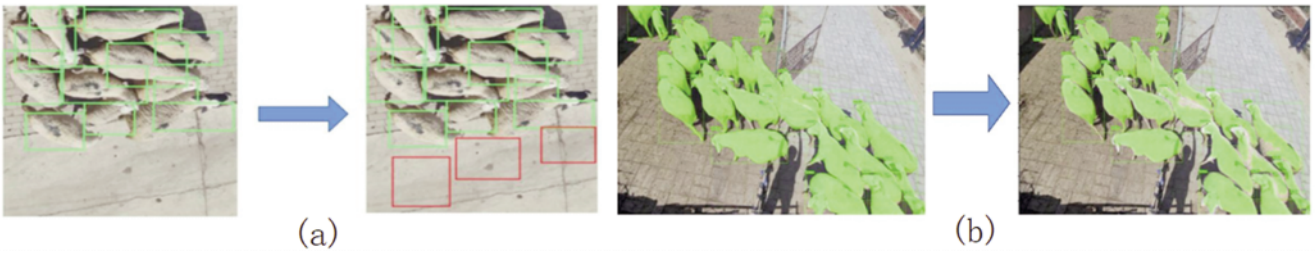

Figure 7. Effects of adjusting box_thresh and mask_thresh thresholds.

Experimental results and comparative analysis

To further evaluate the performance of Mask R-CNN, we conducted comparative experiments with YOLOv5, SSD, and DeepLabv3+, assessing parameter size, computation, inference time, model size, and counting accuracy. A generalization test was also performed using 50 images (25 from the test set and 25 newly collected unannotated images). As shown in Table 3, Mask R-CNN demonstrated a good balance of performance, resource consumption, and adaptability, making it the preferred choice.

Firstly, regarding the number of parameters, Mask R-CNN has 23.5M parameters, while YOLOv5M has 21.2M, representing a reduction of 9.8%, which helps decrease model complexity but results in lower performance in sheep counting. In contrast, SSD has a slightly higher parameter count at 25.1M, an increase of 6.7%, but its accuracy and generalization capability remain inferior to Mask R-CNN. DeepLabv3+ has the highest parameter count at 38.9M, exceeding Mask R-CNN by 65.4%, significantly increasing complexity and making it difficult to apply in resource-constrained scenarios.

Secondly, in terms of computational load, Mask R-CNN has a load of 21.6G. YOLOv5M's computational load is 24.5G, an increase of 13.5%, but its accuracy in sheep counting is low and does not fully utilize computational resources. SSD has the lowest computational load at 15.5G, reducing Mask R-CNN's load by 28.4%, making it suitable for resource-limited environments, though it suffers from accuracy issues. DeepLabv3+ has the highest computational load at 31.4G, exceeding Mask R-CNN by 45.3%, offering high accuracy but at a significant resource cost.

In terms of inference speed, Mask R-CNN takes 0.222s to process an image, while YOLOv5M only takes 0.119s, which is 46.2% faster and more suitable for high real-time demands, although it has lower accuracy in sheep counting. SSD processes an image in 0.212s, 4.3% faster than Mask R-CNN, but has low generalization capability. DeepLabv3+ is the slowest, taking 0.317s, which is 43.2% longer, making it unsuitable for efficient real-time sheep counting.

In terms of model size, Mask R-CNN's size is 339MB, while YOLOv5M is the smallest at only 44.3MB, reducing size by 86.9%, making it suitable for edge device deployment, though it performs poorly in counting accuracy. SSD has a model size of 95.7MB, a reduction of 71.8%, but still lags behind Mask R-CNN in accuracy and generalization capability. DeepLabv3+ is sized at

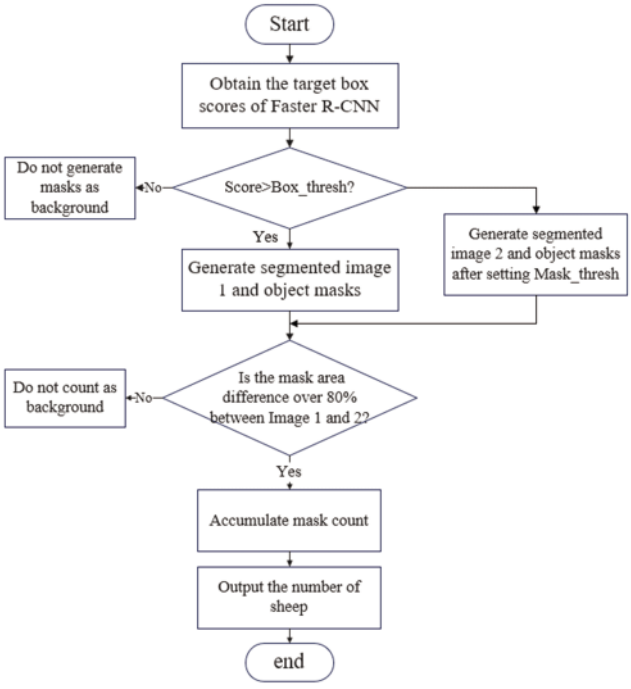

Figure 8. Counting flowchart.

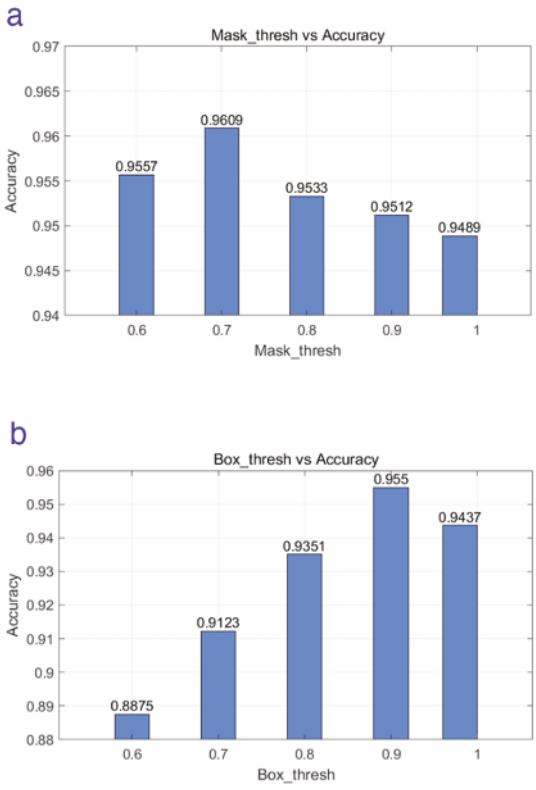

Figure 9. Relationship between threshold and counting accuracy.

Table 3. Comparison and generalization experiments.

Model	Params/M	Macc/G	Single frame time/s	MS/M	Counting accuracy	Generalization counting accuracy
Mask R-CNN	23.5	21.6	0.22	339	96.1%	88.8%
Yolov5M	21.2	24.5	0.12	44.3	80.3%	67.3%
SSD	25.1	15.5	0.21	95.7	83.1%	72.2%
DeepLabv3+	38.9	31.4	0.32	304	92.4%	85%

304MB, only 10.3% smaller than Mask R-CNN, but its computational load and inference speed are less favorable.

Regarding sheep counting accuracy, Mask R-CNN achieves 96.1%, outperforming all other models. YOLOv5M scores only 80.3%, showing a significant gap. SSD achieves 83.1%, which is 13% lower than Mask R-CNN. DeepLabv3+ scores 92.4%, close to Mask R-CNN but still behind. In terms of generalization counting accuracy, Mask R-CNN is at 88.8%, performing best, while YOLOv5M and SSD are at 67.3% and 72.2%, respectively, showing poor adaptability. DeepLabv3+ scores 85%, slightly lower than Mask R-CNN.

In summary, Mask R-CNN demonstrates excellent performance in counting accuracy and generalization ability. Despite its slightly higher resource consumption, it remains the top choice for practical sheep counting tasks.

Counting and generalization experimental results of Mask R-CNN model

The changes in mAP for the segmentation task during the training process of the basic Mask R-CNN counting model are shown in Figure 10. Since the initial model training utilized pre-trained weights, it was able to quickly learn features with high recognition capability from the dataset early on, resulting in a rapid increase in mAP to a high level. In the later stages of training, the mAP value gradually stabilized, indicating that there were no signs of overfitting during the training process and demonstrating the model's good generalization ability. In the inference prediction phase, the basic network performs overall quite well, as shown in Figure 11. The model is able to effectively recognize both occluded and clustered sheep targets. The output images from the basic model exhibit clear edges and accurate segmentation results, allowing for effective segmentation of sheep. This excellent segmentation capability provides strong support for image processing and sheep counting tasks, laying a solid foundation for practical application scenarios.

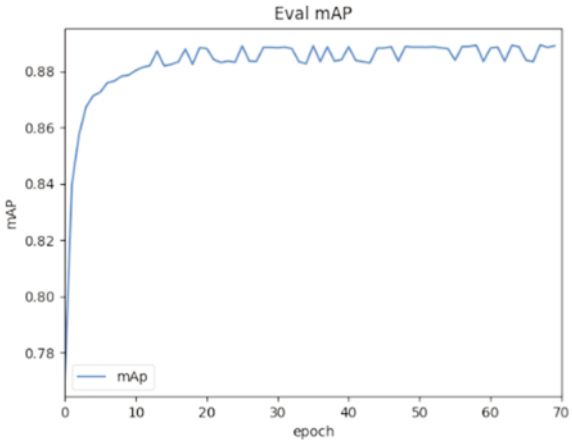

Figure 10. Basic Mask R-CNN mAP curve graph.

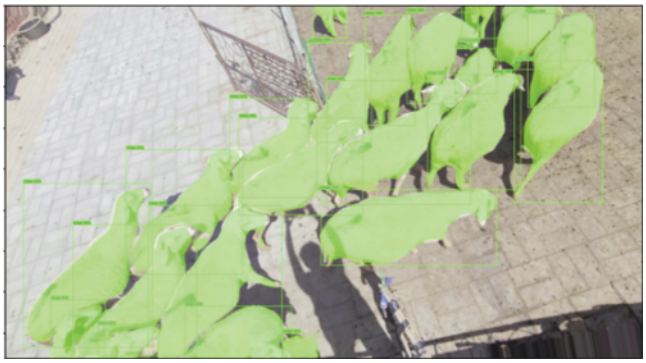

Figure 11. Basic Mask R-CNN network rendering.

While Mask R-CNN demonstrates excellent accuracy and generalization capability, its larger model size and computational resource demands may limit its effectiveness in scenarios requiring high real-time performance. YOLOv5, with its smaller model and faster inference speed, is suitable for applications where real-time performance is critical but accuracy requirements are lower. However, Mask R-CNN has clear advantages in handling complex target segmentation and overlapping issues. Therefore, for scenarios with slightly constrained resources, considering compression techniques for Mask R-CNN, such as model pruning and quantization, can help reduce computational resource consumption.

The above discussion clearly illustrates the performance of different parameters and models under various conditions, highlighting their strengths and weaknesses. Mask R-CNN performs exceptionally well and is suitable for sheep counting tasks, but it requires further optimization for resource-constrained environments.

Conclusions and future prospects

This study effectively addresses the issues of target density and occlusion in complex scenes instance segmentation technology. The counting accuracy reached 96.1%, significantly outperforming traditional detection methods. The innovation of this research lies in the first application of Mask R-CNN to sheep counting tasks, utilizing pixel-level segmentation to enhance precision, and achieving a balance between accuracy and false positive rates by optimizing bounding box and mask thresholds.

Additionally, this study conducted multi-scene testing based on real farming data from Inner Mongolia, validating the model's adaptability and robustness in complex environments and demonstrating its broad application potential. However, the computational complexity of Mask R-CNN remains relatively high. Future work could improve its real-time performance through model compression and lightweight architectures. Expanding to larger-scale datasets would further enhance the model's generalization. In addition, integrating multi-modal sensing technologies—such as drones and infrared sensors—could improve the comprehensiveness and accuracy of livestock monitoring. Reducing image resolution to lower computation time could also enable the use of lower-cost, lower-resolution cameras to build a networked surveillance system that not only counts animals but also tracks their movement, providing richer information beyond simple counting. Overall, this research provides a low-cost and efficient technological solution for intelligent management in animal husbandry and points the way forward for future optimization of model performance and expansion of application scenarios.

References

Bharati, P., Pramanik, A. 2020. Deep learning techniques - R−CNN to Mask R−CNN: A survey. In: Das, A., Nayak, J.,

Naik, B., Pati, S., Pelusi, D. (eds.), Computational Intelligence in Pattern Recognition. Advances in Intelligent Systems and Computing, vol 999. Singapore, Springer. pp. 657-68.

Bridge, E.S., Wilhelm, J., Pandit, M.M., Mordecai, R.S., Mountain, R.D. 2019. An Arduino-based RFID platform for animal research. Front. Ecol. Evol. 7:257.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.), Computer Vision – ECCV 2018. Lecture Notes in Computer Science, vol 11211. Cham, Springer. pp. 801-18.

Dac, H.H., Gonzalez Viejo, C., Lipovetzky, N., Tongson, E., Dunshea, F.R., Fuentes, S. 2022. Livestock identification using deep learning for traceability. Sensors 22:8256.

de Lima Weber, F., de Moraes Weber, V.A., de Moraes, P.H., de Oliveira, J.S., Lima, R.C.F. 2023. Counting cattle in UAV images using convolutional neural network. Remote Sens. Appl. 29:100900.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask R-CNN. Proc. IEEE Int. Conf. on Computer Vision, Venice. pp. 2961-9.

Liu, S., Qi, L., Qin, H., Shi, J., Jia, J., 2018. Path aggregation network for instance segmentation. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Salt Lake City. pp. 8759-68.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Boston. pp. 3431-40.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W., Frangi, A. (eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Cham, Springer. pp. 234-41.

Wang, G. 2019. Machine learning for inferring animal behavior from location and movement data. Ecol. Inform. 49:69-76.

Wang, Y., Xu, D. 2019. Cow target detection in farm environments based on YOLOv3 algorithm. J. Guangdong Univ. Petrochem. Technol. 29:31-35.

Xu, B., Wang, W., Falzon, G., Guo, L., Chen, G. 2020. Livestock classification and counting in quadcopter aerial images using Mask R-CNN. Int. J. Remote Sens. 41:2661-82.

Wang, Y., Xu, D. 2019. Cow target detection in farm environments based on YOLOv3 algorithm. J. Guangdong Univ. Petrochem. Technol. 29:31-35.

Xu, B., Wang, W., Falzon, G., Guo, L., Chen, G. 2020. Livestock classification and counting in quadcopter aerial images using Mask R-CNN. Int. J. Remote Sens. 41:2661-82.