scientific reports

Check for updates

OPEN

DASNet a dual branch multi level attention sheep counting network

Yini Chen[1,2], Ronghua Gao[1]✉, Qifeng Li[1], Hongtao Zhao[2], Rong Wang[1], Luyu Ding[1] & Xuwen Li[1,3]

Grassland sheep counting is essential for both animal husbandry and ecological balance. Accurate population statistics help optimize livestock management and sustain grassland ecosystems. However, traditional counting methods are time–consuming and costly, especially for dense sheep herds. Computer vision offers a cost–effective and labor–efficient alternative, but existing methods still face challenges. Object detection–based counting often leads to overcounts or missed detections, while instance segmentation requires extensive annotation efforts. To better align with the practical task of counting sheep on grasslands, we collected the Sheep1500 UAV Dataset using drones in real–world settings. The varying flight altitudes, diverse scenes, and different density levels captured by the drones endow our dataset with a high degree of diversity. To address the challenge of inaccurate counting caused by background object interference in this dataset, we propose a dual–branch multi–level attention network based on density map regression. DASNet is built on a modified VGG–19 architecture, where a dual-branch structure is employed to integrate both shallow and deep features. A Conv Convolutional Block Attention Layer (CCBL) is incorporated into the network to more effectively focus on sheep regions, alongside a Multi–Level Attention Module (MAM) in the deep feature branch. The MAM, consisting of three Light Channel and Pixel Attention Modules (LCPM), is designed to refine feature representation at both the channel and pixel levels, improving the accuracy of density map generation for sheep counting. In addition, a residual structure connects each module, facilitating feature fusion across different levels and offering increased flexibility in handling diverse information. The LCPM leverages the advantages of attention mechanisms to more effectively extract multi-scale global features of the sheep regions, thereby helping the network to reduce the loss of deep feature information. Experiments conducted on our Sheep1500 UAV Dataset have demonstrated that DASNet significantly outperforms the baseline MAN network, with a Mean Absolute Error (MAE) of 3.95 and a Mean Squared Error (MSE) of 4.87, compared to the baseline's MAE of 5.39 and MSE of 6.50. DASNet is shown to be effective in handling challenging scenarios, such as dense flocks and background noise, due to its dual–branch feature enhancement and global multi–level feature fusion. DASNet has shown promising results in accuracy and computational efficiency, making it an ideal solution for practical sheep counting in precision agriculture.

Sheep counting is recognized as a key factor in grazing management, as it enables the effective monitoring and control of grazing activities. Additionally, it is considered essential for promoting sustainable grassland development and optimizing livestock production[1]. Accurate accounting of the sheep population is necessary not only to help herders prevent overgrazing and grassland degradation but also to provide a scientific basis for the management and protection of grassland ecosystems[2]. Traditional methods of sheep counting have primarily relied on manual observation and recording[3], which are both time-consuming and labor-intensive. Furthermore, these methods are prone to environmental and human interference, often resulting in inaccurate outcomes. With advancements in animal wearable devices, technologies such as Radio Frequency Identification (RFID) tags[4], Electronic Identification (EID) ear tags[4], and GPS tracking collars[5] have been employed for accurate counting and localization of sheep flocks. However, these methods are considered costly, and challenges such as occasional device loss and maintenance difficulties have limited their widespread adoption. Therefore, the development of a more cost–effective and efficient sheep counting method for dense grassland flocks is regarded as of great practical significance.

[1]Information Technology Research Center, Beijing Academy of Agriculture and Forestry Sciences, Beijing 100097, China. [2]Department of Mathematics And Physics, North China Electric Power University, Beijing 102206, China. [3]College of Computer and Information Engineering, Tianjin Agricultural University, Tianjin 300384, China. ✉email: gaorh@nercita.org.cn

With advancements in remote sensing and computer vision, drone imagery has become a widely used data source in modern agriculture. By processing vast amounts of drone images, costs can be controlled effectively while obtaining the necessary information. Drone imagery is utilized not only for herd counting but also for counting crop plants[6], panicles[7], various animals[8], and specific tree species[9]. The images acquired by drones provide multi–resolution, multi–perspective, multis–cale, and multi–scene advantages, improving work efficiency and reducing human interference with natural ecosystems. In our task, drones are employed for data collection across vast grasslands, allowing for more convenient and rapid gathering of information. When combined with rapidly advancing computer vision technology, these drone–based counting tasks are made more efficient and accurate than ever before.

In the realm of computer vision, object counting has developed significantly across various domains. Innovations and breakthroughs have been targeted in crop counting[10], crowd-counting[11], vehicle detection[12], remote sensing image detection[13], and class agnostic counting[14]. As an intersection between object counting and livestock farming, sheep counting still presents substantial energy for growth. While sheep counting shares some similarities with crowd counting, and both face certain common challenges in object counting, our sheep dataset composed of drone–captured images introduces unique difficulties. These challenges include the large background areas, the adhesion between individual sheep, and the interference from background objects that are similar in size and color to the sheep, all of which urgently need to be addressed.

Hence, we propose a novel approach for generating more refined density maps by developing DASNet, which is built upon the traditional VGG–19 convolutional neural networks. In addition, we introduce a dual–branch attention structure and the Multi–Level Attention Module (MAM) to effectively address the problems of accurately counting dense sheep flocks UAV images. The main contributions of this work are as follows:

1. We constructed the Sheep1500 UAV Dataset, a dense grassland sheep dataset under multi-scale complex scenarios. The dataset features the Ujimqin sheep and contains 1,500 images with 149,751 sheep instances, providing data support for large-scale dense sheep counting.
2. To address the problem in the Sheep1500 UAV Dataset where density map regression counting networks struggle to focus on sheep regions. We designed a dual-branch attention network. This allows the feature extraction backbone network to extract both shallow detail features and deep global features simultaneously without increasing parameters, enhancing the network's focus on sheep regions and improving the accuracy of predicted density maps.
3. Considering the limitation of the backbone network in effectively utilizing deep features, we designed the MAM for the deep feature extraction branch. Comprising three Light Channel and Pixel Attention Modules (LCPM), along with light channel attention, pixel attention, and convolutional layers, and with residual connections between components, it enables efficient fusion of multi-scale global features extracted by the LCPM modules. The attention mechanisms assign appropriate weights to features at each scale, resulting in deeper features with richer information compared to those from ordinary convolutional layers.
4. For better multi-scale feature extraction in the deep branch, we designed the LCPM, which includes a Light Channel Attention Layer (LCAL), a Pixel Attention Layer (PAL), and convolutional layers. Both attention mechanisms are composed of convolutional layers with small convolutional kernels. As the network deepens, these small kernels minimize information loss, helping the network capture more deep feature information. This design makes the network focus more on sheep regions and reduces interference from background objects irrelevant to counting, thus reducing counting errors.

The rest of the article is organized as follows, First, we introduce the related work, summarizing the current state of research on object detection counting methods, instance segmentation counting methods, and density map regression counting methods. Secondly, we present the optimizations and improvements made to our network. Based on the MAN network, we improve the backbone to better utilize multi–scale features, addressing the challenges of sheep scale variation and background object interference in the dataset. Finally, we compare and analyze the experimental and visualization results.

Related work

Over the past few decades, object counting has become increasingly complete. Depending on the specific task requirements, the current mainstream methods for object counting can be categorized into three types: detection–based methods, segmentation based methods, and density map regression based methods.

Object detection counting methods

Detection based methods often employ YOLO[15] and SSD[16] as backbone networks, predominantly applied in sparse scenarios. Li et al.[17] designed a cross–line partitioning counting method to overcome the duplicate counting and improved YOLOv7 to detect wheat ears. Bello et al.[18] integrated a masking mechanism into the backbone of the YOLOv7 algorithm to achieve instance segmentation of a single cow object for auxiliary target detection counting. Cao et al.[19] proposed an automatic sheep counting network on grasslands that combines a YOLOv5–based object detection algorithm with a tracking algorithm. This network achieves sheep counting by detecting and tracking sheep heads passing through a channel.

The applicability of these methods relies on counting through close-range target identification. In contrast, our counting work is based on capturing top–down images at varying altitudes using UAVs. In terms of object detection and counting methods on UAV datasets, Sangaiah et al.[20] addressed the challenges of collecting paddy disease datasets using drones by enhancing the YOLOv4 architecture with multi–scale feature extraction modules and attention mechanisms to boost the network's ability to detect diseases. Anandakrishnan et al.[21] proposed the Li–YOLOv9, which is lightweight yet ensures high detection accuracy for rice seedlings. Yang et

al.[22] proposed a deep learning network for detecting and counting Amorphophallus konjac plants during the high coverage growth stage. This network, based on UAV imagery, utilized the YOLOv5–3CBAM architecture to achieve high detection accuracy and precise counting results.

However, the counting method based on object detection is considered more suitable for scenes where the number of target objects is small, the individuals are distinct, and the scale variation is minimal. In cases where there is significant occlusion between objects and scale variations among the targets, the performance of counting models based on object detection networks becomes suboptimal. Our sheep dataset presents challenges not only due to the significant occlusion between sheep caused by the large flock size but also due to the uneven density distribution resulting from the dynamic movement of the sheep. The object detection method has low detection accuracy for these problems, so using this method is not the best choice.

Instance segmentation counting methods

To address the occlusion issue, some scholars have proposed using instance segmentation methods for object counting. Counting methods based on instance segmentation networks primarily involve segmenting the individual target objects, such as rice panicles[23] and food[24], and then counting the generated masks. Nguyen et al.[25] proposed that FoodMask Network simultaneously encodes food category distribution and instance height at the pixel level, addresses instance recognition and provides prior knowledge for extraction. This approach better resolves the occlusion issues encountered in object detection. In addition, Wang et al.[26] developed a high–density group pig counting model that integrates multi–scale feature pyramids with deformable convolutions to improve accuracy in complex scenarios. Similarly, Liu et al.[27] introduced a PDCL block in their instance segmentation-based counting model, leveraging deformable convolutions to enhance performance on multi–scale object segmentation. Some methods segment[28] the entire group of target objects, aiming to better eliminate interference from background objects. Dolezel et al.[29] used instance segmentation–methods to determine the location and quantity of cattle or sheep in large livestock farms, effectively achieving animal localization and counting even under occlusion.

Although the issue of mutual occlusion between objects can be addressed by instance segmentation based counting methods, they require substantial labeling costs for segmentation annotations. This is particularly challenging in our dataset, where sheep are free grazing and dispersed, with flock sizes reaching thousands or even tens of thousands. Because the sheep dataset was captured by drone, the smaller sheep are often heavily occluded by larger ones, and dark–colored sheep tend to blend into the background, making distinction difficult. However, current segmentation networks are not yet sufficiently mature to effectively segment objects with similar pixel edges. Consequently, instance segmentation methods are considered unsuitable for counting in extremely dense scenes.

Density map regression counting methods

Nevertheless, density map regression counting methods can overcome some challenges posed by high–density sheep images, such as partial occlusions between sheep and interference from background colors in grassland scenes. This approach has been proven effective in crowd counting and localization tasks[30–35], where datasets are characterized by high occlusion, significant variation in head distribution, and densely packed target objects. These characteristics closely resemble those found in our drone–captured grassland sheep datasets, making density map regression counting methods more suitable for handling the high altitude, dense grassland sheep herding imagery.

In terms of detection precision, density map regression–based counting methods[36] offer pixel–level regression, reducing the reliance on image quality, while leveraging deep learning to directly map original images to quantities, addressing both object counting and localization tasks. The DMseg–Count model proposed by Zang et al.[37] significantly enhances the detection accuracy of wheat spikes by augmenting local contextual supervision information. The model demonstrates excellent performance in complex field environments, effectively addressing issues of occlusion and overlap among wheat spikes. Chen et al.[38] proposed an attention–based multi–scale convolutional neural network model for accurate mosquito swarm counting from images. The model achieves high precision mosquito swarm counting by generating density maps through a feature extraction network and an attention–based multi–scale regression network.

In terms of data annotation, the cost falls between that of bounding box annotation and instance segmentation-based methods, as density map regression only requires point annotations for target objects, resulting in lower annotation costs.

Compared to object detection and instance segmentation, density map regression methods can more efficiently address the challenges posed by multi–scale variations and occlusions or clumps in dense flocks in UAV sheep datasets. Therefore, we have chosen this method to perform accurate counting on the Sheep1500 UAV Dataset and to provide technical support for the refined management of grassland pastures. However, for density map regression networks, since they can only estimate the target regions, achieving one hundred percent accurate counting is still challenging. Given the counting error issues in our dataset, the primary focus of this paper is to bridge this gap by optimizing the backbone network.

Methods

In this section, we provide a detailed description of the creation process for our herding dataset, as well as the specifics of our reconstruction of DASNet based on the VGG–19 architecture.

Dataset construction

The production process of the dataset in this study is divided into two parts: (1) the actual grassland sheep flock data collection part. (2) the data preprocessing on the collected dataset. We collected data on the widely

distributed Ujimqin sheep herd in Xilinhot, Inner Mongolia, China, in the spring of 2023. The data was captured using a DJI Matrice 300 RTK drone equipped with a Zenmuse L1 camera, as illustrated in Fig. 1. This setup allows for high–definition video recording at a resolution of 1920×1080 and the ability to hover at a fixed altitude during filming. These features ensure that the collected videos are both clear and stable.

Under different natural lighting and natural scenes, we filmed approximately 300 min of sheep activity video with a pixel resolution of 1920×1080 and a frame rate of 60, which includes data at different heights from 25 to 100 m. From each captured video, we extract one image every 120 frames. The extracted images are then divided according to the scale ratio of $25\text{m} : 50\text{m} : 75\text{m} : 100\text{m} = 2 : 5 : 3 : 1$ to ensure a diverse range of scales, as illustrated in Fig. 2; additionally, considering the diversity of scenes, we collected data of sheep entering and leaving the pen at different times of the day, drinking and resting inside the pen, and grazing outside the pen, as shown in Fig. 2. After rigorous screening, we selected 1,500 high–quality images from the collected videos, all in JPG format with a resolution of $720x \times 720$ pixels. We used labeling software to annotate points on the dataset, which we named the Sheep1500 UAV Dataset. Figure 2 visually present examples from the Sheep1500 UAV Dataset.

Architecture of DASNet

In this section, we provide a detailed explanation of the DASNet network, which is primarily composed of two main components: the feature extraction module and the transformer with a regression decoder. The feature extraction module builds upon VGG–19[39], which we have divided into a deep feature branch and a shallow feature branch. Both branches integrate the Conv CBAM Layer (CCBL), resulting in a dual–branch attention structure. Furthermore, we have fused the Multi–Level Attention Module (MAM) into the deep branch to enhance its performance. Addressing the main challenges faced by our Sheep1500 UAV Dataset, the high density of sheep in drone–captured grassland scenes, background object interference in sheep counting, and occlusion between individual sheep, we utilize a dual–branch feature to enhance the structure. This design enables the network to effectively integrate shallow texture details with deep semantic information. Moreover, the deep feature branch mitigates the impact of noise features, enhancing the overall feature learning capacity of the backbone network, this allows the network to focus more accurately on the sheep flock areas, improving counting precision.

Our DASNet's overall structure is illustrated in Fig. 3. Unlike density map regression counting networks[34,40–42] that directly use VGG–16 as the backbone network, we have chosen the deeper VGG–19 networks as our initial feature extraction component. We have removed the last fully connected layers of the network, which has a minimal impact on the accuracy of sheep counting while effectively reducing network parameters. Additionally, previous work[40,43] has demonstrated that enriching low–level features with additional semantic information or incorporating more spatial information into high–level features can significantly improve the effectiveness of feature fusion. Therefore, building upon the work of predecessors, we split the VGG–19 networks into shallow and deep features, with shallow features comprising the first 10 layers and deep features encompassing layers 11 to 21. By processing deep and shallow features according to their distinct characteristics, our network effectively handles multi–scale features, enhancing its robustness and generalization ability.

Shallow features, found in the early layers of the network, are primarily responsible for extracting low–level visual characteristics such as texture and color. They have a small receptive field, typically focusing on local area information, which makes them well–suited for capturing fine structural details. Deep features, found in the later layers of the network, typically extract high–level semantic information and are capable of capturing global context. To address the challenge of subtle pixel–level differences between the boundaries of sheep and

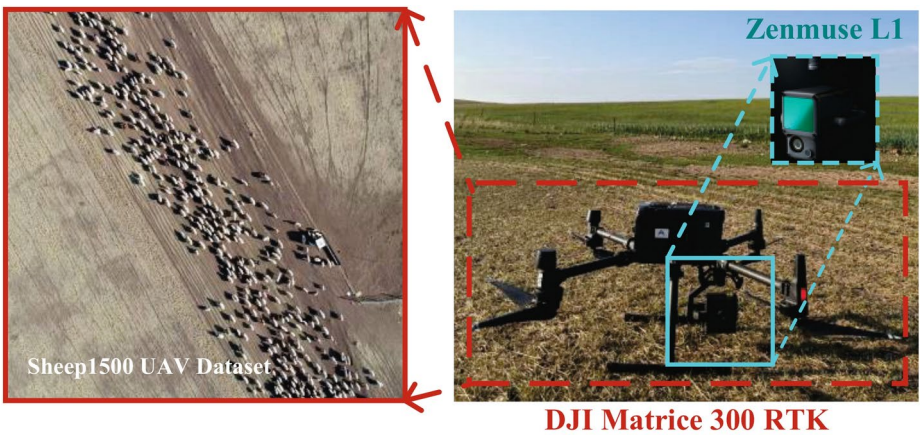

Fig. 1. The DJI Matrice 300 RTK drone and Camera parameters. The scene in the image is from the Sheep1500 UAV Dataset.

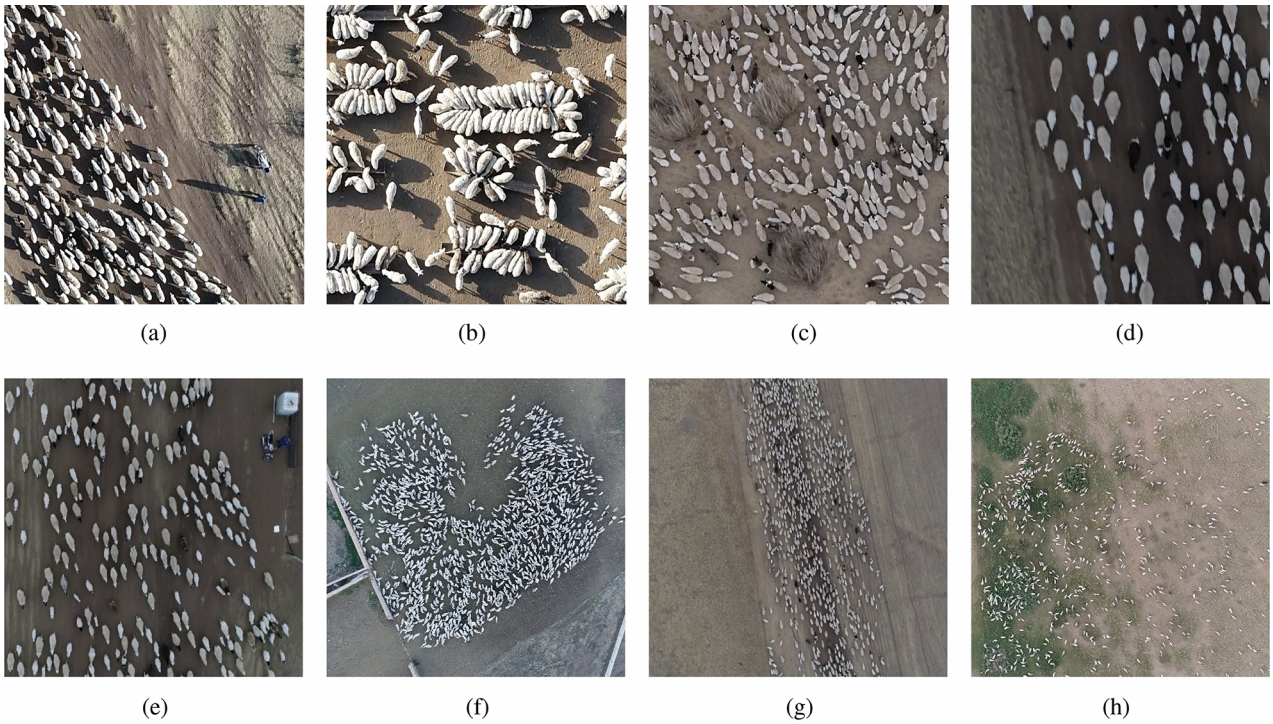

Fig. 2. The above figure is from our own Sheep1500 UAV Dataset and presents sample images captured at different altitudes and in various scenes. (**a**) Scene with people and cars. (**b**) Drinking water scene. (**c**) Scene with haystacks. (**d**) Dark sheep and background. (**e**) 25 m. (**f**) 50 m. (**g**) 75 m. (**h**) 100 m.

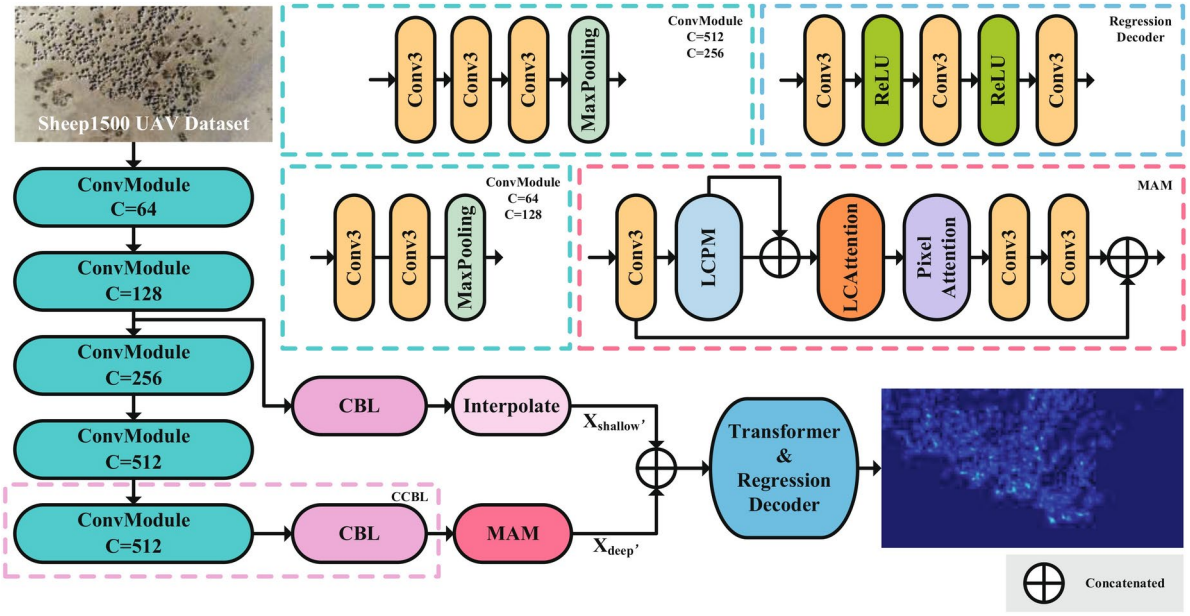

Fig. 3. The structure of DASNet consists of a feature extraction module located before the Regression Decoder. Our constructed module MAM is applied exclusively in the deep feature branch. The input images are from the Sheep1500 UAV Dataset.

the background objects, as well as among the sheep themselves, we have integrated the CCBL Layer into both the shallow and deep feature branches. This integration enables DASNet to emphasize the sheep areas while effectively reducing the influence of background objects. In addition, the Multi–Level Attention Module (MAM) we designed further suppresses the interference of global background noise. Previous baseline networks have revealed that simply employing density map regression methods can result in the misestimation of background

objects that are similar in size to sheep, leading to them being incorrectly classified as sheep. The introduction of the MAM in the deep feature branch has effectively addressed this issue.

Dual-branch attention architecture

To enhance the feature encoding capabilities of both the shallow and deep feature extraction branches in DASNet, the Conv CBAM Layer (CCBL) is introduced. The CCBL is composed of a Conv Module and a CBAM module. Adaptive in nature, the CBAM module learns features from different areas of the input image, thereby increasing DASNet's attention to the characteristics of sheep, which in turn improves the overall performance of the network. As shown in Fig. 4, the CCBL can learn channel and spatial information at the same time, which can enhance model generalization capabilities. Furthermore, as the module does not alter the size of the input features, it requires no additional computational resources or parameters.

The input features of the CBAM first pass through channel attention and then through spatial attention. We assume the input image to be $X_{input} \in \mathbb{R}^{H \times W \times 3}$through shallow branch and deep branch output the $X_{shallow} \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times 128}$ and $X_{deep} \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times 512}$. We take $X_{shallow}$ as an example, first, we pass $X_{shallow}$ through the CAL, where both average pooling and global average pooling are applied. These operations compress the spatial dimensions of each channel to 1×1, generating unique feature descriptors for each channel. The resulting features are then processed through a shared fully connected layer (MLP) for feature fusion. This shared fully connected layer first reduces the number of channels to 1/16 of the original, followed by restoring the channel count to its initial size, thereby achieving both feature compression and expansion. The two feature maps obtained after the fully connected layer are concatenated through element–wise addition and then passed through the Sigmoid activation function. The CAL learns the global channel characteristics of the input features and dynamically adjusts the weight of the channels, thereby enhancing the network's selectivity for features. The CAL calculation formula is as Eq. (1):

$$X_{shallow_{ca}} = Sigmoid(MLP(MaxPooling(X_{shallow})) \bigoplus MLP(AveragePooling(X_{shallow}))) \quad (1)$$

The features output from the CAL are element–wise multiplied with the input features before being passed into the SAL. The SAL applies both average pooling and max pooling operations to the features, compressing the channel dimension. The pooled feature maps are concatenated and then processed by a convolutional layer with a 7×7 kernel, generating a single–channel feature map. This map is subsequently activated using the Sigmoid function to produce the corresponding weights. The addition of the SAL helps emphasize important sheep regions in the image while suppressing irrelevant or insignificant background information like haystacks and people.

Finally, the module employs a residual connection to fuse the input features with the output of the SAL. We denote the final output of the CCBL as the sum of these features. The formulas for the SAL and the output feature $X_{shallow'}$ are given as Eqs. (2) and (3):

$$X_{shallow_{sa}} = Sigmoid(Conv_{7\times7}(MaxPooling(X_{shallow_{ca}}) \bigoplus AveragePooling(X_{shallow_{sa}}))) \quad (2)$$

$$X_{shallow'} = X_{shallow_{ca}} \bigotimes X_{shallow_{sa}} \quad (3)$$

Multi-level attention module for a deep feature branch

Since deep features capture more abstract global representations, they may also include irrelevant or interfering elements, such as stone slabs with pixel patterns resembling sheep, herders, or dark–colored sheep that blend into the background. To reduce these interferences and enhance the deep feature extraction capability, we implemented this through the fusion of multi–scale features. Drawing inspiration from the FFANet approach[44], we simplified the feature fusion module, leading to the design of the Multi-Level Attention Module (MAM), as shown in Fig. 5.

The MAM is composed of three Lightweight Channel and Pixel Attention Modules (LCPMs), which are fused using residual connections and then individually cascaded into the Light Channel Attention Layer (LCAL), Pixel Attention Layer (PAL), two 3×3 convolutional layers, and residual structures. After the feature matrix from the deep feature extraction branch enters the MAM module, it first undergoes dimensionality adjustment through a convolutional layer. Subsequently, it is passed into three LCPM group modules, yielding three feature

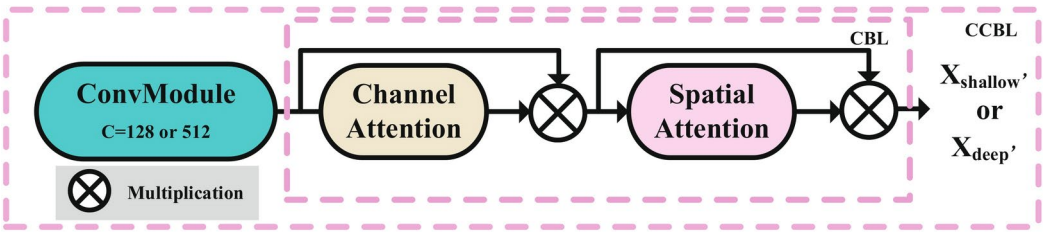

Fig. 4. The basic structure of CCBL: the CBAM module with Conv. CBAM consist of a Channel Attention Layer (CAL) and a Spatial Attention Layer (SAL).

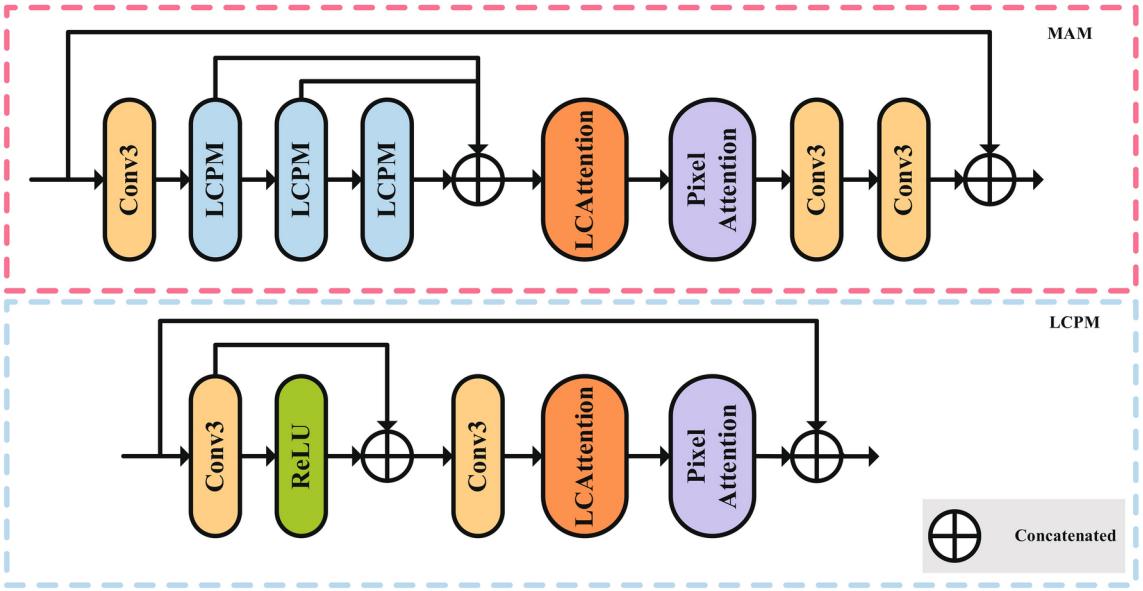

Fig. 5. The structure of the MAM and the structure of the group module LCPM.

matrices of different sizes. These matrices are then concatenated using residual connections to form the output feature. The concatenated feature is processed through light channel attention to generate weights, which are used to adjust the contribution of each LCPM to the final output. The fused features are further optimized through pixel attention, resulting in more refined spatial features. Finally, the features are passed through two 3×3 convolutional layers to restore the channel count to that of the original input feature. The MAM module fuses and optimizes the multi–scale features output from the LCPM, which helps us reduce the loss of feature information.

The composition of LCPM

The LCPM group module is a critical component of the MAM, for each LCPM, the input features first pass through a 3×3 convolutional layer followed by ReLU activation, after which the residual is added to the input features. The resulting feature map is sequentially processed through a second 3×3 convolutional layer, the LCAL, and the PAL, before being combined once again with the input feature residual, generating the final output features of the LCPM.

The basic structure of the LCAL is illustrated in the upper right section of Fig. 6. First, the input features undergo average pooling, to obtain the global feature for each channel. Next, the features are passed through two 1×1 convolutional layers: the first reduces the number of channels to 1/8 of the original, and the second restores them to their original size. Finally, Sigmoid activation is applied to the output. The 1×1 convolution used in the LCAL effectively integrates channel information while avoiding the mixing of spatial information. We suppose f represents the input feature, *Sigmoid* is the activation function, and $Conv1$ is a 1×1 convolution. The formula for the LCAL is as Eq. (4):

$$X_{deep_{lca}} = Sigmoid(Conv_{1\times1}(ReLU(Conv_{1\times1}(AveragePooling(f))))) \bigotimes f \tag{4}$$

The PAL implements pixel–level feature processing, enhancing the information of important pixels while suppressing less important ones. It employs a 1×1 convolutional layer followed by a Sigmoid activation function, which compresses the output values into the range (0,1) to serve as pixel–level attention weights. In the PAL, the output pixel attention features are element–wise multiplied with the input original features to achieve pixel–level weight adjustment. Let f represent the input feature, $Sigmoid$ the activation function, and $Conv1$ the 1×1 convolutional kernel. The formula for the PAL is given in Eq. (5).

$$X_{deep_{pa}} = Sigmoid(Conv_{1\times1}(f)) \bigotimes f \tag{5}$$

Loss function

Our loss function consists of two components: Bayesian loss[45] and cosine similarity loss, which are used in combination. Traditional counting methods convert point annotations into ground truth density maps and apply pixel–wise supervision. However, the quality of these density maps is often compromised due to factors such as occlusion, perspective distortions, and variations in the shape of target objects. The Bayesian loss function addresses these issues by constructing a density contribution probability model based on point annotations and using the expected count as the supervisory target. This approach avoids the limitations of strict pixel–level

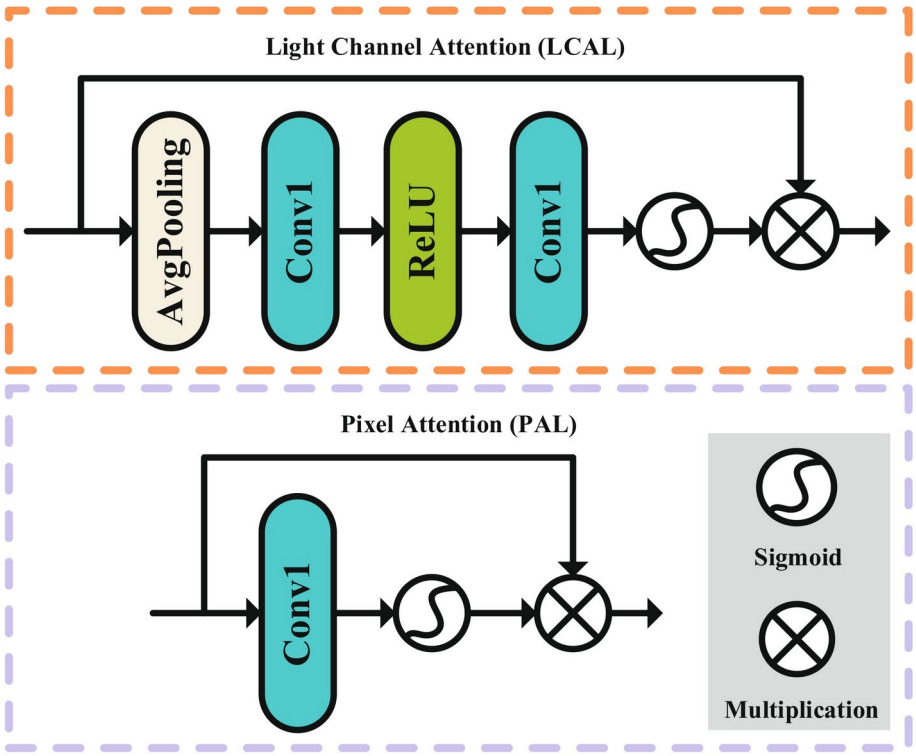

Fig. 6. The structure of the LCAL and the structure of the group module PAL.

supervision, enhancing the accuracy and robustness of the counting model. The calculation formula is as Eqs. (6) and (7):

$$e^j = \left| 1 - \sum_p Prob_j(\mathrm{P}) \cdot D_p \right| \tag{6}$$

$$Loss_B = \sum_j^N m_j \cdot e^j \tag{7}$$

In this context, j represents the j-th label point, and D refers to the final predicted density map, $Prob_j(\mathrm{P})$ represents the posterior probability of the j–th label point at the **P** location. m_j represents the map that serves as the attention feature map signal used for supervision, e^j reflecting the relationship between the predicted values and the ground truth. The bias function encapsulates this relationship and constitutes our Bayesian loss function.

Additionally, we incorporate the cosine similarity loss function to further enhance the model's performance. This loss function calculates the similarity between each feature and the mean feature, encouraging the model to learn more consistent and concentrated features from similar inputs. This helps improve the model's robustness. The calculation formula for the cosine similarity loss function $Loss_{consine}$ is as Eq. (9):

$$G_j = 1 - \frac{\sum_j^N m_j \cdot \bar{m}_j}{(\bar{m} \sum_j^N (m_j^2))^{1/2} + 10^{-5}} \tag{8}$$

$$Loss_{cosine} = \sum_{j=0}^{H \times W} G_j \tag{9}$$

We define m_j as the j–th attention map, and $\bar{m}_j$ as the average feature map of the j–th attention features, used to calculate the similarity between the j–th sample feature and the average feature. $\bar{m}$ represents the L2 norm of all the j average features, serving as a normalization factor. G_j denotes the cosine distance between each feature and the average feature and measures the dissimilarity between them. Finally, the calculation of our loss function is given by Eq. (10). To balance the weights of the cosine similarity distance loss and the Bayesian loss, we conducted relevant ablation experiments, and the results are shown in Table 1. Through these experiments, we determined the value of λ to be 100.

λ	1	20	50	100	120
MAE	4.64	4.55	4.32	3.95	3.97
MSE	5.92	5.83	5.55	4.87	4.92

Table 1. Ablation experiments on the loss function.

Sheep1500 UAV dataset	Images number	Instance number	Max	Min	Average
Train_data	1200	120528	552	1	100.44
Test_data	300	29223	567	4	97.41
Total_data	1500	149751	567	1	98.93

Table 2. This table summarizes the instance statistics for the Sheep1500 UAV Dataset, including image count, total instance count, and maximum, minimum, and average instance numbers for both Train_data and Test_data.

$$Loss = Loss_B + \lambda Loss_{cosine} \tag{10}$$

Experiments

Experimental details

Our experiment is based on Linux and PyTorch 1.10.1, with Python 3.8.18 and CUDA 11.1. Each image is subjected to horizontal flipping and random scaling, with the original size of our dataset images being 720×720 pixels, and the size of the randomly cropped image blocks being 512×512. Adam[46] is utilized as the optimizer, with a learning rate set to 10^{-6} for optimizing parameters, and the loss weight λ is set to 100, with a uniform training epoch of 300.

Sheep1500 dataset and evaluation metrics

Experimental training and verification were conducted on our self collected Sheep1500 UAV Dataset shown in Table 2. Our dataset comprises 1500 photos with 149751 instances, which contain 1200 photos for training and 300 photos for testing. The average number of instances in the dataset is around 100. The resolution of the images is all 720×720.

The performance of the counting model is primarily evaluated through two metrics: Mean Absolute Error (MAE) and Mean Squared Error (MSE). The definitions of these two metrics are Eqs. (11) and (12). In the above formula, N denotes the number of sample images, while X_i^{gt} and X_i^{pre} respectively denote the actual and predicted number of sheep in the i–th image. In addition, the MAE is less sensitive to outliers compared to the MSE, making it more stable. However, when used in conjunction, lower values for both metrics indicate better model performance.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|X_i^{gt} - X_i^{pre}\right| \tag{11}$$

$$MSE = \left(\frac{1}{N}\sum_{i=1}^{N}\left(X_i^{gt} - X_i^{pre}\right)^2\right)^{1/2} \tag{12}$$

Comparison experiment and ablation experiment

In this section, we further demonstrate the advantages of our network through ablation experiments, comparison experiments, and visualizations. Our network strikes a well-balanced trade–off between model size, parameter count, and inference speed, as demonstrated in Table 3. Building on this solid foundation, it consistently outperforms other models in both ablation studies and comparative experiments. The results of the ablation studies, shown in Table 4, highlight the effectiveness of our architectural choices.

Table 3 presents the results of our network compared to other networks. Our network achieves the lowest error metrics, with an MAE of 3.95 and an MSE of 4.87. Under the same structure of a dual–branch network, our model is larger and has a longer inference time compared to SFANet[47] from 2019. However, DASNet achieves superior performance, significantly reducing error metrics with a decrease of 1.28 in MAE and 3.00 in MSE. Besides, we employed the MAN[36] model, which also utilizes the density map regression strategy, for our sheep counting task. However, our network achieved better performance, reducing the MAE by 1.44 and the MSE by 1.63 compared to MAN. In addition, compared to the IIM[48] and STEERER[49], our network has fewer parameters and a faster inference speed. While these three networks, which focus on crowd localization counting tasks, generally achieve higher accuracy than density map regression networks, our DASNet reduced the MAE and MSE by 1.28 and 1.63, respectively, compared to the lowest values among them. In summary, DASNet achieves the highest accuracy in both regression and localization-based counting methods, while maintaining a moderate

Model	Year (Journal)	Estimated total size (MB)	Params size (MB)	Average inference time (ms)	MAE	MSE
SFANet[47]	2019 (CVPR)	64.86	17.0	30.14	5.23	7.87
IIM[48]	2020 (CVPR)	262.76	261.73	114.25	8.87	13.25
MAN[36]	2022 (CVPR)	154.19	154.16	29.16	5.39	6.50
STEERER[49]	2023 (ICCV)	4435.27	2217.64	63.493	5.90	8.20
DASNet	–	326.55	326.49	35.37	3.95	4.87

Table 3. Comparison of different model sizes, parameter quantities,inference speeds, MAE and MSE. And compare our method with other methods.

Module	Estimated total size (MB)	Params size (MB)	Average inference time (ms)	MAE	MSE
VGG–19(MAN as Baseline)	154.19	154.16	29.16	5.39	6.50
VGG–19+DB	155.44	155.41	28.22	4.39	5.56
VGG–19+DB+CCBL	155.58	155.55	29.80	4.34	5.50
VGG–19+DB+MAM	320.42	320.36	33.61	3.79	5.07
VGG–19+DB+CCBL+MAM	326.55	326.49	35.37	3.95	4.87

Table 4. Ablation experiments illustrates the effect of various modules on the performance of DASNet. DB represent the dual–branch structure.

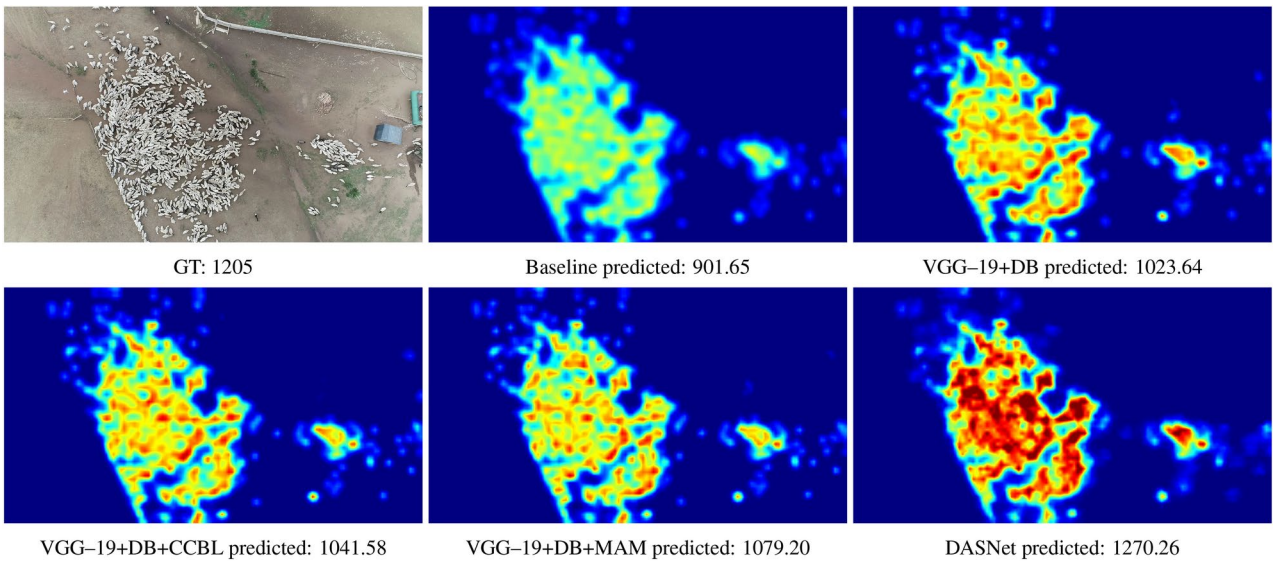

Fig. 7. We selected an image from the **Sheep1500 UAV Dataset** and presented the visualization results of each part of the **ablation study**.

model size and speed. It also demonstrates notable advantages when compared to high precision localization networks.

Our ablation experiments, shown in Table 4 and Fig. 7, are based on the VGG–19 baselines. We verified the effectiveness of both the dual–branch attention structure and the MAM structure introduced in our network. Our error metrics decreased at each step, ultimately achieving optimal results with an MAE of 3.95 and an MSE of 4.87. Clearly, our baseline model has the smallest number of parameters, with a size of 154.19 MB, and the fastest inference speed at 29.16 ms per image, its accuracy remains limited. As the feature map extracted from VGG–19 is directly fed into the regression module, resulting in relatively rough outputs and the largest errors. The MAE and MSE are 5.39 and 6.50, respectively. To enable the network to capture both shallow texture features and deep global features, we introduced a dual–branch structure. As shown in Table 4, this modification improves the accuracy of density map generation, leading to a reduction in errors, with MAE and MSE decreasing by 1.00 and 0.94, respectively. While the dual–branch structure improves accuracy to some extent, it does not fully optimize the use of focused sheep areas and global features. Therefore, we introduced the CCBL in both

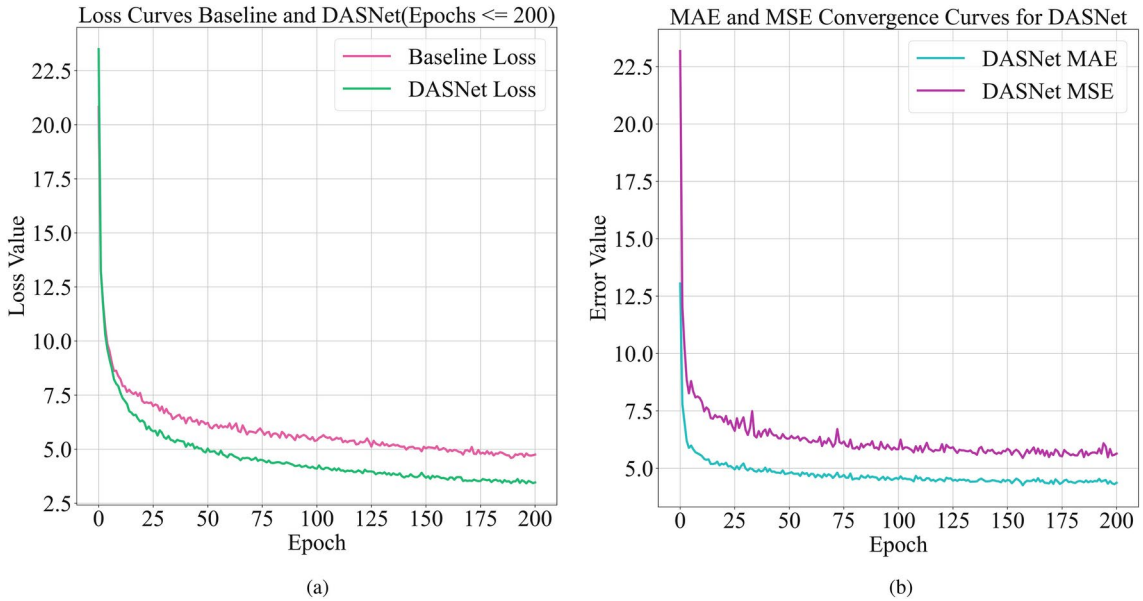

Fig. 8. Loss convergence line chart and DASNet's MAE & MSE convergence line chart. (**a**) The Loss of the Baseline and our network. (**b**) The MAE and MSE of DASNet.

AttentionModule	Year (Journal)	Estimated total size (MB)	Params size (MB)	Average inference time (ms)	MAE	MSE
Coordinate Atten[50]	2021 (CVPR)	326.55	326.49	72.97	4.43	5.33
EC Atten[51]	2020 (CVPR)	326.38	326.35	52.15	4.03	5.04
Axiel Atten[52]	2019 (CVPR)	407.95	407.89	106.66	5.89	6.93
Cota Atten[53]	2022 (TPAMI)	335.74	355.68	60.29	17.65	23.67
Acmix Atten[54]	2022 (CVPR)	329.78	329.74	121.38	5.40	6.29
Ours+CBAM[55]	2018 (ECCV)	326.55	326.49	35.37	3.95	4.87

Table 5. Replace the other attention module in the DASNet structure and prove that adding the CBAM module is optimal by replacing different attention modules.

branches to enhance the networ's focus on the sheep regions. This operation not only maintains the model size and inference speed but also further reduces MAE as 4.34 and MSE as 5.50. In addition, to better utilize deep global features, we designed the MAM structure. The model size increased to 320.42 and the inference speed slowed down to 33.61 ms per image, due to the additional computations from the multi–level attention and residual connection structures in MAM. However, the model's accuracy improved significantly compared to the baseline, highlighting the advantages of the MAM module in enhancing deep feature representation. Therefore, we ultimately integrated the dual–branch attention structure with the MAM module to create DASNet. Since the dual–branch attention structure has minimal impact on model size and inference speed, DASNet's size is 326.55 MB, and its inference speed is 35.37 ms per image similar to the network with only MAM. Our errors were further reduced, demonstrating that the combined network delivers superior performance. As shown in Fig. 8, our network converges more effectively and exhibits a lower loss compared to the baseline.

In addition to the ablation experiments, we highlight the significance of this module by replacing the CBAM module in our network. Table 5 and Fig. 9 present various attention mechanisms that have demonstrated excellent performance in recent years. We compared different attention mechanisms by integrating them into the dual–branch structure. It is evident that the inclusion of CBAM not only reduces the overall model size to 326.55 MB and achieves an inference speed of 35.37 ms per image but also optimizes the model's error parameters. Other attention mechanisms, such as Coordinate Attention[50] and Axial Attention[52], which enhance model performance from a pixel perspective, or EC Attention[51], Cota Attention[53], and Acmix Attention[54], which focus on channel optimization and self–attention, are less suitable for our sheep counting task. This confirms that a combined approach, leveraging both spatial and channel attention, is more effective in improving the accuracy of density map regression–based counting networks. The experimental results indicate that incorporating the CBAM module optimizes both model size and inference speed, while also achieving optimal levels for the detection metrics MAE and MSE.

Besides, We selected StarNet[58], FastNet[57], and ConvNeXt[56] as our feature extraction backbone networks for comparison experiments, and the results are shown in Table 6 and Fig. 10. DASNet achieved the lowest error

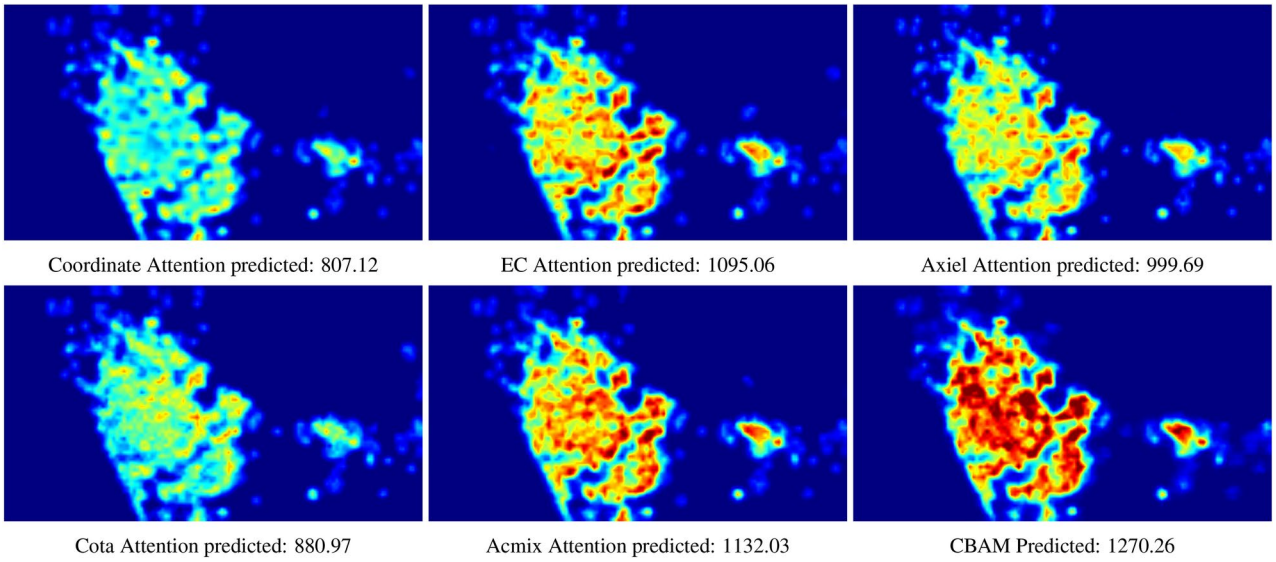

Fig. 9. We selected an image from the **Sheep1500 UAV Dataset** and presented the different visualization effects of **various attention mechanisms** in our network architecture.

Backbone	Year (Journal)	Estimated total size (MB)	Params size (MB)	Average inference time (ms)	MAE	MSE
ConvNext[56]	2022 (CVPR)	255.22	255.14	29.00	6.32	8.25
FastNet[57]	2023 (CVPR)	99.51	99.37	15.06	5.93	8.76
StarNet[58]	2024 (CVPR)	99.42	99.23	18.84	6.66	9.51
Ours DASNet	–	326.55	326.49	35.379	3.95	4.87

Table 6. Replace different backbone networks to demonstrate that our DASNet is more effective.

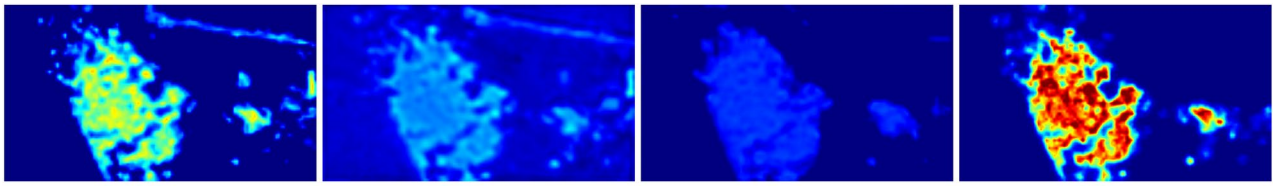

Fig. 10. We selected an image from the **Sheep1500 UAV Dataset** and presented the different visualization effects of **various Backbone** in our network architecture.

metrics while maintaining a moderate model size. Compared to the best-performing metrics from 2022 to 2024, our network reduced the error by 1.98 and 3.38, respectively.

Visualization

We selected visualization results at varying heights and across different scenarios, as shown in Figs. 11 and 12, respectively, to demonstrate the applicability of our network on the Sheep1500 UAV Dataset. Additionally, Figs. 13 and 15 illustrate the visual differences between DASNet and the other models, highlighting the advantages of our network in addressing the sheep counting problem. We also compared DASNet and the baseline model on public datasets, as shown in Figures 14.

Figure 11 presents the overall visualization of the sheep counting network in various scenarios. In particular, scene (a) depicts sheep crowded together while drinking water in a fence, making it more challenging for network learning, especially when capturing images from higher altitudes. Therefore, unlike the other three scenarios, we selected the image in the 25m altitude column for this scene to better illustrate the network's performance. Scene (a) intuitively shows that the estimated count of the baseline network exceeds the ground truth value of 452, while DASNet effectively reduces this error, lowering the estimated number from 470.63 to 464.77. The other three scenes take place outside the fence. Scene (b) shows a crowded area outside the fence, while scene

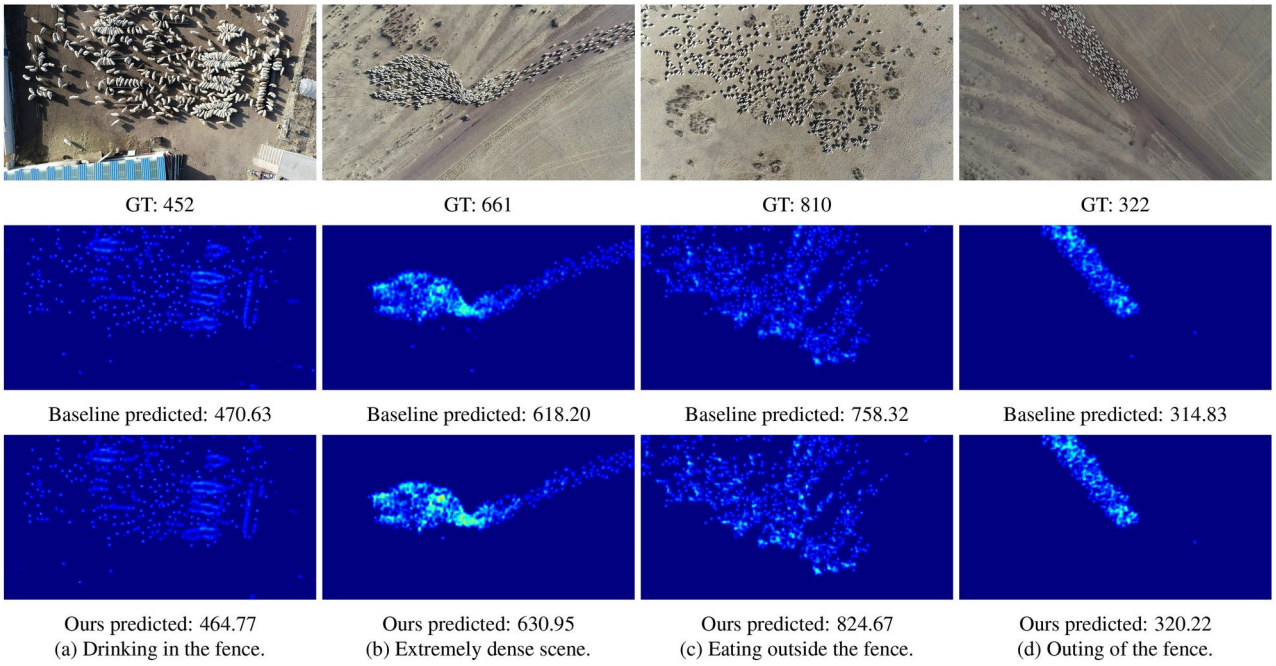

Fig. 11. The figure presents the visualization results of the **Sheep1500 UAV Dataset** in **different scenarios**. We selected different test images to display the visual results. Each column respectively represents the original image with the real number of sheep, the number of Baseline predictions, and the number of DASNet predictions.

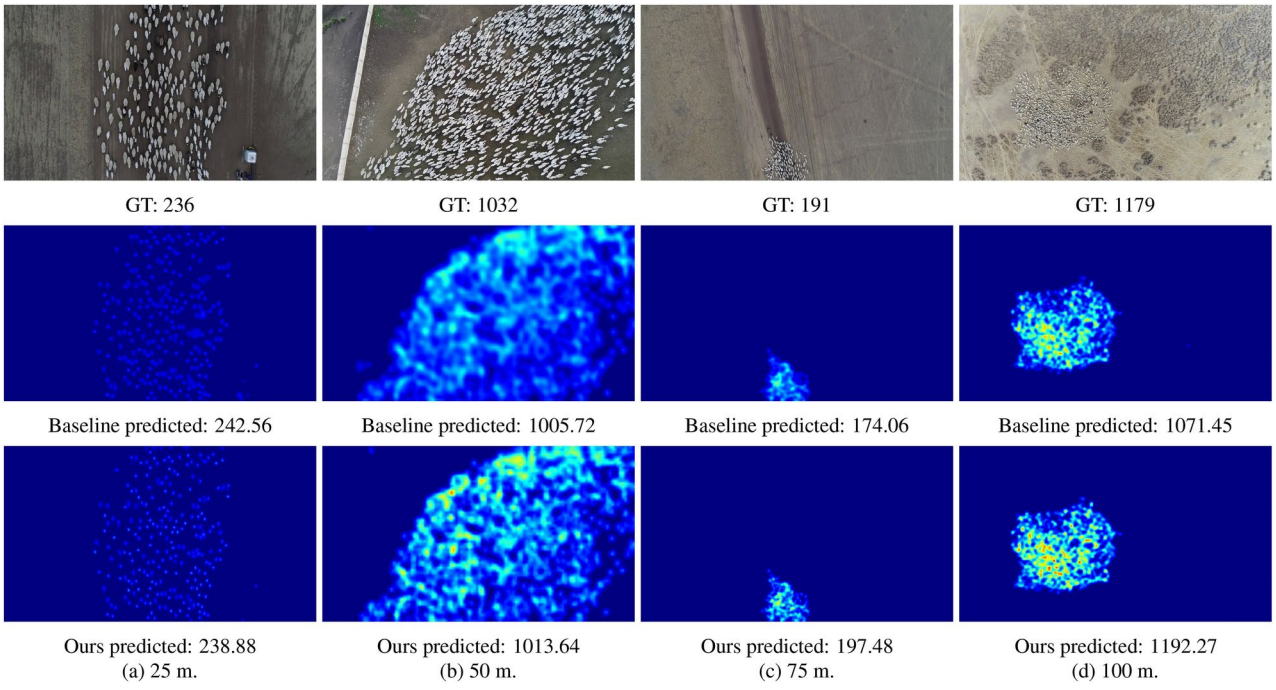

Fig. 12. The above figure presents the comparison results of the **Sheep1500 UAV Dataset** at **different scales**. Each column respectively represents the original image with the real number of sheep, the number of Baseline predict and the number of DASNet predictions.

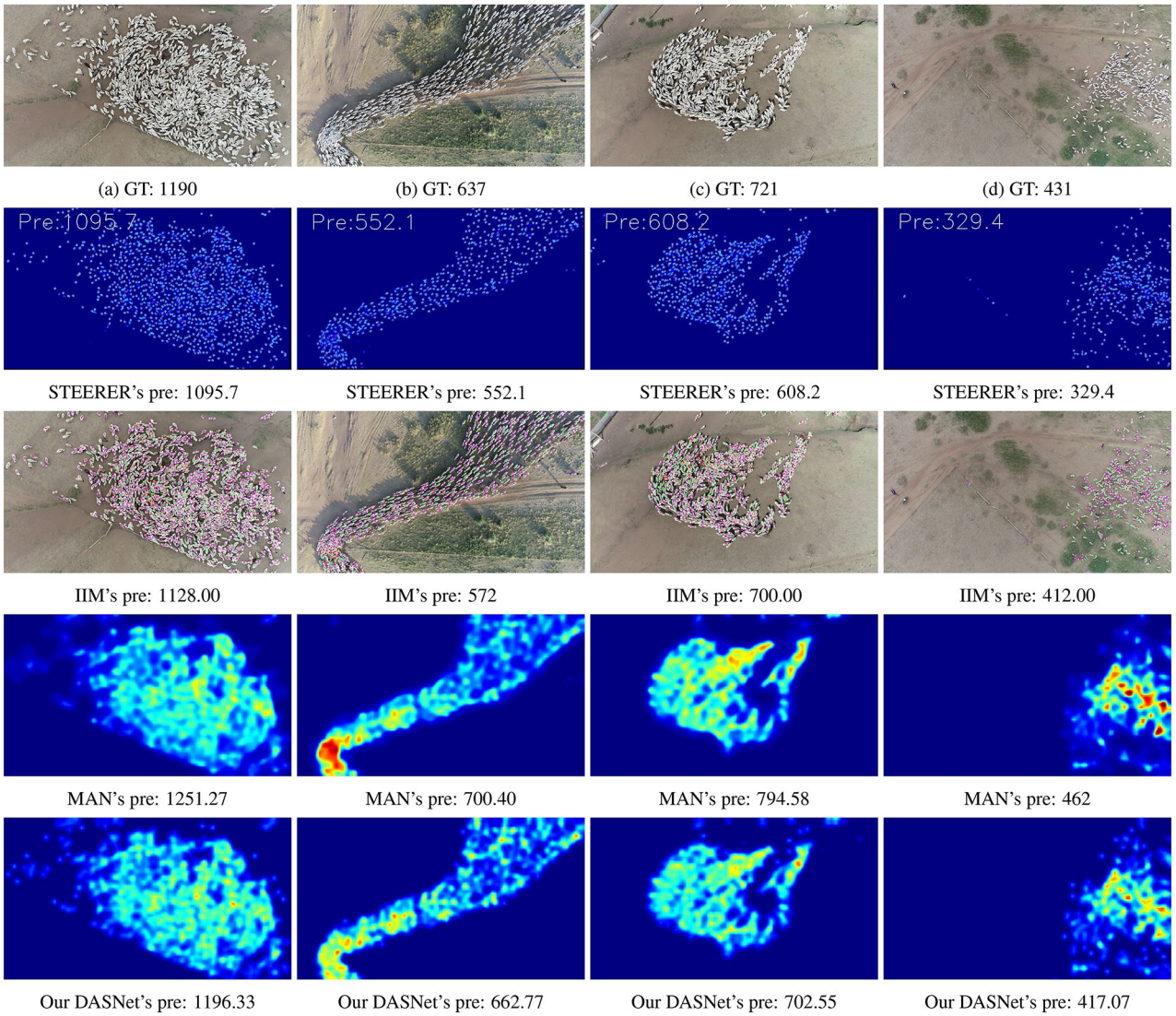

Fig. 13. The above figure presents the comparison results of the **Sheep1500 UAV Dataset** across **different models**. Each row displays the visualization results of different networks. It should be specifically noted that in the STEERER visualization results, the red dots represent the GT, and the Gaussian density map below the dots indicates the prediction results. In the IIM network visualization results, the pink dots represent the prediction results, while the green and red dots indicate the GT.

(c) captures a relatively sparse grazing scene. Scene (d) depicts the sheep returning to the fence, where the distribution is more uniform and elongated. As shown in Figures (b) and (c), our network provides estimates closer to the true value than the baseline, particularly when counting densely packed flocks. In areas where the sheep are closely connected, DASNet demonstrates stronger visualization features, leading to more accurate estimations. Additionally, Figures (b) and (d) demonstrate that our network is more effective in distinguishing between sheep and backgrounds with similar size and color. DASNet can better suppress irrelevant background noise, resulting in estimates that are closer to the ground truth value. As shown in Figure (d), the estimate increases from 314.83 to 320.22, which is only 1.78 off from the ground truth.

Furthermore, Fig. 12 highlights the advantages of our network in multi scale scenarios. By utilizing a dual–branch attention structure, our network effectively addresses the multi–scale challenges presented by the Sheep1500 UAV dataset. Whether processing high–altitude or low–altitude images, as well as dense or sparse sheep populations, our approach consistently demonstrates superior counting accuracy in handling scale variations. Figure 12 (a) shows a 25m low–altitude, sparse image. Compared to the baseline, our network better isolates the feature points for each sheep, leading to an improvement in counting accuracy, with the error reduced from 6.56 to 2.88. Figure 12 (c) differs from (a) as it depicts a high–altitude, sparse sheep image. At this altitude, the feature points are less distinct compared to the low–altitude image. Nevertheless, our network still demonstrates high accuracy in processing this scenario.

We selected the visualization results of different networks from 2020 to 2023 on our Sheep1500 UAV Dataset, as shown in Fig. 13. Each row in the figure represents the original image, the ground-truth labels converted

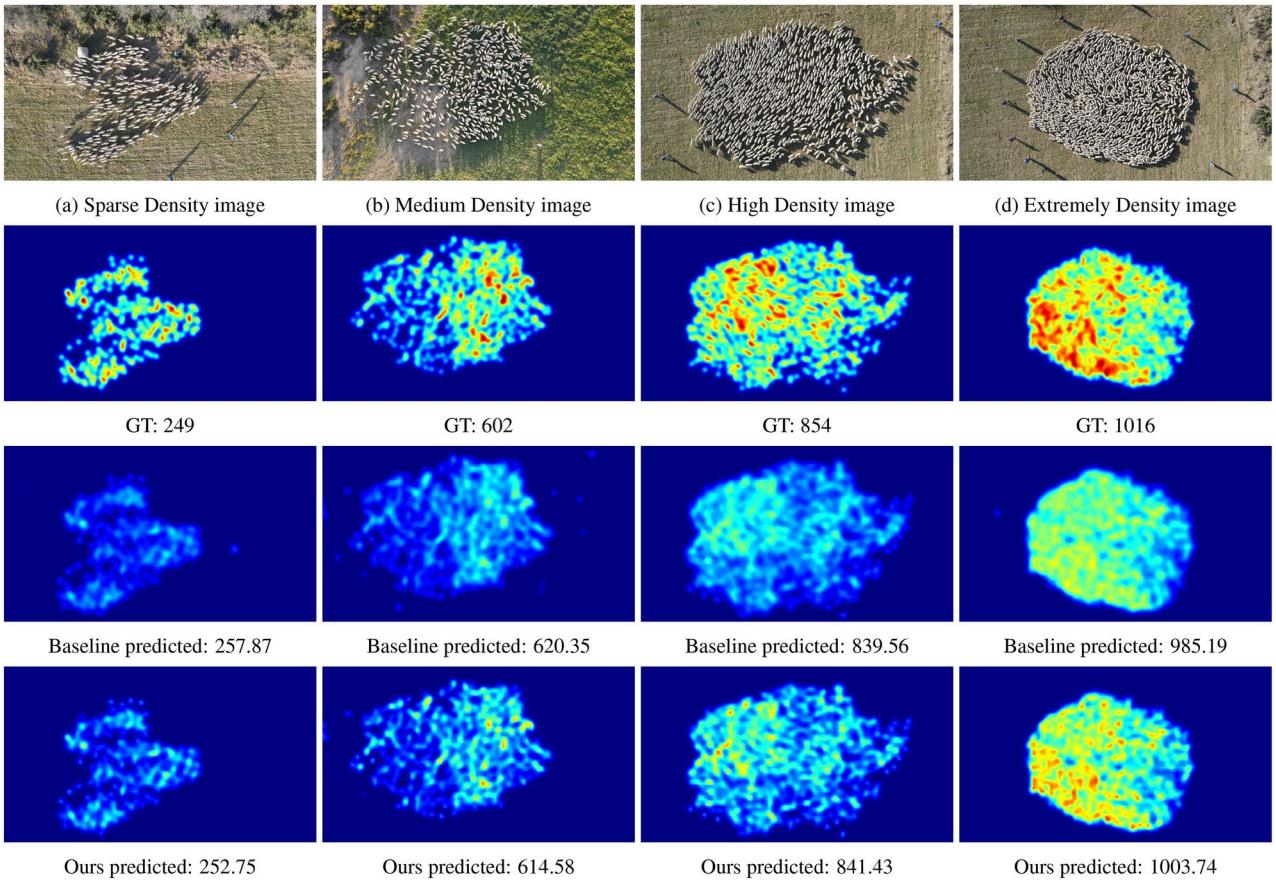

Fig. 14. Visualization of density map for **Public Datasets**[59] and sheep counts in **different density**. We selected different test images to display the visual results. Each column respectively represents the original image with the real number of sheep, the number of Baseline predict and the number of DASNet predictions.

into a density map, the prediction results of STEERER, the prediction results of IIM, the prediction results of MAN, and the prediction results of our network, respectively. The STEERER and IIM networks are point supervised counting networks, while the MAN and DASNet are density map regression–based networks. The results indicate that point–supervised counting networks still suffer from the problems of missing sheep and misidentifying background objects. This also demonstrates that our network is capable of better addressing the diverse density distribution of sheep flocks, achieving an average accuracy rate of 90%.

Figure 14 presents the visualization results of our network on a public dataset of South African Merino sheep[59]. We selected sheep flock images with varying density distributions for prediction and calculated the error metrics of the baseline and DASNet. The MAE and MSE of the baseline are 18.09 and 19.82, respectively, while those of DASNet are 10.29 and 10.96. These results indicate that our improved network is not only effective on this dataset but also generalizable to other datasets. It demonstrates the ability to handle sheep flock images across different densities and focuses more effectively on the regions where the sheep are located, thereby minimizing interference from irrelevant background features.

Both Fig. 12d and 14d depict high–altitude, dense sheep images, where many sheep overlap and crowd together. In these cases, DASNet outperforms the baseline by better focusing on areas with clustered sheep, thereby reducing issues of missed or incorrect detections.

The top and bottom rows of Fig. 15 demonstrate the ability of the baseline network and DASNet, respectively, to resist background noise interference in both crowded and general scenes. We used rectangular boxes to highlight sections of the background in the real image, baseline predicted image, and DASNet predicted image for comparison. From the areas framed in the original image, it is clear that there were no sheep present in these regions, yet the baseline incorrectly predicts feature information. In contrast, our network effectively overcomes this issue. Moreover, adding the MAM module to the deep branch has proven to be effective. The MAM module enhances the network's ability to focus on the sheep areas and efficiently fuse multi–level features, further learning global feature information. This helps in eliminating background noise, allowing DASNet to produce higher precision density maps.

Conclusion

This study proposes a Dual–Branch Multi–Level Attention Sheep Counting Network (DASNet), which significantly improves counting accuracy by designing a dual-branch attention structure and incorporating

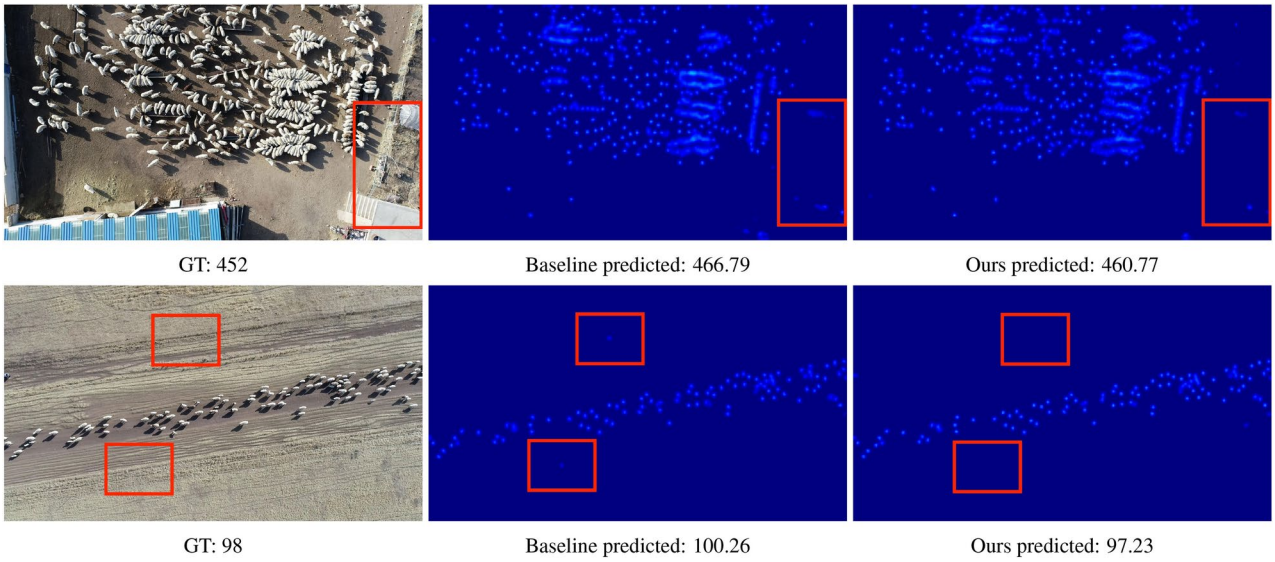

Fig. 15. We selected several images from the **Sheep1500 UAV Dataset** to verify the effectiveness of DASNet in removing background noise. Each row respectively represents the ground truth, the number of sheep predicted by the baseline network, and the number of sheep predicted by DASNet. The areas outlined in red and blue show the effect before and after the network improvement.

a Multi–Level Attention Module (MAM) in the deep feature extraction branch. DASNet enhances the representation of sheep regions while suppressing background interference, demonstrating strong robustness in complex scenarios.

Despite its success in density map regression–based counting tasks, DASNet still relies on density maps as supervision and predicts the actual count by generating Gaussian density maps. However, this approach does not fully leverage the coordinate information of individual sheep, making it difficult to eliminate errors fundamentally.

Currently, for counting tasks in dense scenes, novel supervision methods such as point-based supervision are rapidly emerging. The advantage of point-based supervision lies in its ability to fully utilize the coordinate information in images, thereby significantly reducing counting errors. This approach may provide a more effective solution for accurately counting sheep in large–scale dense scenes. Additionally, point–based localization counting methods yield clearer and more intuitive visual outputs compared to density maps.

In future research, we will further explore innovative methods for dense sheep counting in grasslands, striving to overcome the limitations of density map regression-based counting methods.

Data availability

Source data are available from Yini Chen (cyn19990609@163.com) or Ronghua Gao (gaorh@nercita.org.cn) upon request.

Code availability

All code related to the model was implemented in Python. Code related to the deep learning models is available from the authors upon request.

Received: 25 October 2024; Accepted: 8 April 2025
Published online: 02 July 2025

References

1. Ming-Zhou, L. et al. Review on the intelligent technology for animal husbandry information monitoring. *Sci. Agric. Sin.* **45**, 2939. https://doi.org/10.3864/j.issn.0578-1752.2012.14.017 (2012).
2. Chen, Y., Li, S., Liu, H., Tao, P. & Chen, Y. Application of intelligent technology in animal husbandry and aquaculture industry. In *2019 14th International Conference on Computer Science & Education (ICCSE)* 335–339 (2019).
3. Sarwar, F., Griffin, A., Rehman, S. U. & Pasang, T. Detecting sheep in uav images. *Comput. Electron. Agric.* **187**, 106219 (2021).
4. Aquilani, C., Confessore, A., Bozzi, R., Sirtori, F. & Pugliese, C. Precision livestock farming technologies in pasture-based livestock systems. *Animal* **16**, 100429 (2022).
5. Handcock, R. N. et al. Monitoring animal behaviour and environmental interactions using wireless sensor networks, gps collars and satellite remote sensing. *Sensors* **9**, 3586–3603 (2009).
6. Yang, T. et al. Unmanned aerial vehicle-scale weed segmentation method based on image analysis technology for enhanced accuracy of maize seedling counting. *Agriculture* **14**, 175 (2024).
7. Chen, Y. et al. Refined feature fusion for in-field high-density and multi-scale rice panicle counting in uav images. *Comput. Electron. Agric.* **211**, 108032 (2023).
8. Chen, A., Jacob, M., Shoshani, G. & Charter, M. Using computer vision, image analysis and uavs for the automatic recognition and counting of common cranes (grus grus). *J. Environ. Manage.* **328**, 116948 (2023).

9. Chowdhury, P. N. et al. Oil palm tree counting in drone images. *Pattern Recogn. Lett.* **153**, 1–9 (2022).
10. Qian, Y. et al. Mfnet: multi-scale feature enhancement networks for wheat head detection and counting in complex scene. *Comput. Electron. Agric.* **225**, 109342 (2024).
11. Ranasinghe, Y., Nair, N. G., Bandara, W. G. C. & Patel, V. M. Diffuse-denoise-count: accurate crowd-counting with diffusion models. arXiv preprint arXiv: 2303.12790 (2023).
12. Yuan, M., Wang, Y. & Wei, X. Translation, scale and rotation: cross-modal alignment meets rgb-infrared vehicle detection. In *European Conference on Computer Vision* 509–525 (Springer, 2022).
13. Wang, H. et al. Hierarchical kernel interaction network for remote sensing object counting. In *IEEE Transactions on Geoscience and Remote Sensing* (2024).
14. Jiang, S., Wang, Q., Cheng, F., Qi, Y. & Liu, Q. A unified object counting network with object occupation prior. In *IEEE Transactions on Circuits and Systems for Video Technology* (2023).
15. Lin, J.-P. & Sun, M.-T. A yolo-based traffic counting system. In *2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI)* 82–85 (IEEE, 2018).
16. Liu, W. et al. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14* 21–37 (Springer, 2016).
17. Li, Z. et al. Real-time detection and counting of wheat ears based on improved yolov7. *Comput. Electron. Agric.* **218**, 108670 (2024).
18. Bello, R.-W. & Oladipo, M. A. Mask yolov7-based drone vision system for automated cattle detection and counting. In *Mask YOLOv7-Based Drone Vision System for Automated Cattle Detection and Counting. Artificial Intelligence and Applications* (eds. Bello, R.-W. & Oladipo, M. A.). https://doi.org/10.47852/bonviewAIA42021603 (2024).
19. Cao, Y., Chen, J. & Zhang, Z. A sheep dynamic counting scheme based on the fusion between an improved-sparrow-search yolov5x-eca model and few-shot deepsort algorithm. *Comput. Electron. Agric.* **206**, 107696 (2023).
20. Sangaiah, A. K. et al. R-uav-net: Enhanced yolov4 with graph-semantic compression for transformative uav sensing in paddy agronomy. *IEEE Trans. Cogn. Commun. Netw.* **2024**, 1–1. https://doi.org/10.1109/TCCN.2024.3452053 (2024).
21. Anandakrishnan, J. et al. Precise spatial prediction of rice seedlings from large-scale airborne remote sensing data using optimized li-yolov9. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **18**, 2226–2238. https://doi.org/10.1109/JSTARS.2024.3505964 (2025).
22. Yang, Z. et al. Enhanced recognition and counting of high-coverage amorphophallus konjac by integrating uav rgb imagery and deep learning. *Sci. Rep.* **15**, 6501 (2025).
23. Tan, S., Ma, X., Mai, Z., Qi, L. & Wang, Y. Segmentation and counting algorithm for touching hybrid rice grains. *Comput. Electron. Agric.* **162**, 493–504 (2019).
24. Nguyen, H.-T., Ngo, C.-W. & Chan, W.-K. Sibnet: food instance counting and segmentation. *Pattern Recogn.* **124**, 108470 (2022).
25. Nguyen, H.-T., Cao, Y., Ngo, C.-W. & Chan, W.-K. Foodmask: real-time food instance counting, segmentation and recognition. *Pattern Recogn.* **146**, 110017 (2024).
26. Rong, W. et al. High-density pig herd counting method combined with feature pyramid and deformable convolution. *Trans. Chin. Soc. Agric. Mach.* **53**, 252. https://doi.org/10.6041/j.issn.1000-1298.2022.10.027 (2022).
27. Liu, S. et al. Icnet: a dual-branch instance segmentation network for high-precision pig counting. *Agriculture* **14**, 141 (2024).
28. Chen, J. & Wang, Z. Crowd counting with segmentation attention convolutional neural network. *IET Image Proc.* **15**, 1221–1231 (2021).
29. Dolezel, P. et al. Counting livestock with image segmentation neural network. In *15th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2020) 15* 237–244 (Springer, 2021).
30. Zhang, Y., Zhou, D., Chen, S., Gao, S. & Ma, Y. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 589–597 (2016).
31. Boominathan, L., Kruthiventi, S. S. & Babu, R. V. Crowdnet: a deep convolutional network for dense crowd counting. In *Proceedings of the 24th ACM international conference on Multimedia* 640–644 (2016).
32. Gao, G. et al. Psgcnet: a pyramidal scale and global context guided network for dense object counting in remote-sensing images. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–12 (2022).
33. Yi, J., Pang, Y., Zhou, W., Zhao, M. & Zheng, F. A perspective-embedded scale-selection network for crowd counting in public transportation. *IEEE Trans. Intell. Transport. Syst.* (2023).
34. Yang, S., Guo, W. & Ren, Y. Crowdformer: an overlap patching vision transformer for top-down crowd counting. *IJCAI* **1**, 2 (2022).
35. Yu, J. & Hu, H. Multiscale regional calibration network for crowd counting. *Sci. Rep.* **15**, 2866 (2025).
36. Lin, H., Ma, Z., Ji, R., Wang, Y. & Hong, X. Boosting crowd counting via multifaceted attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 19628–19637 (2022).
37. Zang, H. et al. Automatic detection and counting of wheat spike based on dmseg-count. *Sci. Rep.* **14**, 29676 (2024).
38. Chen, H., Ren, J., Sun, W., Hou, J. & Miao, Z. Mosquito swarm counting via attention-based multi-scale convolutional neural network. *Sci. Rep.* **13**, 4215 (2023).
39. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv: 1409.1556 (2014).
40. Gao, J., Gong, M. & Li, X. Congested crowd instance localization with dilated convolutional swin transformer. *Neurocomputing* **513**, 94–103 (2022).
41. Huang, Z.-K., Chen, W.-T., Chiang, Y.-C., Kuo, S.-Y. & Yang, M.-H. Counting crowds in bad weather. arXiv preprint arXiv: 2306.01209 (2023).
42. Wu, Z. et al. Cranet: cascade residual attention network for crowd counting. In *2021 IEEE International Conference on Multimedia and Expo (ICME)* 1–6 (IEEE, 2021).
43. Xu, J., Liu, W., Qin, Y. & Xu, G. Sheep counting method based on multiscale module deep neural network. *IEEE Access* **10**, 128293–128303 (2022).
44. Qin, X., Wang, Z., Bai, Y., Xie, X. & Jia, H. Ffa-net: feature fusion attention network for single image dehazing. ArXiv arXIv: abs/1911.07559 (2019).
45. Ma, Z., Wei, X., Hong, X. & Gong, Y. Bayesian loss for crowd count estimation with point supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* 6142–6151 (2019).
46. Kingma, D. P. Adam: a method for stochastic optimization. arXiv preprint arXiv: 1412.6980 (2014).
47. Zhu, L. et al. Dual path multi-scale fusion networks with attention for crowd counting. arXiv preprint arXiv: 1902.01115 (2019).
48. Gao, J., Han, T., Wang, Q., Yuan, Y. & Li, X. Learning independent instance maps for crowd localization. arXiv preprint arXiv: 2012.04164 (2020).
49. Han, T., Bai, L., Liu, L. & Ouyang, W. Steerer: Resolving scale variations for counting and localization via selective inheritance learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* 21848–21859 (2023).
50. Hou, Q., Zhou, D. & Feng, J. Coordinate attention for efficient mobile network design. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 13713–13722 (2021).
51. Wang, Q. et al. Eca-net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 11534–11542 (2020).
52. Ho, J., Kalchbrenner, N., Weissenborn, D. & Salimans, T. Axial attention in multidimensional transformers. arXiv preprint arXiv: 1912.12180 (2019).
53. Li, Y., Yao, T., Pan, Y. & Mei, T. Contextual transformer networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**, 1489–1500 (2022).

54. Pan, X. et al. On the integration of self-attention and convolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* 815–825 (2022).
55. Woo, S., Park, J., Lee, J.-Y. & Kweon, I. S. Cbam: convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)* 3–19 (2018).
56. Liu, Z. et al. A convnet for the 2020s. arXiv: 2201.03545 (2022)
57. Chen, J. et al. Run, don't walk: chasing higher flops for faster neural networks. arXiv: 2303.03667 (2023).
58. Ma, X., Dai, X., Bai, Y., Wang, Y. & Fu, Y. Rewrite the stars. arXiv:2403.19967 (2024).
59. Theart, R. & Schreve, K. UAV footage of Merino sheep in South Africa.https://doi.org/10.17632/nppwbkdf85.1 (2024).

Acknowledgements
This research is supported by National Key Research and Development Program of China (2021YFD1300502), and the Beijing Nova Program (2022114), and Special Project for the Construction of Scientific and Technological Innovation Capacity of Beijing Academy of Agriculture and Forestry Sciences (KJCX20251301).

Author contributions
Y.C. was responsible for the design of the entire network, the collection and processing of datasets, the execution of experiments, and the writing of the entire article. R.G., Q.L., H.Z. and L.D. were responsible for designing the initial thesis proposal and for the subsequent revision and refinement of the thesis. R.W. and X.L. assisted with data collection and preparation, as well as contributing to some of the experiments.

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to R.G.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.