Computers and Electronics in Agriculture 193 (2022) 106636

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

An intelligent method for dairy goat tracking based on Siamese network

Qingguo Su [a], Jinglei Tang [a,b,c,*], Mingxin Zhai [a], Dongjian He [b,c]

[a] College of Information Engineering, Northwest A&F University, Yangling 712100, Shaanxi, China
[b] The Key Laboratory of Agricultural Internet of Things, Ministry of Agriculture, Yangling 712100, Shaanxi, China
[c] Shaanxi Key Laboratory of Agricultural Information Perception and Intelligent Service, Yangling 712100, Shaanxi, China

ARTICLE INFO

Keywords:
Livestock tracking
Siamese network
Efficient network
Target interference
Attention mechanism

ABSTRACT

Tracking livestock can obtain their behavioral information, position information, activity data, and health status. Research on a robust, high-precision, non-contact, and real-time visual object tracking algorithm has important practical value for the management of livestock farming. Since dairy goats are collectively raised livestock, the difficulty in accurately monitoring large-scale goats farms lies in automatically tracking the individual. In this study, a novel Siamese Network Guided by Attention Mechanism (AMTracker) was proposed to track single dairy goat in the real farm scene. First of all, through analysis, we found that there were lots of similar goats raised in the same sheepfold, which led to similarity interference challenge when tracking someone. Secondly, the EfficientNet was employed to map features and the BiFPN model was employed to fuse the features at different levels. Next, we introduced the attention mechanism to improve the correlation between template frame and search frame to deal with the problem of similar target interference. Finally, we used an Anchor-free network to predict the position of the dairy goat in the search frame. The experimental results showed that AMTracker was superior to four state-of-the-art methods in terms of four evaluation indicators, and the Expected Average Overlap (EAO), Robustness (R), Precision (Prec) and Success (Succ) of AMTracker was 0.340, 0.455, 0.835 and 0.657, respectively. The tracker ran in a real-time manner with an average analysis speed of 30 *fps*. Hence, it was demonstrated that the proposed approach could offer one effective way for automatically tracking a dairy goat in real farms with complex conditions.

1. Introduction

The market demand for meat and dairy products continues to grow, which has led to a rapid increase in the scale of livestock farming such as dairy goats, pigs, and cows Jiang et al. (2020a); Zheng et al. (2020). As the scale of dairy goat farming continues to expand, the potentially diseased or vulnerable individual in group-bred dairy goats is increasing. The livestock industry and the public is increasingly concerned about the production efficiency and animal welfare of the livestock (Liu et al., 2020; Rachel et al., 2018). Tracking offers the possibility of collecting a large amount of quantitative data in terms of physiology, health, or behavior (Nasirahmadi et al., 2017). Timely and effective tracking of an abnormal individual can effectively assess movement status, health status, and social interaction, which has positive significance for disease prediction, prevention, and treatment. Therefore, research on real-time, reliable, and automated tracking algorithms for a single dairy goat is of great significance to breeding management and animal welfare.

As shown in Table 1, there are currently three ways to track the dairy goat and record information. The first method is manual observation (Zheng et al., 2018), the second method is to use wearable sensors (Kolarevic et al., 2016; Sakai et al., 2019), and the third method is to employ computer vision technology (Ahrendt et al., 2011; Jiang et al., 2020a). However, manual observation relies to a large extent on the prior experience of the staff and the cost increase with the expansion of the scale of farming. It is no longer possible for managers to track all animals in a reliable way with manual observation in large-scale farms. The sensors are uncomfortable and expensive, which is not conducive to the application in commercial farms. In contrast, computer vision technology provides an automated, real-time, animal-friendly and cost-effective way to track and monitor animals.

In recent years, researchers have become increasingly interested in using computer vision technology to detect and track animals in video sequences. Vayssade et al. (2019) utilized a combination of an adaptive threshold method and the LDA algorithm to automatically detect goats and track their activities from images taken by commercial drones. In

* Corresponding author at: College of Information Engineering, Northwest A&F University, Yangling 712100, Shaanxi, China.
E-mail address: tangjinglei@nwsuaf.edu.cn (J. Tang).

https://doi.org/10.1016/j.compag.2021.106636
Received 12 April 2021; Received in revised form 25 November 2021; Accepted 15 December 2021
Available online 24 December 2021

Table 1
Comparison of three livestock tracking strategies.

	Cost-effective	Comfortable	Real-time	High accuracy
By manual observation		✓	✓	×
By wearable sensors	×	×	✓	✓
By computer vision	✓	✓	✓	✓

order to monitor the growth and development, production performance, and genetic characteristics of sheep, Zhang et al. (2018) proposed a method to measure the body size of the sheep based on visual image analysis. Compared with manual observation, this method can reduce the stress response of sheep. In the context of precision animal husbandry, Tang et al. (2019) provided a significant target detection method based on background and foreground priors to automatically monitor the population size of dairy goats in the farm. The goat's behavior monitoring mode during estrus based on visual tracking analysis was studied by (Endo et al., 2016) who tracked the goat's position and movement in a pen with a size of $2.5m \times 2.5m$ and recorded trajectory data for behavior analysis. To solve the problem of inefficient and low accuracy in monitoring dairy goats, Wang et al. (2018a) proposed a target detection method based on Faster R-CNN, which is more than twice as fast as the original Faster R-CNN (Ren et al., 2017). Jiang et al. (2020a) studied the goat individual detection model based on deep learning and designed a goat behavior recognition algorithm based on temporal and spatial location features. At the same time, some work based on computer vision technology has studied other types of livestock. To accurately monitor and evaluate the abnormal behavior of poultry in large poultry farms, (Fang et al., 2020) proposed a deep regression network based on Alexnet and compares it with existing tracking algorithms. Chen et al. (2020) combined CNN and Long Short-term Memory (LSTM) to extract Spatio-temporal features to monitor the aggressive behavior of pigs. In recent research, to quickly and accurately perceive the behaviors of cows, Jiang et al. (2020b) proposed a cow's behavior recognition algorithm based on the EfficientNet-LSTM, which achieves an accuracy of 97.87%. The above-mentioned deep learning-based livestock tracking and monitoring algorithms have significant advantages because they can map semantic information as target representations in a non-contact and stress-free manner.

However, there are still common limitations of the above-mentioned algorithms. Specifically, it is difficult for these methods to track an object in some actual scenes. By analyzing the movement characteristics of dairy goats, we found that the interference of similar objects and scale variation of the selected object in the group confinement scene poses a huge challenge to track the single individual. Due to these complexities, detections are always inevitably lost, which increases the risk of the tracker missing the target or tracking the target incorrectly (Liu et al., 2020). To overcome these difficulties, we proposed a dairy goat tracking framework that accurately tracks the selected individual in real-time from the dairy goat motion video acquired by the camera. Recently, the Siamese network has attracted great attention in the field of visual tracking because of its good balance between accuracy and speed (Li et al., 2019; Li et al., 2018). First, we used the Siamese network as the structure of the tracker. The visual object tracking problem was formulated as learning a general similarity map through the cross-correlation between the feature representations learned for the target template and the search region. Second, the lightweight EfficientNet (Tan and Le, 2019) was adopted as the backbone network to learn feature representations at different levels, and these feature maps were fused by the Bi-directional Feature Pyramid Network (BiFPN) (Tan et al., 2020). Low-level features help locate targets and high-level semantic information can help distinguish targets. The BiFPN introduces weights to learn the importance of different features while fusing low-level features with high-level features to enhance the robustness of features. Next, the introduced Attention Module learns the correlation between the strong search image and the target template by cross attention mechanism and employs a self-spatial attention mechanism to learn strong context information, thereby enhancing the tracker's ability to deal with similarity interference. Moreover, to obtain a more accurate estimation of the position, the Anchor-free Network (Tian et al., 2019) is used to train the network. The Anchor-free Network effectively circumvents the shortcomings of the fixed anchor ratio, which views each pixel as an anchor point to directly predict the position of the object, so that the predicted bounding box is closer to the ground-truth. In addition, we use the Cross-entropy loss and the Distance-IoU (DIoU) loss to train the classification network and regression network, respectively. The DIoU directly minimizes the normalized distance between the center points of the two bounding boxes by adding a penalty term to the IoU loss. Experiments show that the proposed framework provides an effective method for real-time tracking and monitoring of the dairy goat from video sequences, which facilitates the improvement of farming management and the creation of livestock-friendly conditions.

In this paper, we make the following contributions:

(1) We propose a novel Siamese Network Guided by Attention Mechanism (AMTracker) to track a single dairy goat in a real farm scene. The AMTracker employs EfficientNet as the backbone network and uses BiFPN to integrate multi-level features.
(2) The cross attention is used to establish an interactive bridge between the target template and the search framework to share the same channel weight. As a supplement to cross attention, self-spatial attention is used to enhance the discriminability of related feature maps. At the same time, an Anchor-free network is used to obtain the accurate bounding box without being negatively affected by changes in target scale.
(3) The experimental results showed that AMTracker was superior to five state-of-the-art methods with an average analysis speed of 30 *fps*. The proposed method can accurately track dairy goats in real farms with complex conditions.

In the remainder of this paper, we first introduce the materials and methods in Section. 2. Experiments and results are reported in Section. 3. We end the paper with a conclusion in Section. 4.

2. Materials and methods

2.1. Video and dataset acquisition

The experiments in this study were implemented in an outdoor sheep pen located at the Animal Husbandry Experimental Base of Northwest A&F University. As shown in Fig. 1, the layout of the outdoor sheep pen includes a food trough and a playground with a size of $15m^2$. The subjects of this study were the bred Saanen goats (age $\in$ [2yearsold, 5yearsold]) with different activity states. During the whole experiment,

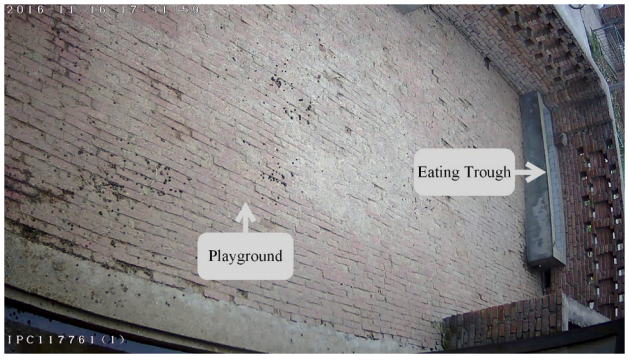

Fig. 1. Layout of the outdoor activity area for dairy goats.

the dairy goat could move around freely in the fence without external interference.

The camera was the DS-IPC-B12-I/POE (HANGZHOU HIKVISION DIGITAL TECHNOLOGY CO., LTD) which can capture the video with a solution of 1080P at a sample speed of 25 frames per second (*fps*) and can be connected to the computer through an internet cable. The camera was installed on the support wall of the pen at a height of 3 m and was kept tilted down about 60 degrees to ensure that the camera's field of view can monitor the entire sheepfold.

To evaluate the algorithm, 200 videos with a total of 161,200 frames of images were selected as the dataset. Each video in the dataset has a resolution of 1920 × 1080 pixels. We tried to collect various motion videos of dairy goats in different individuals and states, such as running, fighting, walking, and climbing, to ensure a more representative dataset. The light conditions of dark are ignored in this study. This study mainly focuses on the tracking of the head and the whole body of the single dairy goat. The ground-truth of the dairy goat in each video was labeled with a rectangular box. Table 2 shows the statistics of the dataset. The size of the dataset used to train the algorithm is 105 video sequences, and the remaining 95 video sequences are used to test the model. In detail, the size of the dataset used to train and test the model's ability to track the head position is 35 videos and 30 videos, respectively. The number of videos used to train and test the whole body tracking ability is 70 and 65, respectively.

2.2. Analysis of the dairy goat

Through observation, we found that the dairy goats raised in the same sheep pen of a commercial farm are the same species with similar ages, body shapes and colors. Although the management staff printed different numbers on the back of the dairy goats for the convenience of statistics and identification, the cameras still could not capture these marks due to the fixed field of view and the activity of the goats. (an example is shown in Fig. 2**(a)**). At the same time, the scale of the dairy goat in the video frame changes with its movement (an example is shown in Fig. 2**(b)**). Factors, such as similarity interference and scale variation can deteriorate the tracking performance (Zhu et al., 2018). Therefore, it is necessary to solve these problems in order to track a single dairy goat robustly and accurately in a real farm scene.

2.3. Single dairy goat tracking network

2.3.1. The overall framework of the tracker

To achieve robust and real-time tracking of a single dairy goat in outdoor captivity scenarios, we proposed an attention-guided dairy goat tracking algorithm. The basic idea of the proposed method is that the tracking task was regarded as detecting the selected dairy goat designated in the first frame in each frame of the video. The technical route adopted in this research is shown in Fig. 3 and **Algorithm 1**.

Algorithm 1 AMTracker

Input: Video sequence × , template **z**
Output: Bounding box (x, y, w, h)
1: $P_j(\mathbf{z}) \leftarrow$ extraction features of **z** by the EfficientNet, $j \in [3,7]$
2: $p(\mathbf{z}) \leftarrow$ fusion $P_j(\mathbf{z})$ by the BiFPN
3: $p(\mathbf{z})^A \leftarrow$ obtained by the Attention Model

(*continued on next column*)

Table 2
Statistics and distribution of the dataset.

Position	Train		Test	
	Number of videos	Number of frames	Number of videos	Number of frames
Head	35	26,312	30	19,603
Body	70	62,008	65	53,277
Total	105	88,320	95	72,880

(*continued*)

Algorithm 1 AMTracker

4: **for** $i \leftarrow 1$ **to** len(**x**) **do**
5: $P_j(\mathbf{x}_i) \leftarrow$ extraction features of $\mathbf{x}_i$ by the EfficientNet, $j \in [3,7]$
6: $p(\mathbf{x}_i) \leftarrow$ fusion $P_j(\mathbf{x}_i)$ by the BiFPN
7: $p(\mathbf{x}_i)^A \leftarrow$ obtained by the Attention Model
8: The correlation feature map $F_i \leftarrow p(\mathbf{z})^A \odot p(\mathbf{x}_i)^A$
9: (x, y, w, h) in $\mathbf{x}_i \leftarrow$ classification and regression through F_i by the Anchor-free Network

The template **z** of dairy goat selected manually in the first frame of the video is resized to 127 × 127 × 3, and each frame of the video sequence is resized to 255 × 255 × 3 as the search region $\mathbf{x}_i$, $i \in [1, n]$, n denotes the number of frames of the current video.

EfficientNet is employed to convert the **z** and $\mathbf{x}_i$ into multi-level discriminative features. **z** is calculated only once. $P_j(\mathbf{z})$ and $P_j(\mathbf{x}_i)$ are used to represent the 3–7 layer features of the template and search region respectively, where $i \in [3,7]$.

The feature maps of 3–7 levels of $P_j(\mathbf{z})$ and $P_j(\mathbf{x}_i)$ are respectively fused by the BiFPN Modules. Its output is expressed as $p(\mathbf{z})$ and $p(\mathbf{x}_i)$. The $p(\mathbf{z})^A$ and $p(\mathbf{x}_i)^A$ are obtained by the Attention Model which consists of self-spatial attention for learning the global context and cross attention for exploring the interdependencies between the $p(\mathbf{z})$ and $p(\mathbf{x}_i)$.

The correlation feature map F_i is calculated by the depth-wise correlation operation between $p(\mathbf{z})^A$ and $p(\mathbf{x}_i)^A$, i.e.,

$$F_i = p(\mathbf{z})^A \odot p(\mathbf{x}_i)^A \tag{1}$$

The Anchor-free Network is used to predict the bounding box of the dairy goat in the search region $\mathbf{x}_i$.

Repeat **Step 2** to **Step 6** unstil $i = n$.

2.3.2. Siamese Network for Tracking

As often done in the community, this paper introduces the Siamese network for visual object tracking, which views visual object tracking as a similarity learning problem. Concretely speaking, the Siamese network consists of a pair of CNN backbones with sharing parameters p, which are used to embed the target image (**z**) and the search image (**x**) into a common feature space. A similarity metric F can be computed to measure the similarities between them,

$$F(\mathbf{z}, \mathbf{x}) = P(\mathbf{z}) \odot P(\mathbf{x}) + \mathbb{b} \tag{2}$$

where P denotes the output feature maps of the Siamese network, $\mathbb{b}$ is the bias.

2.3.3. Extract multi-level features from the EfficientNet

In the task of target detection, researchers increasingly tend to obtain a single high-level feature by deepening the backbone network. The inherent defect is that the feature has less pixel information, which makes it impossible to effectively distinguish the selected object from similar targets. The low-level feature maps have higher resolution, contain more location and detailed information, which is beneficial for locating the target, and the high-level feature maps have stronger semantic information, which is helpful for foreground and background classification (Li et al., 2017). So, it is necessary to use multi-level features to accurately track the selected dairy goat in similar target groups. At the same time, the computing power of the computer hardware of commercial farms is weak, large-scale and expensive computing costs will limit the application of these algorithm models in real farms due to delays and resource constraints. Therefore, it is necessary to choose a CNN with low computational complexity and fast speed.

EfficientNet is a new type of CNN network with extremely high parameter efficiency and speed (Yin et al., 2020). The network uses a

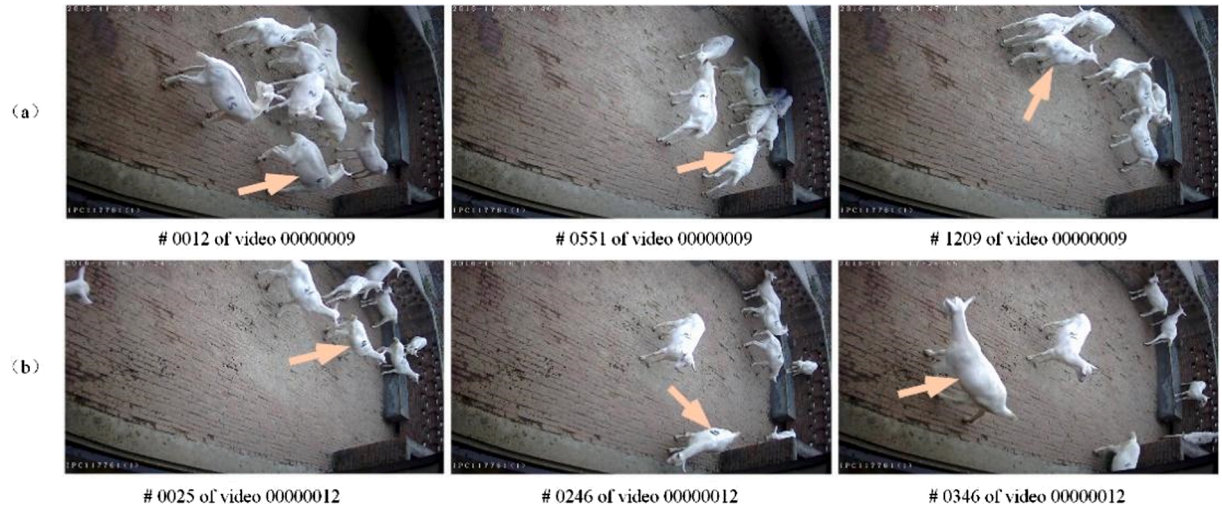

Fig. 2. Illustration of the difficulty of tracking in a real farm scene. (a) shows the interference of similar objects in the background. (b) shows the same object has inconsistent scales in different frames.

new model composite scaling method, which utilizes simple and efficient composite coefficients to weigh the network depth, width, and input image resolution, thereby reducing the number of parameters and the floating point operations (FLOPs) by an order of magnitude. We use EfficientNet-B0 to extract features from P3 to P7 layers. As shown in Fig. 4, the low-level feature maps $P3-P4$ contain information such as the contour and position of the dairy goat. As the network deepens, the feature maps include more discriminative semantic information. In this study, we made minor changes to EfficientNet to make it more suitable for tracking dairy goats. Adjusting the size of the template image $(\mathbf{z})$ to $127 \times 127 \times 3$ and inputting it to the template branch. Similarly, adjusting the size of the search region image $(\mathbf{x}_i)$ to $255 \times 255 \times 3$ and inputting it to the search branch. The $P3$ - $P7$ feature maps with the size of $64 \times 64 \times 24$, $32 \times 32 \times 40$, $16 \times 16 \times 80$, $16 \times 16 \times 112$, and $8 \times 8 \times 160$ are extracted by the EfficientNet in template branch and the $P3-P7$ feature maps with the size of $127 \times 127 \times 24$, $64 \times 64 \times 40$, $32 \times 32 \times 80$, $32 \times 32 \times 112$, and $16 \times 16 \times 160$ are extracted by the EfficientNet in search branch. After that, the feature maps are sent to the BiFPN Modules for a fusion of multi-level feature maps.

In order to strengthen the robustness of object representation, the BiFPN Module is employed to fuse the feature maps of $P3$ - $P7$. The fusion of multi-level feature maps can aggregate features of different resolution scales, which is conducive to accurately tracking the individual in complex farming scenes. The BiFPN Module consists of the Multi-level Connection and the Weighted Feature Fusion.

(1) Multi-level Connection. As shown in Fig. 5**(a)**. The Multi-level Connection includes a top-down connection and a bottom-up connection. The top-down connection is from P7 to P3, and its purpose is to retain semantic information that helps classification. It employs upsampling to scale up the small feature map of the high-level layer to the same size as the feature map of the low-level and merge it with the low-level feature map. The bottom-up connection is to sequentially downsample and merge from P3 to P7. Its function is to use accurate low-level positioning signals to improve the entire feature layers, thereby shortening the information path from low-level to high-level. By adopting top-down and bottom-up paths, the bi-directional feature fusion network is formed. Since a node has only one input edge and no feature fusion, it contributes little to the feature network that aims to aggregate different features. Therefore, the middle nodes of $P3$ and $P7$ in Fig. 5**(b)** are removed from the module to simplify the structure.

(2) Weighted Feature Fusion. Since feature maps of different level have different resolutions, they usually contribute unevenly to output features. The fast normalized fusion is employed to add extra weight to each input and learn the importance of each input feature, i.e.,

$$O = \sum_m \frac{w_m}{\epsilon + \sum_n w_n} \cdot \mathrm{P_m} \tag{3}$$

where w is a learnable weight and $\epsilon = 0.0001$.

In detail, we take the template branch as an example to introduce the specific connection method. As shown in Fig. 6, the feature map of $P6^{st} \in \mathbb{R}^{16\times16\times112}$ is obtained by the addition between $P6^{in} \in \mathbb{R}^{16\times16\times112}$ and $Resize_{up}\left(P7^{in}\right) \in \mathbb{R}^{16\times16\times112}$ and then convolving with a kernel of $1 \times 1 \times 112$. The feature map of $P5^{st} \in \mathbb{R}^{16\times16\times80}$ is obtained by the addition between $P5^{in} \in \mathbb{R}^{16\times16\times80}$ and $Conv(P6^{st}) \in \mathbb{R}^{16\times16\times80}$ and then convolving with a kernel of $1 \times 1 \times 80$. The feature map of $P4^{st} \in \mathbb{R}^{32\times32\times40}$ is obtained by the addition between $P4^{in} \in \mathbb{R}^{32\times32\times40}$ and $Resize_{up}\left(P5^{in}\right) \in \mathbb{R}^{32\times32\times40}$ and then convolving with a kernel of $1 \times 1 \times 40$. As shown in the following formula.Fig. 7.

$$\begin{cases} P6^{st} = Conv\left(P6^{in} + Reize_{up}\left(P7^{in}\right)\right) \\ P5^{st} = Conv\left(P5^{in} + Conv(P6^{st})\right) \\ P4^{st} = Conv\left(P4^{in} + Reize_{up}(P5^{st})\right) \end{cases} \tag{4}$$

Where the $Conv$ denotes a convolution operation, $Resize_{up}$ denotes upsampling.

The feature map of $P3^{out} \in \mathbb{R}^{64\times64\times80}$ is obtained by the addition between $P3^{in} \in \mathbb{R}^{64\times64\times24}$ and $Resize_{up}(P4^{st}) \in \mathbb{R}^{64\times64\times24}$ and then convolving with a kernel of $1 \times 1 \times 80$. The feature map of $P4^{out} \in \mathbb{R}^{32\times32\times80}$ is obtained by the addition between $P4^{in}$, $P4^{st}$ and $Resize_{down}(P3^{out}) \in \mathbb{R}^{64\times64\times40}$ and then convolving with a kernel of $1 \times 1 \times 80$. The feature map of $P5^{out} \in \mathbb{R}^{16\times16\times80}$ is obtained by the addition between $P5^{in}$, $P5^{st}$ and $Resize_{down}(P4^{out}) \in \mathbb{R}^{32\times32\times80}$ and then convolving with a kernel of $1 \times 1 \times 80$. The feature map of $P6^{out} \in \mathbb{R}^{16\times16\times80}$ is obtained by the addition between $P6^{in}$, $P6^{st}$ and $P5^{out}$ and then convolving with a kernel of $1 \times 1 \times 80$. The feature map of $P7^{out} \in \mathbb{R}^{8\times8\times80}$ is obtained by the addition between $P7^{in}$ and $Resize_{down}(P6^{out}) \in \mathbb{R}^{16\times16\times24}$ and then convolving with a kernel of $1 \times 1 \times 80$. As shown in the following formula.

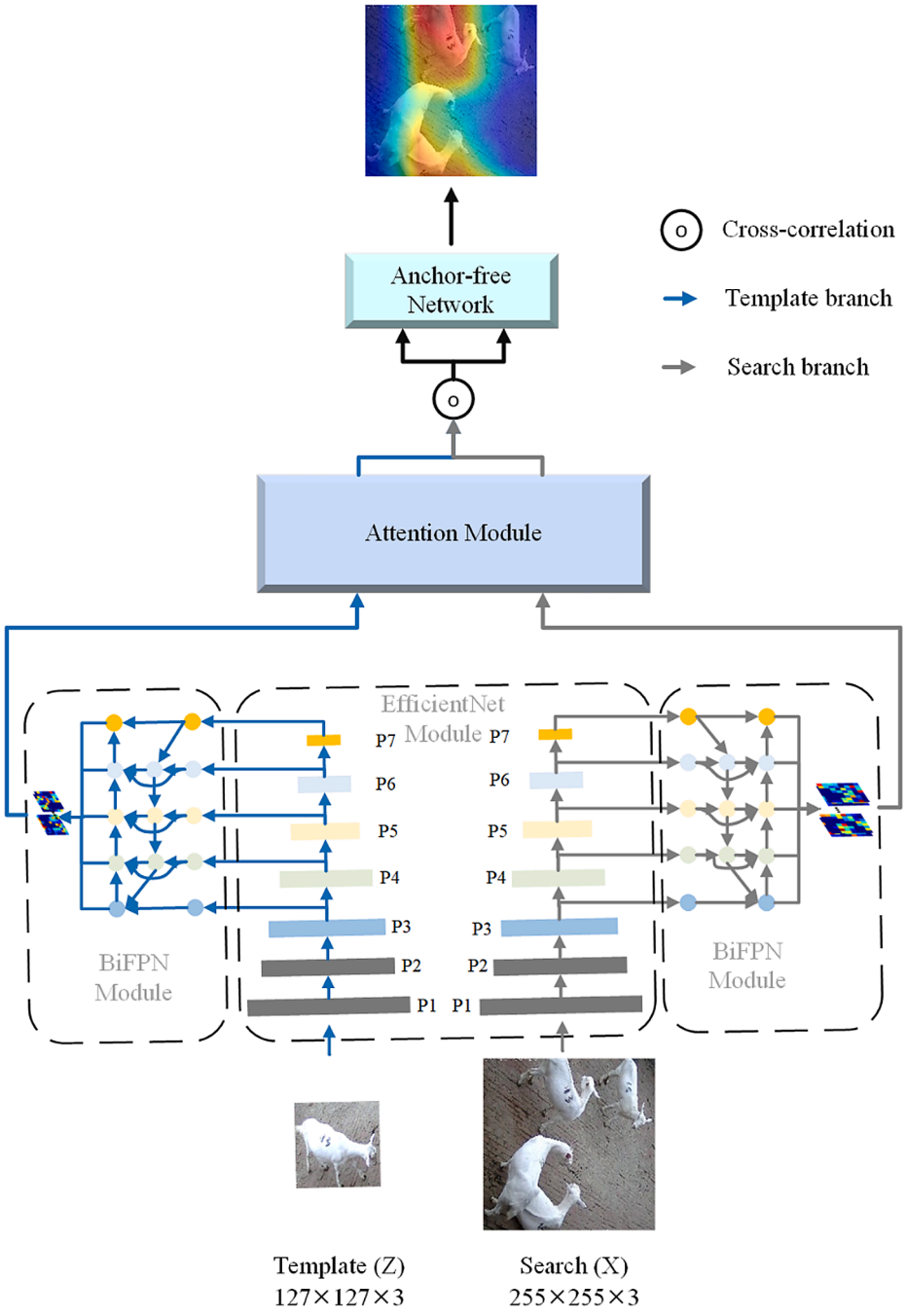

Fig. 3. The overview of the AMTracker. It consists of two EfficientNet Modules, two BiFPN Modules, an Attention Module, a Depth-wise Correlation Layer and an Anchor-free Network.

$$\begin{cases} P3^{out} = Conv\left(P3^{in} + Resize_{up}\left(P4^{st}\right)\right) \\ P4^{out} = Conv\left(P4^{in} + P4^{st} + Resize_{down}\left(P3^{out}\right)\right) \\ P5^{out} = Conv\left(P5^{in} + P5^{st} + Resize_{down}\left(P4^{out}\right)\right) \\ P6^{out} = Conv\left(P6^{in} + P6^{st} + P5^{out}\right) \\ P7^{out} = Conv\left(P7^{in} + Resize_{down}\left(P6^{out}\right)\right) \end{cases} \tag{5}$$

Combining with Equation (3), the feature maps of different levels are constructed by the fast normalized fusion. As an example, here we describe the two-level fused features of *P*6 in Fig. 6:

$$\begin{cases} P6^{st} = Conv\left(\dfrac{w_1 \cdot P6^{in} + w_2 \cdot Reize_{up}\left(P7^{in}\right)}{w_1 + w_2 + \varepsilon}\right) \\ P6^{out} = Conv\left(\dfrac{w_1^{'} \cdot P6^{in} + w_2^{'} \cdot P6^{st} + w_3^{'} \cdot P5^{out}}{w_1^{'} + w_2^{'} + w_3^{'} + \varepsilon}\right) \end{cases} \tag{6}$$

All other features, including the features of the Search branch, are constructed similarly. Finally, the features of the 3–7 layers are sent to the fusion layer and combined with the up-sampling and down-sampling

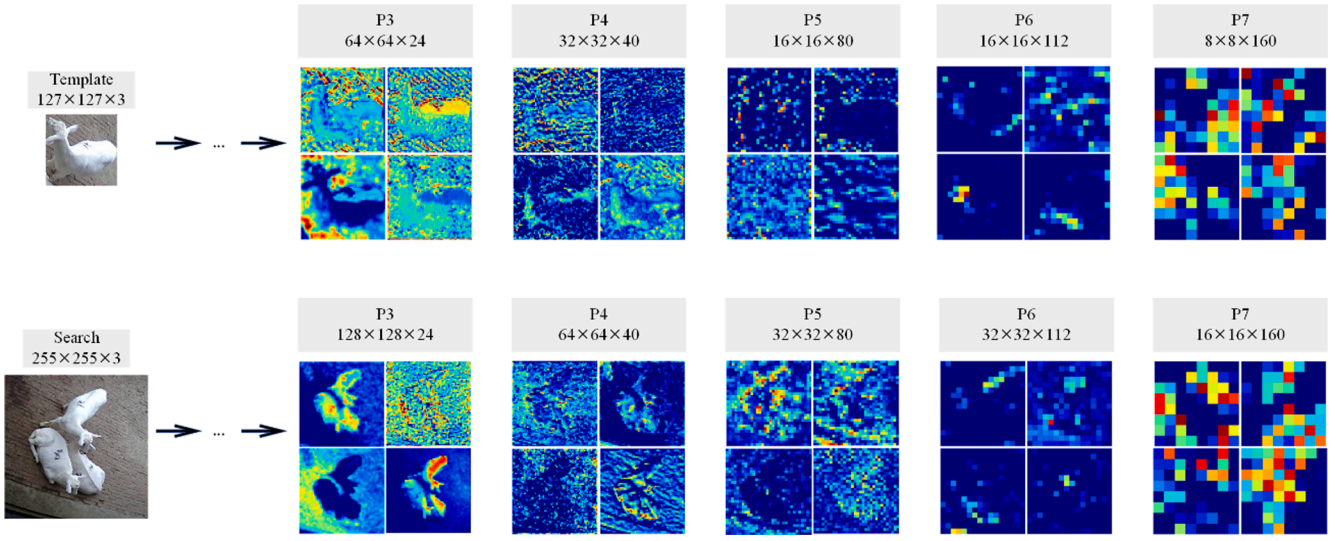

Fig. 4. Feature maps of 3–7 layers output by the EfficientNet.

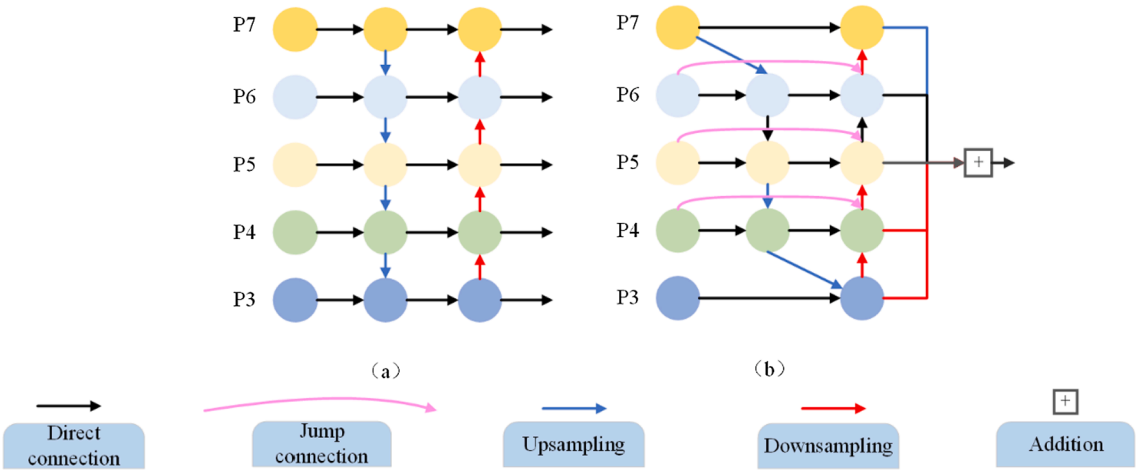

Fig. 5. (a) PANet with the bi-directional feature fusion network, (b) BiFPN with simplified bi-directional feature fusion network.

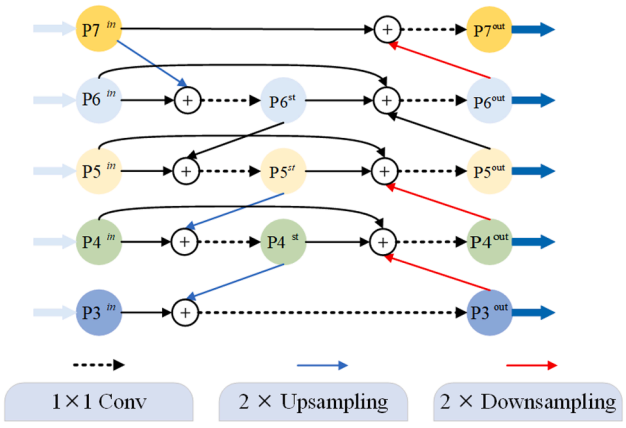

Fig. 6. Detailed structure of the BiFPN.

to fuse the final features $p(\mathbf{z}) \in \mathbb{R}^{16\times16\times80}$ or $p(\mathbf{x}_i) = p(\mathbf{z}) \in \mathbb{R}^{32\times32\times80}$ to obtain a rich combination of low level, middle level and deep level features, which is helpful for tracking the dairy goat accurately and robustly.

Strengthening the features guided by the Attention Module

The purpose of creating the Attention Module is to focus on the special parts which are of great importance, i.e., channel information and spatial information. In the process of tracking a dairy goat, since the template branch and the search branch are calculated independently, the model is difficult to deal with the interference of similar objects. So, it is crucial that the Search branch and the Template branch learn related information through cross attention, which enables it to generate a more discriminative representation that helps to detect the target more accurately (Yu et al., 2020). At the same time, when the feature information from the Search branch is encoded, the template representation can be more useful. Self-spatial attention tends to make CNN pay more attention to the global context so that the selected dairy goat can be distinguished from the background (Wang et al., 2018b). Inspired by (Yu et al., 2020), we add cross-attention and self-spatial attention to strengthen the features.

Specifically, the self-spatial attention is calculated separately on $p(\mathbf{z})$ and $p(\mathbf{x}_i)$. Taking the spatial self-attention of $p(\mathbf{z})$ for example:

Two separate convolution layers with 1×1 kernels are employed on $p(\mathbf{z})$ to generate key features $K_\mathbf{z} \in \mathbb{R}^{16\times16\times10}$ and query features $Q_\mathbf{z} \in \mathbb{R}^{16\times16\times10}$.

The key features $K_\mathbf{z}$ and query features $Q_\mathbf{z}$ are reshaped to $\overline{K}_\mathbf{z} \in \mathbb{R}^{(16\times16)\times10}$ and $\overline{Q}_\mathbf{z} \in \mathbb{R}^{(16\times16)\times10}$.

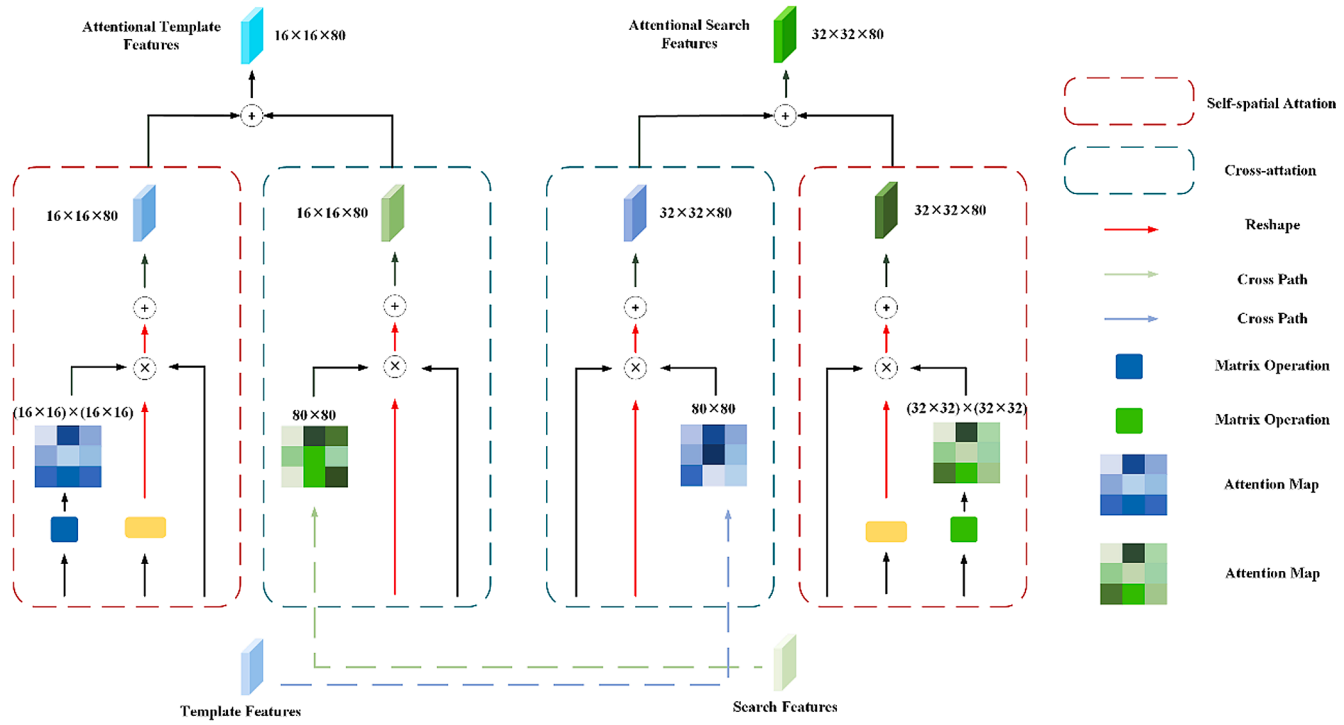

Fig. 7. Illustration of the Attention Module. It consists of self-spatial attention and cross-attention.

The self-spatial attention feature map $A_{\mathbf{z}}^{S} \in \mathbb{R}^{(16\times16)\times(16\times16)}$ is obtained by matrix multiplication and column-wise softmax operations between $\overline{K}_{\mathbf{z}}$ and $\overline{Q}_{\mathbf{z}}{}^{T}$ as,

$$A_{\mathbf{z}}^{S} = \mathrm{softmax}_{col}\left(\overline{Q}_{\mathbf{z}}{}^{T}\overline{K}_{\mathbf{z}}\right), A_{\mathbf{z}}^{S} \in \mathbb{R}^{(16\times16)\times(16\times16)} \tag{7}$$

The value features $\overline{V}_{\mathbf{z}} \in \mathbb{R}^{(16\times16)\times80}$ is obtained by applying a 1×1 convolution layer with a reshape operation to $p(\mathbf{z})$.

The $A_{\mathbf{z}}^{S}$ are multiplied with $\overline{V}_{z}$ and then are added to the reshaped $\overline{p(\mathbf{z})} \in \mathbb{R}^{(16\times16)\times80}$ with a residual connection as,

$$\overline{p(\mathbf{z})}_{\mathbf{z}}^{S} = \alpha\overline{V}_{\mathbf{z}}A_{\mathbf{z}}^{S} + \overline{p(\mathbf{z})}, \overline{p(\mathbf{z})}_{z}^{S} \in \mathbb{R}^{(16\times16)\times80} \tag{8}$$

where α is a scalar parameter. The $\overline{p(\mathbf{z})}_{\mathbf{z}}^{S}$ is reshaped to $p(\mathbf{z})_{\mathbf{z}}^{S} \in \mathbb{R}^{16\times16\times80}$.

The cross attention is obtained by both $p(\mathbf{z})$ and $p(\mathbf{x}_i)$. Take the cross-attention of the Template branch for example:

Step 1. The search features $p(\mathbf{x}_i)$ are reshaped to $\overline{p(\mathbf{x}_i)} \in \mathbb{R}^{(16\times16)\times80}$.

Step 2. The cross attention from the Search branch is computed by the equation as

$$A_{p(\mathbf{x}_i)}^{C} = \mathrm{softmax}_{row}\left(\overline{p(\mathbf{x}_i)}\overline{p(\mathbf{x}_i)}^{T}\right) \in \mathbb{R}^{80\times80} \tag{9}$$

where $\mathrm{softmax}_{row}$ is row-wise softmax operation.

Step 3. The $A_{p(\mathbf{x}_i)}^{C}$ is encoded into the template features $p(\mathbf{z})$ as

$$\overline{p(\mathbf{z})}_{\mathbf{z}}^{C} = \gamma A_{p(\mathbf{x}_i)}^{C}\overline{p(\mathbf{z})} + \overline{p(\mathbf{z})}, \overline{p(\mathbf{z})}_{\mathbf{z}}^{C} \in \mathbb{R}^{(16\times16)\times80} \tag{10}$$

where γ is a scalar parameter. The $\overline{p(\mathbf{z})}_{\mathbf{z}}^{C}$ is reshaped to $p(\mathbf{z})_{\mathbf{z}}^{C} \in \mathbb{R}^{16\times16\times80}$.

Step 4. The attentional feature map $p(\mathbf{z})^{A}$ of the object template is obtained by element-wise sum between $p(\mathbf{z})_{\mathbf{z}}^{C}$ and $p(z)_{\mathbf{z}}^{S}$. The attentional feature map of search images can be computed similarly. As shown in the following formula,

$$\begin{cases} p(\mathbf{z})^{A} = p(\mathbf{z})_{\mathbf{z}}^{C} + p(\mathbf{z})_{\mathbf{z}}^{S}, p(\mathbf{z})^{A} \in \mathbb{R}^{(16\times16)\times80} \\ p(\mathbf{x}_i)^{A} = p(\mathbf{x}_i)_{\mathbf{x}_i}^{C} + p(\mathbf{x}_i)_{\mathbf{x}_i}^{S}, p(\mathbf{x}_i)^{A} \in \mathbb{R}^{(16\times16)\times80} \end{cases} \tag{11}$$

2.3.4. Detecting the dairy goat by the Anchor-free Network

The anchor-free methods have become popular in object detection tasks because it does not require frequent adjustment of the anchor ratio according to the object scales (Duan et al., 2019). In order to solve the challenge of the scale ratio change of the dairy goat, this study introduces the Anchor-free Network to predict the bounding box. The core idea is to estimate the distance from the center of the target object to the four sides of the ground-truth box.

First, as shown in Equation (1), combining the extracted features $p(\mathbf{z})^{A}$ of the template image and that $p(\mathbf{x}_i)^{A}$ of search images by deep cross-correlation operation, and generating corresponding similarity features F_i for the subsequent target position.

Next, similar to (Tian et al., 2019), the Anchor-free Network shown in Fig. 8, has two branches, one for classification and the other for regression.

Then, let $B_i = \left\{x_i^{(0)}, y_i^{(0)}, x_i^{(1)}, y_i^{(1)}\right\} \in \mathbb{R}^4$ denotes the coordinates of the left-top and right-bottom corners of the bounding box in $\mathbf{x}_i$. The pixel on the $\mathbf{x}_i$ corresponds to a pixel on F_i is denoted as (x,y). If (x,y) falls into the ground-truth B_i, it is considered as a positive sample. Also, the pixel has a 4-D vector $\boldsymbol{t}^{*} = (l^{*}, t^{*}, r^{*}, b^{*})$ for regressing the position, and the $\boldsymbol{t}^{*}$ of samples can be formulated as

$$\begin{cases} l^{*} = x - x_0 \\ t^{*} = y - y_0 \\ r^{*} = x_1 - x \\ b^{*} = y_1 - y \end{cases} \tag{12}$$

where the l^{*}, t^{*}, r^{*} and b^{*} denotes the distance from the (x,y) to the four sides of the B_i.

Finally, as shown in Fig. 9, to limit the low-quality predicted bounding boxes produced by locations far away from the center of the dairy goat, the center-ness is used to describe the standardized distance from the (x,y) to the center of the dairy goat. As shown in Fig. 8, the

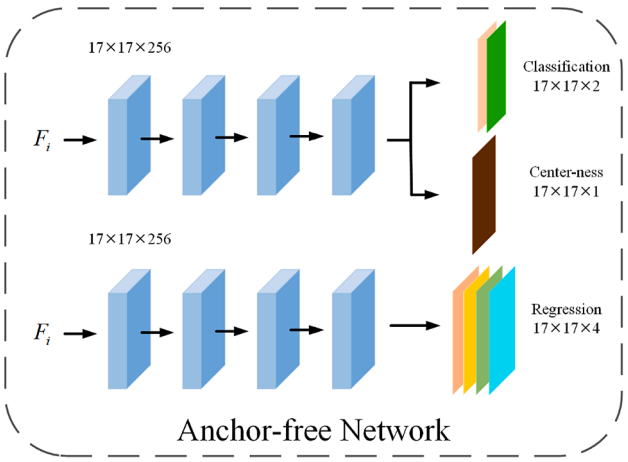

Fig. 8. Illustration of the Anchor-free Network. It consists of the classification branch and regression branch. A single-layer branch, in parallel with the classification branch to predict the "center-ness" of a location.

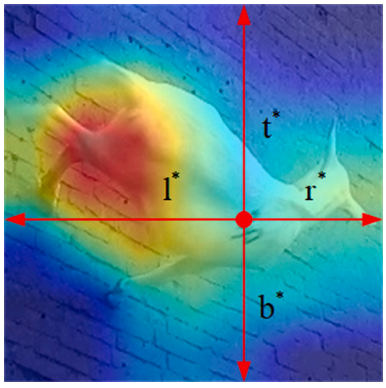

Fig. 9. Illustration of the Center-ness. Red, blue, and other colors represent 1, 0, and the values between them, respectively. The center-ness decays from 1 to 0 as the position deviates from the center of the object.

center-ness is predicted by a single-layer branch which is in parallel with the classification branch. The center-ness is defined as

$$centerness^{*} = \sqrt{\frac{min(l^{*}, r^{*})}{max(l^{*}, r^{*})} \times \frac{min(t^{*}, b^{*})}{max(t^{*}, b^{*})}} \tag{13}$$

where $centerness^{*} \in [0, 1]$ and decays from 1 to 0 as the position deviates from the center of the dairy goat. The final classification score is calculated by multiplying the center-ness and the classification score during the testing period. The Non-Maximum Suppression (NMS) (Neubeck and Gool, 2006) algorithm is employed to filter out the low-quality predicted bounding boxes, improving the tracking performance remarkably. In the end, the position of the predicted bounding box is the location.

3. Experiments and results

Algorithm 2 Training Scheme

1: Parameters $\wp$ pre-trained on ImageNet Dataset
2: EfficientNet Module initialized with$\wp$
3: Selecting **D** (105 goat videos and 510 videos from LaSOT, Youtube-BB and GOT-10 k)
4: **for** *video* ← 1 **to** 615 **do**
5: Cropping $\mathbf{z} \in \mathbb{R}^{127\times127\times3}$ and $\mathbf{x} \in \mathbb{R}^{255\times255\times3}$ of $\mathbf{D}_{video}$
6: **for** *epoch* ← 1 **to** 5 **do**
7: Freezing EfficientNet Module

(*continued on next column*)

(*continued*)

Algorithm 2 Training Scheme

8: **for** *video* ← 1 **to** 615 **do**
9: **for** *frame* ← 1 to len($\mathbf{D}_{video}$) do
10: Training AMtracker with z_{frame} and x_{frame}
batchsize = 32, weight decay=1×10^{-3},optimizing with SGD
learning rate=1×10^{-3}, momentum=0.9
11: **for** *epoch* ← 6 **to** 50 **do**
12: Unfreezing EfficientNet Module
13: **for** *video* ← 1 **to** 615 **do**
14: **for** *frame* ← 1 **to** len($\mathbf{D}_{video}$) **do**
15: Training AMtracker with z_{frame} and x_{frame}
batchsize = 32, weight decay=1×10^{-3}, optimizing with SGD
learning rate = $5 \times 10^{-3}$to 1×10^{-5}, momentum=0.9

Our experiments were implemented using Python and PyTorch on a PC with Xeon E5 2.4 GHz CPUs and Nvidia RTX2080 GPUs. The performance of the tracker will be discussed in the following sections.

3.1. Implementation details

3.1.1. Experimental parameter setting

As shown in **Algorithm 2**, the EfficientNet Module was initialized with parameters pre-trained on ImageNet Dataset (Russakovsky et al., 2015). The dataset used for training consists of 105 dairy goat videos and 510 videos selected from LaSOT (Fan et al., 2019), Youtube-BB (Real et al., 2017), and GOT-10 k (Huang et al., 2019). The 510 videos from other benchmarks were used to enhance the learning ability of the tracker in real scenarios. The size of an exemplar image is $127 \times 127 \times 3$, while the size of a search image is $255 \times 255 \times 3$. There were 50 epochs in total. For the first five epochs, the parameters of the feature extraction network were frozen and the others were trained with a learning rate of 1×10^{-3}. For the remaining epochs, the model was trained end-to-end with a learning rate decreased in log space from 5×10^{-3} to 1×10^{-5}. During the training period, the Cross-entropy loss L_{cls} was employed to train the classification branch and the distance-IoU (DIoU) loss L_{reg} was adopted to train the regression branch. The total loss is expressed as,

$$L = 0.1 \times L_{cls} + 0.9 \times L_{reg} \tag{14}$$

The model was optimized and fine-tuned by the Stochastic Gradient Descent (SGD) (Lecun et al., 1989) algorithm in turn from back to front over four GPUs with a total of 128 pairs per minibatch. The weight decay and the momentum were set as 1×10^{-3} and 0.9, respectively.

3.1.2. Evaluation indicators

In this study, five evaluation methods were employed to comprehensively evaluate the performance of the AMTracker in tracking a single dairy goat in farms. The tracker is repeatedly tested 5 times on the dataset with random position bias, and we used the average score as the final result. The following evaluation methods were used:

1) The Robustness (R) (Kristan et al., 2017; Kristan et al., 2015). It reflects the robustness of the tracker by recording the number of times the tracker fails to track the goat in each video and calculating the failure rate. The lower the value, the higher the efficiency of the algorithm. When the accuracy value (A) in each frame of the video is less than the threshold of 0.2, we consider that the tracker fails to track the object. It is formulated as

$$A_t = \frac{A_t^G \bigcap A_t^P}{A_t^G \bigcup A_t^P} \tag{15}$$

where A_t^G denotes the target ground-truth of the *i-th* frame of the video sequence, denotes the predicted bounding box of the *i-th* frame of the video sequence. The R is calculated as

$$R = \frac{\sum_{sep=1}^{N}\sum_{t=1}^{N_{seq}} F(A_t)}{\sum_{sep=1}^{N}\sum_{i=1}^{N_{seq}} i} \tag{16}$$

where N denotes the number of video sequences, *sep* represents the number of frames of *seq-th* video, let$F(A_i) = 1$ where $A_t < 0.2$, otherwise $F(A_i) = 0$.

2) The Expected Average Overlap (*EAO*) (Kristan et al., 2017; Kristan et al., 2015). It is used to reflect the comprehensive performance of the tracker by calculating the expected value of the tracker's average overlap on a video sequence. The higher the value, the higher the efficiency of the algorithm. First, we calculate the average overlaps $\widehat{A}^{seq}$ of each video sequence as

$$\widehat{A}^{seq} = \frac{1}{N_{seq}}\sum_{t=1}^{N_{seq}} A_t \tag{17}$$

then, evaluating it for a range of sequence lengths, i.e., $seq = 1 : N$ results in the expected average overlap curve. Finally, computing the expected average overlap curve values over an interval $[N_{low}, N_{high}]$ of sequence lengths,

$$\widehat{A} = \frac{1}{N_{high} - N_{low}}\sum_{seq=N_{low}:N_{high}} \widehat{A}^{seq} \tag{18}$$

3) The Success (*Succ*) (Wu et al., 2015; Wu et al., 2013). It represents the percentage of the number of frames tracked successfully by the tracker to all frames. The higher the value, the higher the efficiency of the algorithm. First, calculate the accuracy value (A) in each frame of the video. Second, we calculate the percentage of the number of frames successfully tracked to the total number of frames under different thresholds ($thr_{secc} \in [0,1]$). Last, the *area under the curve* of success is used to indicate the average of all success rates at different thresholds when the thresholds are evenly distributed.

4) The Precision (*Prec*) (Wu et al., 2015; Wu et al., 2013). It is measured by the center distance between the predicted bounding box and the ground-truth. It reflects the precision between the center of the predicted bounding box and the center of the ground-truth. The higher the value, the higher the efficiency of the algorithm. First, we calculate the distance between the center point of the predicted bounding box and the center point of the ground-truth. Next, calculating the percentage of frames whose distance between two points is less than a given threshold ($thr_{prec} \in [0, 50]$). Finally, we use the *area under the curve* of precision to indicate the average of all precision at different thresholds.

5) The Speed (*Speed*) (Kristan et al., 2017; Kristan et al., 2015). It is used to evaluate the processing speed of the tracker in tracking the dairy goat. The Speed is computed as the ratio of the total number of frames to the processing time.

Results and analysis

3.1.3. Efficacy of the dairy goat tracking method

To verify the effectiveness of our proposed dairy goat tracking algorithm, 95 videos collected from an real commercial farm were used for testing. In detail, the ability of the tracker to track the head position of the dairy goat was tested on 30 videos, and the ability of the tracker to track the whole body of the dairy goat was tested on the remaining 65 videos. The tracker was repeatedly tested 5 times on the datasets and we used the average scores as the final results.

Fig. 10 illustrates the change in the loss value of the proposed tracker during the training period. It can be seen that the regression loss and classification loss show a stable close-companion state. Although the loss score dropped rapidly in the first 5 epochs of training, the model tends to be stable with the training progresses. After 50 epochs, the loss score of regression reaches the minimum value of 0.055, the loss score of classification reaches the minimum value of 0.025 and the total loss score is 0.052. This proves that the tracker proposed in this paper has strong robustness and stability, and is suitable for tracking the dairy goat in real scenes.

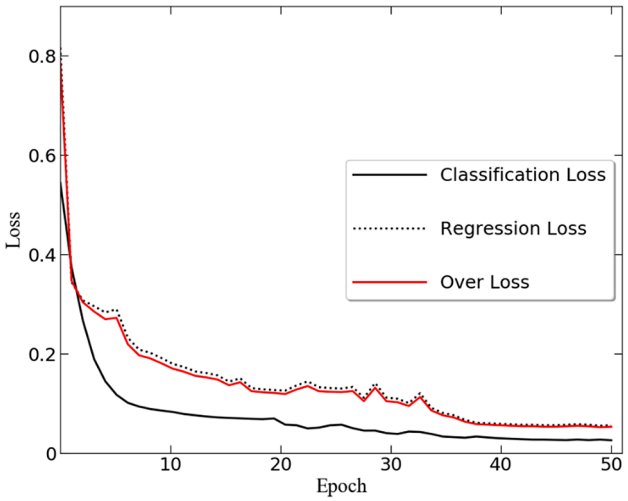

Fig. 10. The loss of the model. The over-loss consists of classification loss and regression loss.

The results of *EAO*, *R* and *Speed* of the tracker are shown in Table 3. *EAO* is used to verify the expected average overlap curve between the ground-truth and the predicted bounding boxes. Our method achieves the *EAO* score of 0.340. The tracker has the ability to predict a bounding box with a higher coincidence rate with the dairy goat position in each frame, which is essential for further behavioral analysis. The *R* score of our tacker is 0.455, which reflects that the tracker can track the dairy goat with a high success rate. The tracker runs at a speed of 30 *fps*, which meets the real-time requirements. This shows that it is feasible to apply our algorithm to the dairy goat tracking in real scenes.

3.1.4. Comparative analysis on the performance of tracking different part

The head and body of the dairy goat are easy to be injured due to their own and external reasons, which has been paid more attention to research on (Wang et al., 2018a). This study performed experiments separately to verify the performance of tracking the head and body.

Here, we focus on investigating the results in Fig. 11 for revealing the effectiveness of goat tracking provided by the proposed algorithm. Fig. 11**(a)** shows the *Prec* scores of tracking the head and body, it can be seen that the precision of our tracker to track the head and body of the dairy goat is 0.842 and 0.831 respectively. Compared with AMTracker_Body, AMTracker_Head even has achieved better performance gains. The gap between the bounding box predicted by AMTracker_Body and AMTracker_Head and the center of the ground-truth is significantly increased in the acceptable distance error threshold area $[0, 10]$. As the error threshold increases, the score increases slowly or does not increase, which means that our tracker rarely loses the object. Ultimately, the overall *Prec* of the AMTracker is 0.835.

Fig. 11**(b)** shows the *Succ* scores of the tracker. The score of AMTracker_Body and AMTracker_Head is 0.650 and 0.669 respectively. The overall *Succ* of the AMTracker is 0.657. It can be seen that within the overlap threshold of 0 to 0.5, the curve changes slowly, but as the overlap threshold changes from 0.5 to 1, the score curve changes significantly. This shows that the bounding box predicted by our tracker is very consistent with the actual area of the dairy goat. These findings demonstrate that the AMTracker can accurately track the head and body of the dairy goat.

Table 3
The results of EAO, R and Speed of the AMTracker.

	EAO↑	*R*↓	*Speed*↑
AMTracker	0.340	0.455	30 *fps*

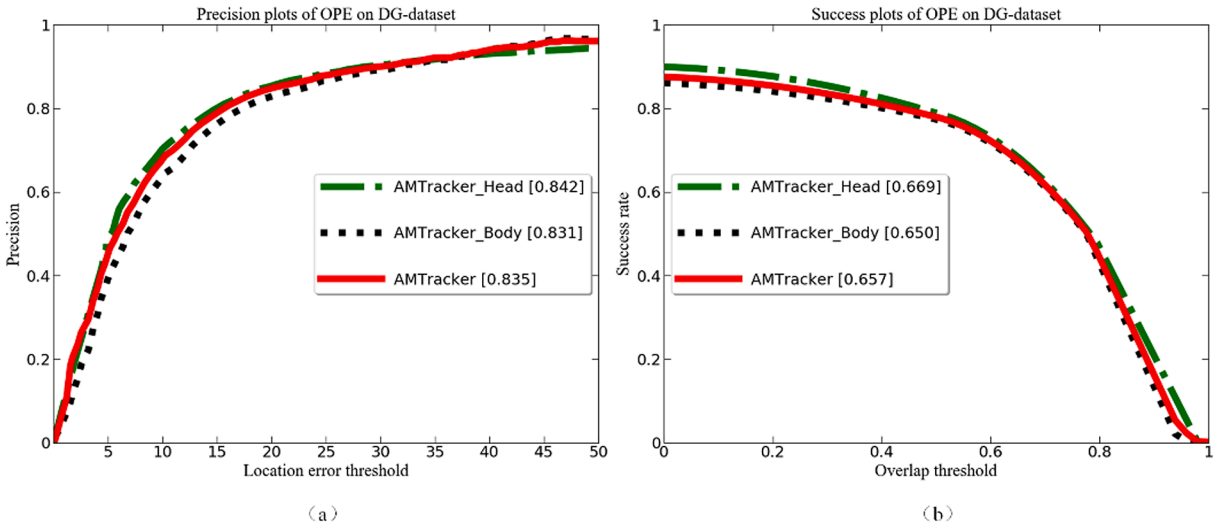

Fig. 11. The Precision plot and Success plot of the tracker. AMTracker_Head means the result of tracking the head of the goat, AMTracker_Body means the result of tracking the body of the goat.

3.1.5. Comparative analysis with other algorithms

As shown in Table 4, a comparative analysis was carried out with five state-of-the-art tracking algorithms, namely SiamRPN (Li et al., 2018), DASiamRPN (Zhu et al., 2018), SiamRPN+ (Zhang and Peng, 2019) and UpdateNet (Zhang et al., 2019), all these trackers utilize two CNNs to independently extract features of the template and search region and detect the dairy goat based on anchors to realize tracking. Similar to AMTracker, we used the DG-Dataset dataset to perform additional training on SiamRPN, SiamRPN+, DaSiamRPN and UpdateNet. Fig. 12 illustrates the *EAO* scores rank on the dairy goat datasets. Compared with the other five tracking algorithms, our tracker achieved the best *EAO* score of 0.340 which surpasses that of the UpdateNet by 13.3%. The *R* of the algorithm in this study reached 0.465 which is 6.8% lower than that of UpdateNet and 11.8% lower than that of the classic SiamRPN. Our tracker executes at 30 *fps*, which is lower than other trackers. In commercial applications, trackers are required to show higher accuracy on the basis of satisfying real-time performance. The AMTracker meets real-time requirements because it has a running speed of 30 frames per second faster than the camera's acquisition rate of 24 frames per second. These findings demonstrate the effectiveness of the proposed AMTracker.

3.1.6. Ablation study

Table 5 illustrates the contribution of the EfficientNet and the BiFPN Module, Attention Module and the Anchor-free Network to the AMTracker.

With vs. without the EfficientNet and BiFPN Module. To test the effectiveness of the EfficientNet and BiFPN Module, it was replaced by the ResNet-50 as the feature extraction network and the feature maps of the last layer were sent to the Attention Module. As shown in the second column of Table 5, the EAO score of the method reaches 0.299 which drops the AMTracker by 12.0%. This shows that the extraction and fusion of multi-level features through EfficientNet and BiFPN Module can aggregate rich feature information that is conducive to tracking the selected dairy goat in the video sequence.

With vs. without the Attention Module. The Attention Module was removed from the AMTracker. It can be seen that the EAO score was 17.1% lower than that of AMTracker. This suggests that the Attention Module is critical to the tracking results. The cross-attention to construct an interactive bridge between the dairy goat template and the search frame to share the same channel weight and the self-spatial attention focuses on the identification part of the relevant feature map, which greatly improves the accuracy of detecting the selected dairy goat in a large number of similar target interference situations.

With vs. without the Anchor-free Network. The Anchor-free Network was removed from the network and replaced with the RPN similar to (Li et al., 2018). The EAO score has dropped by 3.8%. The reason is that the anchor-based method has an inherent disadvantage, that is, the

Table 4
Overall performance comparisons on the DG-dataset, and the best two results highlighted in red and blue bold fonts, respectively.

	EAO↑	*R*↓	*Speed*↑
SiamRPN (Li et al., 2018)	0.236	0.527	**150 *fps***
DASiamRPN (Zhu et al., 2018)	0.254	0.500	**130 *fps***
SiamRPN+ (Zhang and Peng, 2019)	0.271	0.515	120 *fps*
UpdateNet (Zhang et al., 2019)	**0.300**	**0.488**	90 *fps*
AMTracker (ours)	**0.340**	**0.465**	30 *fps*

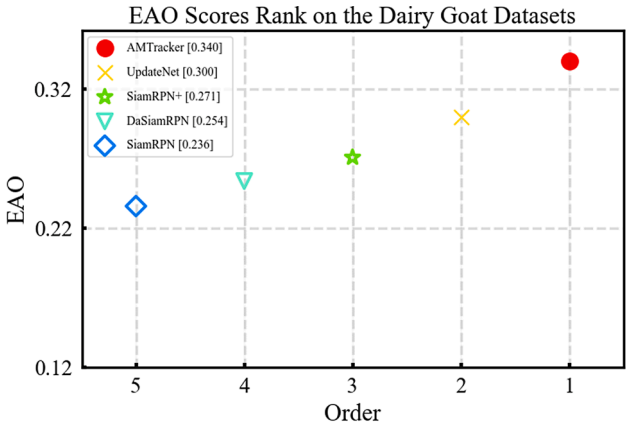

Fig. 12. Expected average overlap (EAO) plot on the dairy goat datasets. The listed methods, such as UpdateNet, SiamRPN+, DaSiamRPN, and SiamRPN are compared with our tracker.

Table 5
Contribution of different modules to network.

EfficientNet + BiFPN Module	×	✓	✓	✓
Attention Module	✓	×	✓	✓
Anchor-free Network	✓	✓	×	✓
EAO	0.299	0.282	0.327	0.340

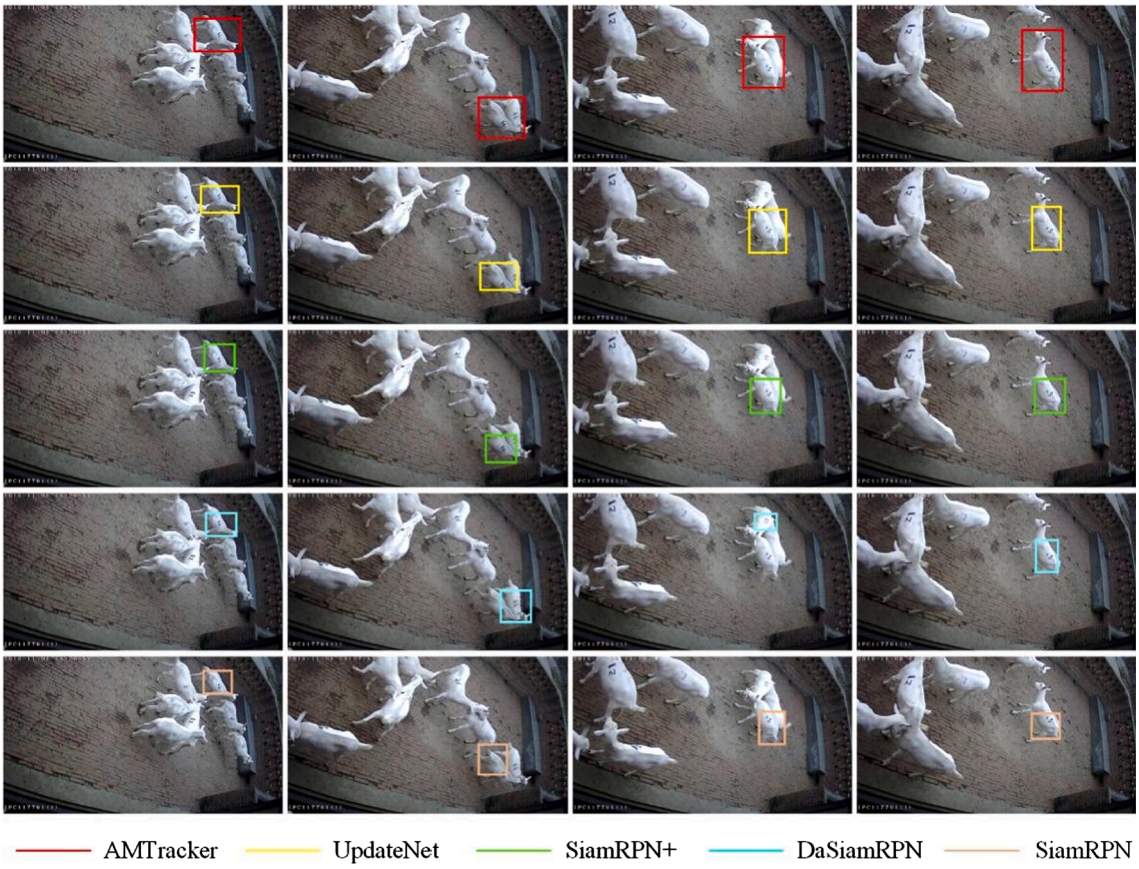

Fig. 13. Performance of different algorithms in tracking the whole body of the dairy goat. Our method can predict the body bounding boxes more precisely than UpdateNet, SiamRPN+, DaSiamRPN and SiamRPN in the real scene.

predicted bounding box is limited by the ratio of the anchors, but the Anchor-free Network can detect the dairy goat without the scale limitation of anchors.

Visual analysis of the results

Fig. 13 and Fig. 14 illustrate the performance of different algorithms in tracking the dairy goat in the dataset. The results of tracking the whole body of the dairy goat were randomly selected from four frames in the video sequence '00000085′, namely 117th frame, 426th frame, 601th and 629th frame, and results of tracking the head of the dairy goat were randomly selected from four frames in the video sequence '00000186′, namely 2th frame, 211th frame, 383th and 423th frame. Each row from top to bottom of the picture shows the performance of AMTracker, UpdateNet, SiamRPN+, DaSiamRPN and SiamRPN, respectively. It can be seen that as the dairy goat moves, the bounding box predicted by the AMTracker has the ability to cover the tracked position of the dairy goat as much as possible while introducing less background, which is of great significance for further behavior recognition. It can be seen from Fig. 13 that the UpdateNet tracker can track the target accurately, but the overlap rate between the predicted region of the dairy goat and the location region is not as high as that of the AMTracker. When the shape of the dairy goat changes and there is interference from other dairy goats, DaSiamRPN and SiamRPN drifted, but AMTracker did not. When the target scale changes, benefiting from not being restricted by the ratio of anchor, the bounding box predicted by the AMTracker has a higher coverage rate than the bounding box predicted by SiamRPN+, DaSiamRPN and SiamRPN. These are the reasons why the *EAO* of our tracker is higher than other algorithms. The result shows that our tracker can be applied in real farm scenes to lay the foundation for further behavior recognition and analysis.

4. Conclusion

Tracking and monitoring the dairy goat is an important prerequisite for evaluating abnormal behavior and predicting disease. In this paper, we have proposed a single dairy goat tracking algorithm AMTracker guided by an attention mechanism. We utilized the EfficientNet and the BiFPN to embed and fuse multi-level feature maps respectively, and then the Attention Module was used to enhance the dependence of template features and search region features. Through these methods, we effectively prevent similar objects in the background from interfering with the dairy goat track. The Anchor-free network was used to predict the position of the dairy goat. In order to test the effectiveness of the algorithm, we collected 200 surveillance videos by the camera in the real farm. The experimental results showed that the AMTracker was superior to the four state-of-the-art algorithms, with the expected average overlap score of 0.340 and the robustness score of 0.455. This study showed that it is possible to use computer vision technology to track animals under the premise of solving the problem of scale change and similarity interference. In a word, the method proposed in this paper has good practical value. Through accurate tracking of the dairy goat, continuous observation of dairy goat behavior can be realized for further analysis, which is conducive to accurate monitoring of dairy goat production in

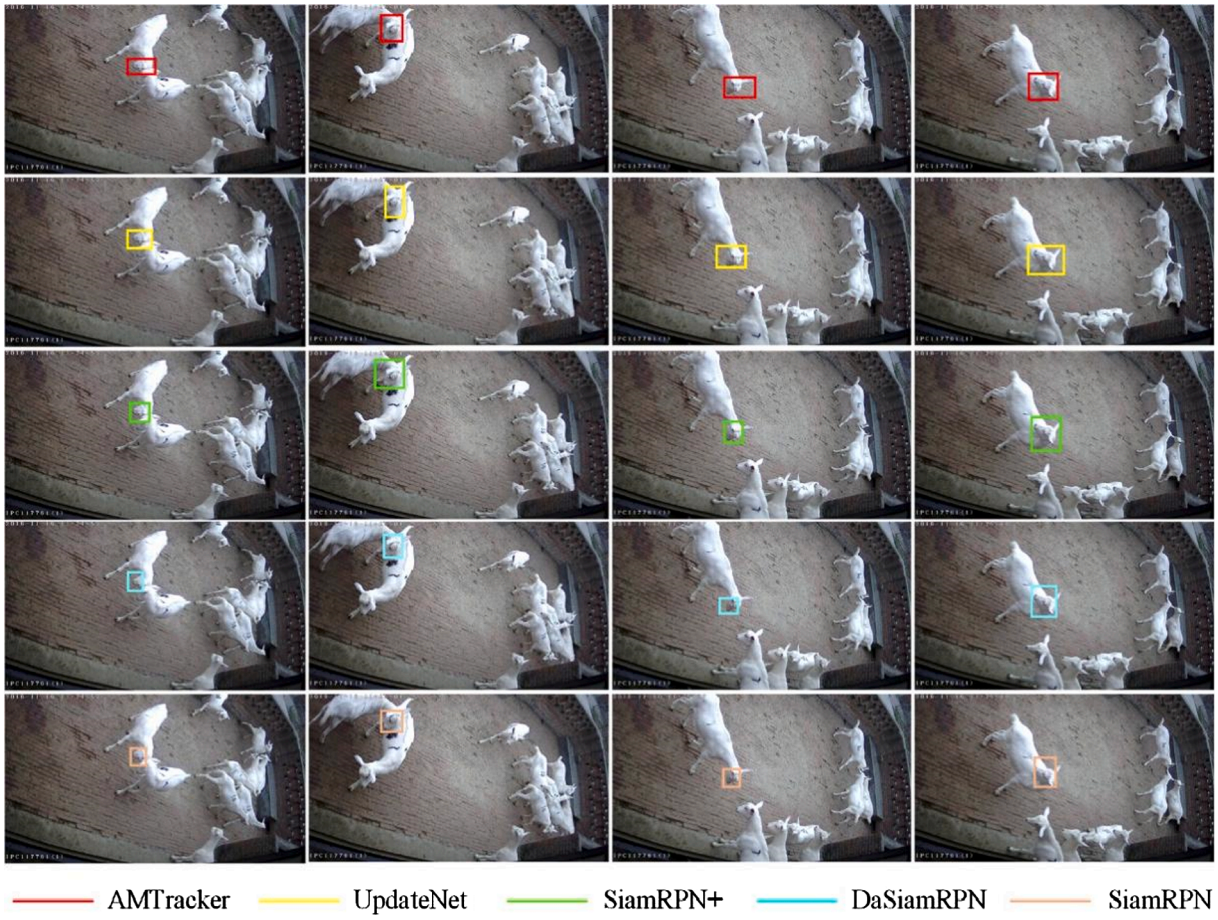

Fig. 14. Performance of different algorithms in tracking the head of the dairy goat. Our method can predict the head bounding boxes more precisely than UpdateNet, SiamRPN+, DaSiamRPN, and SiamRPN in the real scene.

large-scale commercial farms.

Although our algorithm has achieved excellent results, it is still a difficult problem to accurately track a single individual in a group due to a variety of complex difficulties. To improve the robustness of tracking the dairy goat, we will consider more about how to solve the problem that the target is occluded by other goats or background objects in future research. At the same time, we will explore abnormal behavior monitoring technology for dairy goats in future research work to improve animal welfare.

CRediT authorship contribution statement

Qingguo Su: Software, Validation, Formal analysis. **Jinglei Tang:** Conceptualization, Methodology, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Mingxin Zhai:** Writing – review & editing, Data curation. **Dongjian He:** Conceptualization, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the Innovation Support Plan of Shaanxi Province (No.2021TD-31), the Agricultural Science and Technology Innovation Driven Project of Shaanxi Province (No.NYKJ-2021-YL(XN)09), the Key Research and Development Project of Shaanxi Province (Grant No.2020NY098,2021NY-138). The authors appreciate the funding organizations for their financial supports.

References

Ahrendt, P., Gregersen, T., Karstoft, H., 2011. Development of a real-time computer vision system for tracking loose-housed pigs. Comput Electron Agr 76, 169–174.

Chen, C., Weixing, Z., Juan, S., Janice, S., Kaitlin, W., Junjie, H., Tomas, N., 2020. Recognition of aggressive episodes of pigs based on convolutional neural network and long short-term memory. Comput Electron Agr 169, 105–166.

Duan, K.W., Bai, S., Xie, L.X., Qi, H.G., Huang, Q.M., Tian, Q., 2019. CenterNet: Keypoint Triplets for Object Detection, 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul, Korea (South), pp. 6568–6577.

Endo, N., Rahayu, L.P., Arakawa, T., Tanaka, T., 2016. Video tracking analysis of behavioral patterns during estrus in goats. J Reprod Develop 62, 115–119.

Fan, H., Ling, H., Lin, L., Yang, F., Liao, C., 2019. LaSOT: A High-Quality Benchmark for Large-Scale Single Object Tracking, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA, pp. 5369–5378.

Fang, C., Huang, J.D., Cuan, K.X., Zhuang, X.L., Zhang, T.M., 2020. Comparative study on poultry target tracking algorithms based on a deep regression network. Biosyst Eng 190, 176–183.

Huang, L.H., Zhao, X., Huang, K.Q., 2019. GOT-10k: A Large High-Diversity Benchmark for Generic Object Tracking in the Wild. IEEE Transactions on Pattern Analysis and Machine Intelligence 1, 1-1.

Jiang, M., Rao, Y., Zhang, J.Y., Shen, Y.M., 2020a. Automatic behavior recognition of group-housed goats using deep learning. Comput Electron Agr 177.

Jiang, M., Rao, Y., Zhang, J.Y., Shen, Y.M., 2020b. Automatic behavior recognition of group-housed goats using deep learning. Comput Electron Agr 177, 105–706.

Kolarevic, J., Aas-Hansen, O., Espmark, A., Baeverfjord, G., Terjesen, B.F., Damsgard, B., 2016. The use of acoustic acceleration transmitter tags for monitoring of Atlantic salmon swimming activity in recirculating aquaculture systems (RAS). Aquacult Eng 72–73, 30–39.

Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., 2017. The Visual Object Tracking VOT2017 challenge results, 2017 IEEE International Conference on Computer Vision Workshops (ICCVW). Venice, Italy, pp. 1949–1972.

Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Cehovin, L., Fernandez, G., Vojir, T., Hager, G., Nebehay, G., Pflugfelder, R., 2015. The Visual Object Tracking VOT2015 Challenge Results. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops, pp. 1–23.

Lecun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L. D., 1989. Backpropagation Applied to Handwritten Zip Code Recognition. Neural Computation 1, 541–551.

Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J., 2019. SiamRPN++: Evolution of Siamese Visual Tracking With Very Deep Networks, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA, pp. 4277–4286.

Li, B., Yan, J.J., Wu, W., Zhu, Z., Hu, X.L., 2018. High Performance Visual Tracking With Siamese Region Proposal Network, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, pp. 8971–8980.

Li, T., Dollár, P., Girshick, R., He, K.M., Hariharan, B., Belongie, S., 2017. Feature Pyramid Networks for Object Detection, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA, pp. 936–944.

Liu, D., Oczak, M., Maschat, K., Baumgartner, J., Pletzer, B., He, D., Norton, T., 2020. A computer vision-based method for spatial-temporal action recognition of tail-biting behaviour in group-housed pigs. Biosyst Eng 195, 27–41.

Nasirahmadi, A., Edwards, S.A., Sturm, B., 2017. Implementation of machine vision for detecting behaviour of cattle and pigs. Livest Sci 202, 25–38.

Neubeck, A., Gool, L., 2006. Efficient Non-Maximum Suppression, 18th International Conference on Pattern Recognition (ICPR'06). China, Hong Kong, pp. 850–855.

Rachel, S.E.P., Simon, P.T., Laura, A.B., Irene, C., 2018. The translation of animal welfare research into practice: The case of mixing aggression between pigs. Appl Anim Behav Sci 204, 1–9.

Real, E., Shlens, J., Mazzocchi, S., Pan, X., Vanhoucke, V., 2017. YouTube-BoundingBoxes: A Large High-Precision Human-Annotated Data Set for Object Detection in Video, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, pp. 7464–7473.

Ren, S.Q., He, K.M., Girshick, R., Sun, J., 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Ieee Transactions on Pattern Analysis and Machine Intelligence 39, 1137–1149.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L., 2015. Imagenet large scale visual recognition challenge. International Journal of Computer Vision 115, 211–252.

Sakai, K., Oishi, K., Miwa, M., Kumagai, H., Hirooka, H., 2019. Behavior classification of goats using 9-axis multi sensors: The effect of imbalanced datasets on classification performance. Comput Electron Agr 166.

Tan, M., Le, Q., 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, In: Kamalika, C., Ruslan, S. (Eds.), Proceedings of the 36th International Conference on Machine Learning. PMLR, Long Beach, California, USA, pp. 6105-6114.

Tan, M., Pang, R., Le, Q.V., 2020. EfficientDet: Scalable and Efficient Object Detection, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, WA, USA, pp. 10778–10787.

Tang, J.L., Yang, G.X., Sun, Y.R., Xin, J., He, D.J., 2019. Salient object detection of dairy goats in farm image based on background and foreground priors. Neurocomputing 332, 270–282.

Tian, Z., Shen, C., Chen, H., He, T., 2019. FCOS: Fully Convolutional One-Stage Object Detection, 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul, Korea (South), pp. 9626–9635.

Vayssade, J.A., Arquet, R., Bonneau, M., 2019. Automatic activity tracking of goats using drone camera. Comput Electron Agr 162, 767–772.

Wang, D., Tang, J.L., Zhu, W.J., Li, H., Xin, J., He, D.J., 2018a. Dairy goat detection based on Faster R-CNN from surveillance video. Comput Electron Agr 154, 443–449.

Wang, Q., Teng, Z., Xing, J.L., Gao, J., Hu, W.M., Maybank, S., 2018b. Learning Attentions: Residual Attentional Siamese Network for High Performance Online Visual Tracking, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA, pp. 4854–4863.

Wu, Y., Lim, J., Yang, M.-H., 2015. Object Tracking Benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence 37, 1834–1848.

Wu, Y., Lim, J., Yang, M., 2013. Online object tracking: A benchmark, 2013 IEEE Conference on Computer Vision and Pattern Recognition. Portland, OR, USA, pp. 2411–2418.

Yin, X.Q., Wu, D.H., Shang, Y.Y., Jiang, B., Song, H.B., 2020. Using an EfficientNet-LSTM for the recognition of single Cow's motion behaviours in a complicated environment. Comput Electron Agr 177.

Yu, Y.C., Xiong, Y.L., Huang, W.L., Scott, M.R., 2020. Deformable Siamese Attention Networks for Visual Object Tracking, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, WA, USA, pp. 6727–6736.

Zhang, A.L.N., Wu, B.P., Wuyun, C.T.N., Jiang, D.X.H., Xuan, E.C.Z., Ma, F.Y.H., 2018. Algorithm of sheep body dimension measurement and its applications based on image analysis. Comput Electron Agr 153, 33–45.

Zhang, L., Abel, G.-G., De, W.J.V., Martin, D., Shahbaz, K.F., 2019. Learning the Model Update for Siamese Trackers, 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul, Korea (South), pp. 4009–4018.

Zhang, Z.P., Peng, H.W., 2019. Deeper and Wider Siamese Networks for Real-Time Visual Tracking, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, pp. 4586–4595.

Zheng, C., Zhu, X.M., Yang, X.F., Wang, L.N., Tu, S.Q., Xue, Y.J., 2018. Automatic recognition of lactating sow postures from depth images by deep learning detector. Comput Electron Agr 147, 51–63.

Zheng, Z., Wang, P., Liu, W., Li, J., Ren, D., 2020. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression, AAAI Conference on Artificial Intelligence 2020, New York, US.

Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., Hu, W., 2018. Distractor-aware Siamese Networks for Visual Object Tracking, Computer Vision – ECCV 2018. Munich, Germany, pp. 101–117.