Automatic lameness detection in dairy cows based on machine vision

Zongwei Jia*, Xuhui Yang, Zhi Wang, Ruirui Yu, Ruibin Wang

(*College of Information Science and Engineering, Shanxi Agricultural University, Taigu 030800, Shanxi, China*)

Abstract: This study proposed a method for detecting lameness in dairy cows based on machine vision, addressing the challenges associated with manual detection. Data from a dairy farm in Taigu, Shanxi, China were collected and divided into two parts. The first part was utilized to precisely position the cow's back by employing a dedicated deep learning model named GhostNet_YOLOv4, which can be implemented on mobile or embedded devices. The second part was used with the Visual Background Extractor (Vibe) algorithm, incorporating additional morphological processing techniques. Enhancing the Vibe algorithm, a widely used background subtraction algorithm for image sequences, achieved more accurate recognition of the specific pixel areas of cows. Subsequently, cow shape-related feature parameters were extracted from the back area using the combined approach. These parameters were used to calculate the average curvature, which describes the degree of curvature of the cow's back contour during walking. The differences in curvature values were employed for classification to detect lameness. Through extensive experimentation, distinct average curvature ranges of [−0.025, −0.125], [−0.025, +∞], and [−∞, −0.125] were established for normal cows, early lameness, and moderate-severe lameness, respectively. The algorithm's effectiveness was validated by processing 600 image sequences of dairy cows, resulting in a lameness detection accuracy of 91.67%. These findings can serve as a reference for the timely and accurate recognition of lameness in dairy cows.

Keywords: dairy cow, lameness detection, machine vision, object detection, deep learning

DOI: 10.25165/j.ijabe.20231603.8097

Citation: Jia Z W, Yang X H, Wang Z, Yu R R, Wang R R. Automatic lameness detection in dairy cows based on machine vision. Int J Agric & Biol Eng, 2023; 16(3): 217–224.

1 Introduction

Cow lameness refers to an abnormal gait resulting from pain from foot disease[1]. According to a research report by Goldman Sachs, the global average rate of lameness in dairy cows is 23.5%. The associated annual costs for treatment, loss of milk production, and labor amount to more than 11 billion[2]. Lameness in dairy farms has emerged as a recurring issue that affects the well-being of dairy cows and poses a significant challenge to modern dairy production. Unfortunately, current farm conditions make cow lameness appear inevitable[3]. Therefore, automatic cow lameness identification technology has increasingly attracted attention in the dairy industry.

Traditional methods for detecting lameness in dairy cows rely on manual observation. The widely used five-point scoring system was innovatively proposed by Sprecher et al.[4] This system evaluates lameness by assessing cows' posture and gait differences when lameness occurs. However, this method is subject to subjective influences and requires substantial human and financial resources[5]. Since the beginning of the new century, advancements in sensing and electronic technologies have introduced faster and more effective ways to detect lameness in cows. The kinetic method directly collects data on the four-balance leg weights using a weighing platform[6]. Chapinal et al.[7] indirectly obtained a cow's walking acceleration information using sensors attached to the cow' s leg. These variable data, collected automatically by sensor systems, allow for a more objective lameness evaluation. Electronic information technology, which differs from sensor technology, employs noncontact methods to identify cow lameness automatically. It has been a key research direction in recent years for detecting lameness in dairy cows. Bahr et al.[8] detected lameness based on hoof-ground contact time, achieving an accuracy rate of 84%. Song et al.[9] fit a straight line by extracting the slope feature at the junction of the head, neck, and back to automatically detect lameness, with an accuracy of 93.89%. Zhao et al.[10] utilized computer vision technology to analyze leg swings for lameness detection, achieving an accuracy rate of 90.18%. Currently, most studies focus on acquiring features related to lameness, such as stride length[11], head posture[12], and back posture[13]. These features can be utilized to detect lameness in dairy cows. This study aimed to provide targeted and adaptable technical support for the early detection of cow lameness to reduce losses, save labor and costs, and improve the efficiency of pastoral stations.

2 Materials and methods

2.1 Data

The data were collected from the PuYuanTai Cow Breeding Company in Taigu, Shanxi, China. The data were captured from a side-view angle of Holstein cows while walking, enabling effective extraction of critical features such as the cow's head, neck, back, and legs. This perspective is also advantageous for fitting a contour curve. After milking, the cows traversed a long, narrow passage. A Sony FDR-AX45 camera was positioned on one side of the passage alley to ensure clear and complete side-view footage of a walking cow. Figure 1 illustrates the layout of the cattle farm.

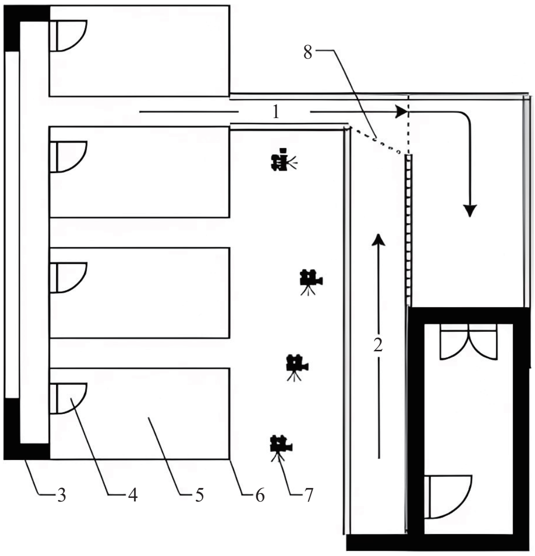
In this experiment, multiple video segments were collected from May to July 2021, encompassing three periods: morning, noon, and dusk. The duration of each segment ranged from 8 to 6

Received date: 2022-12-17 **Accepted date:** 2023-07-05

Biographies: **Xuhui Yang**, MS candidate, research interest: agricultural informatization, Email: 2452170870@qq.com; **Zhi Wang**, MS candidate, research interest: agricultural informatization, Email: 1164615153@qq.com; **Ruirui Yu**, MS candidate, research interest: agricultural informatization, Email: 3362447727@qq.com; **Ruibin Wang**, MS, research interest: agricultural informatization, Email: 1006000617@qq.com.

***Corresponding author:** **Zongwei Jia**, MS, Associate Professor, research interest: deep learning and agricultural informatization, Mailing address: College of Information Science and Engineering, Shanxi Agricultural University, Taigu 030801, Shanxi, China. Email: jiazw@sxau.edu.cn.

min. The videos were captured at a frame rate of 25.00 fps and a resolution of 3840 pixels (horizontal)×2160 pixels (vertical). In addition to capturing the recording time, the videos encompassed various weather conditions, including overcast and sunny skies and distinct variations in backgrounds. Moreover, the videos featured numerous interfering factors, such as birds and insects, railings, mechanical equipment, buildings, and personnel. These factors pose additional challenges in accurately detecting cow lameness.



1. Access channel 2. Returned channel 3. Wall 4. Door 5. Cattle house 6. Railing 7. Camera 8. Chain

Figure 1 Diagram of the cattle farm from the PuYuanTai Cow Breeding Company

2.2 Data set

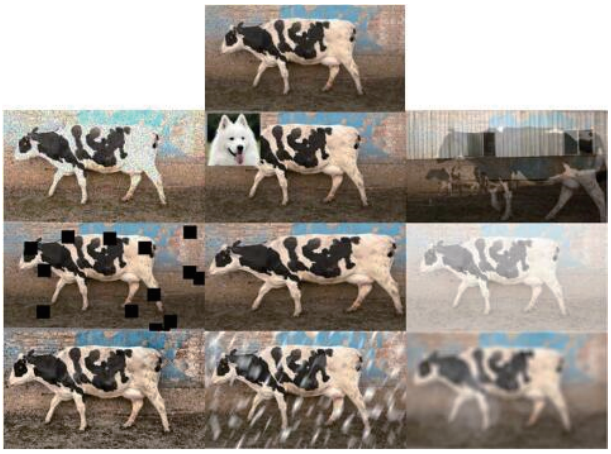
The collected data were subjected to three steps of screening and processing to establish the final data set for this experiment.

1) Python was used to extract all video frames and save them as JPG format images. After manual screening, the sample size was uniformly adjusted to 608 pixels×608 pixels. This adjustment aimed to reduce the computational load of the model and improve training speed.

2) To enhance the data set's robustness and improve the model's generalization ability, data augmentation was performed on the images filtered in Step 1). Several methods were employed to transform the spatial geometry and pixel content. First, luminosity conversion introduced a random augmentation factor to enhance the image's chroma, contrast, and sharpness while incorporating salt and pepper noise and Gaussian noise. The strength of the augmentation was determined by this random factor. Second, the Mixup method[14], the Cutout method[15], and CutMix method[16] were used for image cropping and filling. Third, the Albumations image augmentation library[17] provided more than ten different methods of random-degree augmentation, including scaling, rotation, elastic transformation, gamma transformation, adaptive histogram equalization, and GridDropout. In response to challenging weather conditions that interfered with high-quality data acquisition and the difficulty of data collection, weather augmentation was introduced to simulate rainy and foggy conditions. Figure 2 demonstrates a portion of the data augmentation process.

3) The data obtained from the previous steps were labeled using an image labeling tool called Labelimg.

Following the aforementioned operations, two data sets were established. The first data set consisted of 5000 images capturing critical parts of cows. Among these, 3000 images were randomly assigned to the training set, 1000 to the validation set, and 1000 to the test set. The second data set comprised sequential image data obtained in Step 1). The 600-segment sequence data set was divided randomly, with 480 segments allocated to the training set and 120 segments to the test set using the leave-out method.



Note: The data augmentation process is presented from top to bottom and left to right, in the following order: original image, luminosity conversion, CutMix, Mixup, Cutout, elastic transform, gamma transform, adaptive histogram equalization, simulated rainy day, and simulated foggy day.

Figure 2 Data augmentation process of dairy cows images

2.3 Detection method of cow lameness

The cow lameness detection method employed in this study is illustrated in Figure 3, comprising three main components. The first component is the GhostNet_YOLOv4 deep learning model, utilized for detecting and localizing the back area of the cows. The second component involves the Vibe algorithm with additional morphological processing methods. The Vibe algorithm model is applied to separate the foreground and background, thereby obtaining the pixel area corresponding to the target cow. The third component focuses on the lameness judgment model. By utilizing the target cow's back location and pixel area, the contour of the cow's back was extracted, and a curve was fitted. The average curvature was then computed, and the differences in curvature were employed to classify and identify normal cows, early-stage lameness, and moderate-to-severe lameness.

2.4 GhostNet_YOLOv4 deep learning model

In this study, the GhostNet architecture served as the backbone network, while the path aggregation network (PANet) functioned as the neck, and the YOLOv3 framework was employed as the head. These components were combined to form a deep learning model known as GhostNet_YOLOv4. The specific improvements incorporated in this model are outlined as follows:

1) Backbone

Figure 4 displays the outcome of the first convolutional feature map generated by the VGG16 Model. Inputting an image yields many feature maps, ensuring that the machine comprehensively understands the input data. Many similarities were observed among some feature maps and can be considered redundancies. However, these redundancies are not without merit; they enhance the machine's image recognition capabilities. GhostNet adopts computationally less expensive operations to generate features without attempting to eliminate or reduce these redundancies. Consequently, GhostNet emerges as an efficient, lightweight architecture for deep convolutional models[18].

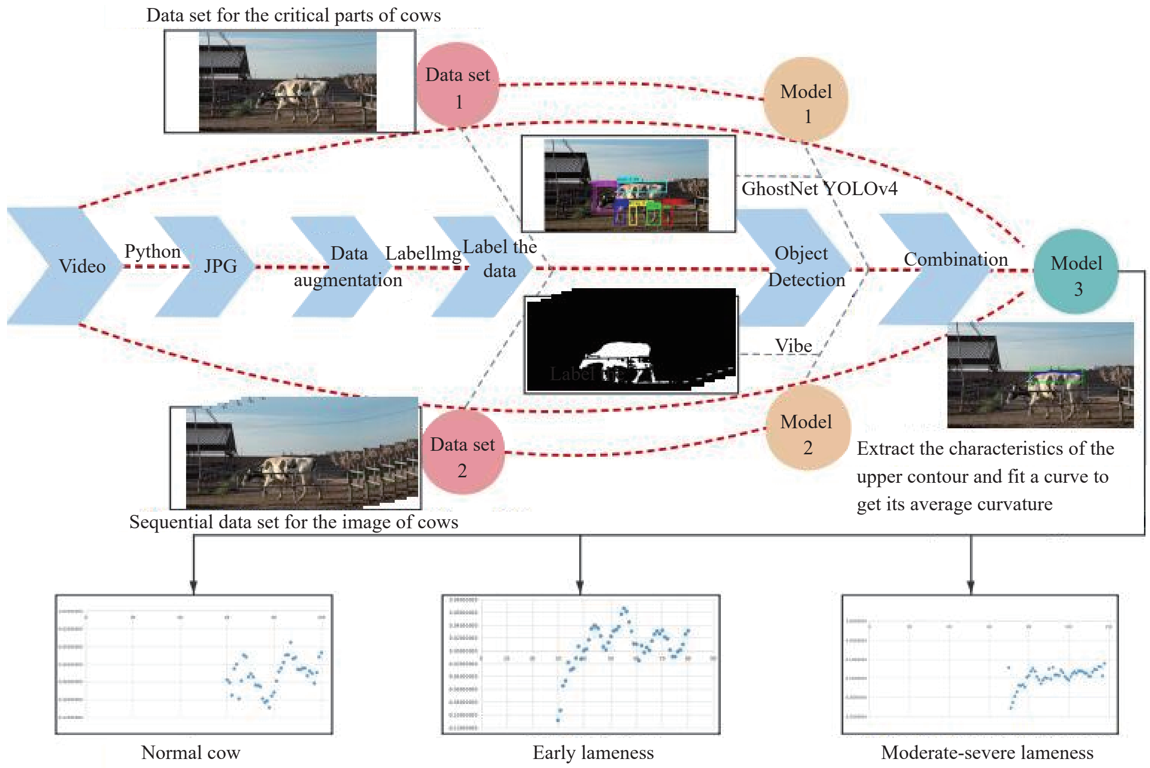
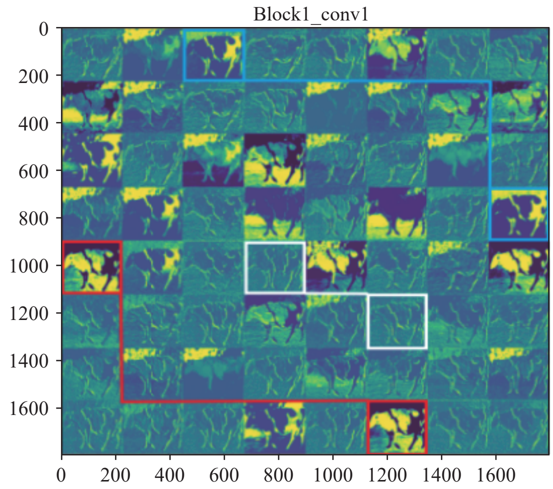


Figure 3 Technology roadmap of the method proposed in this study



Note: The boxes connected by blue, white, and red lines in the above figure are redundant feature maps

Figure 4 First convolution feature graph from the VGG16 Model

To extract the features from the input image, GhostNet was employed as the backbone network owing to its lightweight nature and the ability to deliver satisfactory performance. The three effective feature layers obtained through GhostNet are substituted with corresponding layers from Cross Stage Partial darknet53 (CSPdarknet53) in the original YOLOv4[19]. The Ghost module is crucial to generating the feature map of the input image. It follows a two-step process. First, a 1×1 convolution operation is applied to generate a small portion of the feature map. Second, a cost-effective depth-separable convolution is employed to linearly transform the features obtained in the previous step, resulting in the Ghost feature map. The output is a combination of the two feature maps, requiring fewer parameters compared with a regular convolutional layer while producing the same number of feature maps. Furthermore, the Ghost module stacking technique is employed to construct Ghost Bottlenecks resembling a bottleneck structure. A lightweight Squeeze-and-Excitation (SE) module attention model is incorporated in some residual layers. The SE module adjusts the weight of each channel, introducing a slight increase in model complexity and computational burden but resulting in significant performance enhancements. The overall architecture of GhostNet, comprised of the Ghost Bottlenecks, is illustrated in Figure 5.

2) Neck

The neck module plays a crucial role in fusing the features extracted by the backbone network. YOLOv4 incorporates a Spatial Pyramid Pooling (SPP) structure and a path aggregation network (PANet) to construct the neck module. SPP combines local and global features, effectively expanding the receptive field and enhancing feature representation. In addition to the feature pyramid network (FPN) layer used in YOLOv3, YOLOv4 introduces a feature pyramid behind the FPN layer to form the PANet. The FPN layer transfers high-level semantic features to lower layers and conveys precise location information and weaker semantic features from lower layers to upper layers, leading to more accurate positioning signals.

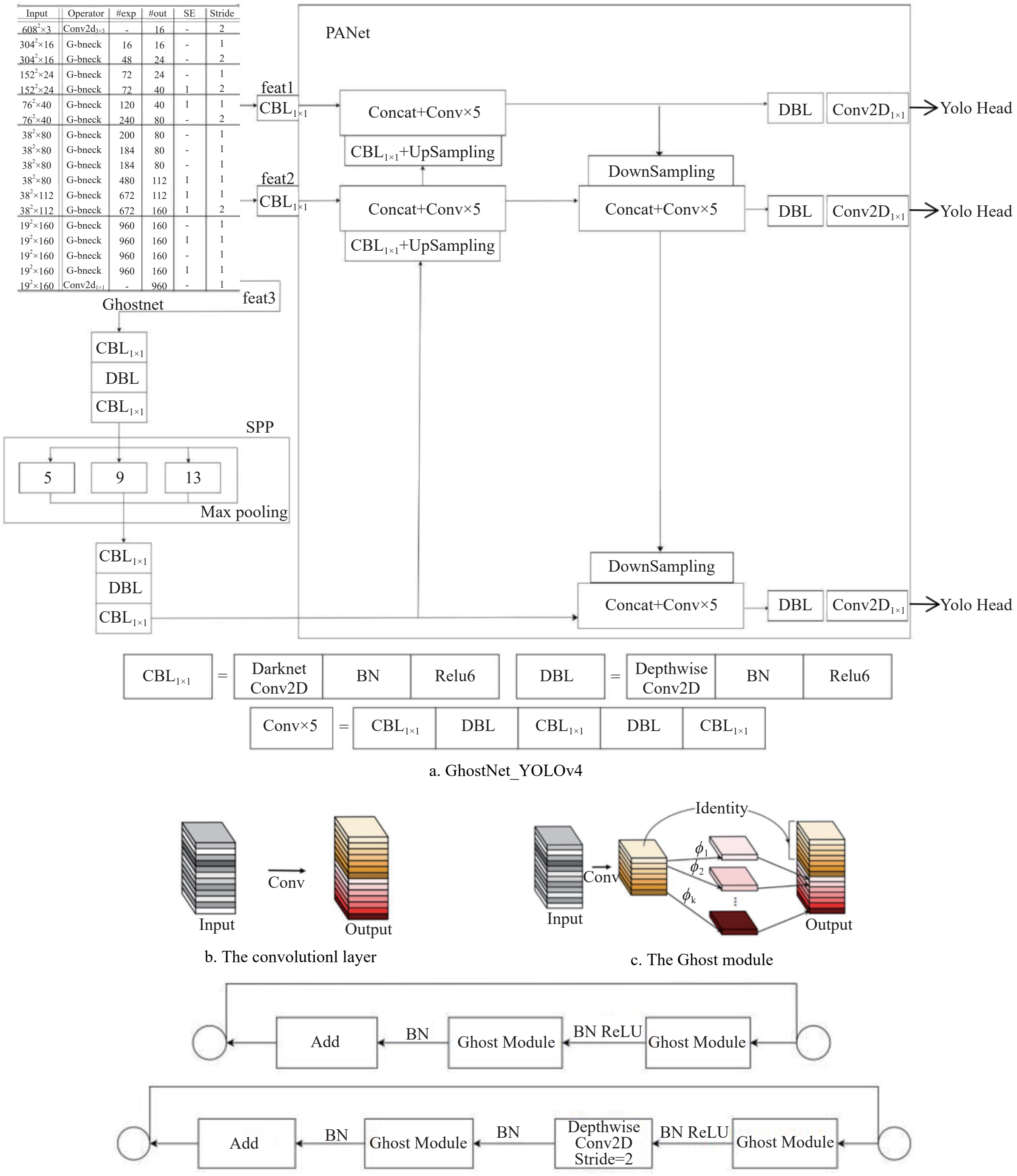
The neck part of the original YOLOv4 extensively employs 3×3 convolutions. However, this study used deep separable convolutions to replace them, reducing the number of parameters.

3) Bounding box regression loss function

Regarding the bounding box regression loss function, the Intersection over Union (IOU) is replaced by the complete-Intersection over Union (CIOU) as the regression optimization loss. CIOU incorporates the scale of the detection frame, aligning the predicted frame more closely with the actual frame[20]. The formula for CIOU is as follows:

$$\mathrm{LOSS}_{\mathrm{CIOU}}=1-\mathrm{IOU}+\frac{\rho^2(b,b^{gt})}{c^2}+\alpha\upsilon \tag{1}$$

$$R_{\mathrm{CIOU}}=\frac{\rho^2(b,b^{gt})}{c^2}+\alpha\upsilon \tag{2}$$



Input	Operator	#exp	#out	SE	Stride
$608^2\times3$	Conv2d$_{3\times3}$	-	16	-	2
$304^2\times16$	G-bneck	16	16	-	1
$304^2\times16$	G-bneck	48	24	-	2
$152^2\times24$	G-bneck	72	24	-	1
$152^2\times24$	G-bneck	72	40	1	2
$76^2\times40$	G-bneck	120	40	1	1
$76^2\times40$	G-bneck	240	80	-	2
$38^2\times80$	G-bneck	200	80	-	1
$38^2\times80$	G-bneck	184	80	-	1
$38^2\times80$	G-bneck	184	80	-	1
$38^2\times80$	G-bneck	480	112	1	1
$38^2\times112$	G-bneck	672	112	1	1
$38^2\times112$	G-bneck	672	160	1	2
$19^2\times160$	G-bneck	960	160	-	1
$19^2\times160$	G-bneck	960	160	1	1
$19^2\times160$	G-bneck	960	160	-	1
$19^2\times160$	G-bneck	960	160	1	1
$19^2\times160$	Conv2d$_{1\times1}$	-	960	-	1

a. GhostNet_YOLOv4

b. The convolutionl layer

c. The Ghost module

d. Ghost bottleneck.

Note: Top: Ghost bottleneck with strde=1; Bottom: Ghost bottleneck with stride=2 d. Ghost bottleneck.

Figure 5 GhostNet_YOLOv4 network structure

$$\upsilon = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \tag{3}$$

$$\alpha = \frac{\upsilon}{(1-\mathrm{IOU})+\upsilon} \tag{4}$$

where, Loss represents the bounding box regression loss function, IOU measures the overlap between two bounding boxes; R is a penalty term; c represents the diagonal distance of the smallest enclosing area that can contain both the prediction box and the ground truth box; v measures the consistency of the aspect ratio between the predicted frame and the actual frame; α is a weight function; $\rho^2(b, b^{gt})$ denotes the Euclidean distance between the center points of the predicted frame and the actual frame, respectively; w and w^{gt} respectively represent the width of the predicted box and the ground truth box, h and h^{gt} respectively represent the height of the predicted box and the ground truth box.

2.5 Vibe computer vision model

Vibe is an object detection algorithm that offers advantages over Background Subtraction, Optical Flow, and other methods. It is characterized by its low computational requirements, minimal memory usage, and fast processing speed. It performs well in complex background conditions and effectively eliminates ghosting artifacts[21]. The algorithm is implemented in Python and consists of the following four main steps:

1) Initialization

Traditional background subtraction methods typically require multiple sequential images for background model initialization. However, the Vibe algorithm can build a background model using a single image frame. By considering the approximate temporal and spatial distribution characteristics of similar pixels, the algorithm randomly selects n pixels to model in the vicinity of each pixel. The background model is defined as follows:

$$M = \{v_1, v_2, v_3, \cdots, v_n\} \tag{5}$$

2) Foreground and background detection

After initialization, for each new pixel $v(x)$ in the subsequent frame, a Euclidean space SR($v(x)$) with $v(x)$ as the center and R as the radius is defined (as illustrated in Figure 6). The set of pixels within this space is compared with the sample set in the background model $M(x)$. If the similarity between the sets is high, the pixel is classified as a background point (label 0); otherwise, it is classified as a foreground point (label 1). A similarity coefficient K is used to quantify the degree of similarity. When the intersection $N(x)$ is greater than or equal to a certain threshold nmin, it indicates a similarity between the new pixel point set and the sample set. Otherwise, they are considered dissimilar. The formula is as follows:

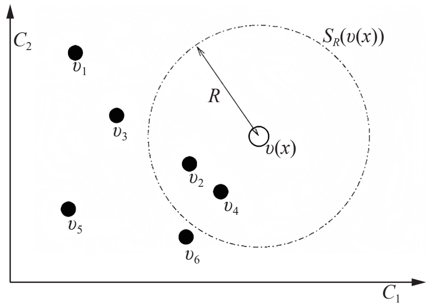


Figure 6 Comparison of a pixel value with a set of samples in a 2-D Euclidean color space

$$K = \begin{cases} 0, & N(x) \geq n_{\min} \\ 1, & N(x) < n_{\min} \end{cases} \tag{6}$$

$$N(x) = S_R(v(x)) \cap M(x) \tag{7}$$

3) Background update

The algorithm employs an update strategy that considers time and space to ensure an exponentially decayed, smooth life cycle for each sample. This approach accelerates the detection rate while maintaining spatial consistency. Specifically, if a pixel is consecutively detected in the foreground N times, it is updated as a background point. Each pixel has a probability of $1/\varphi$ to update its own sample set, and there is also a probability of $1/\varphi$ to update the sample set of its neighboring pixels.

4) Morphological post-processing

The Vibe algorithm generally satisfies the requirements of real-time detection and achieves high accuracy. However, there are still limitations observed in the experiments, such as false detection of shadows as foreground and incomplete detection of moving targets. This is primarily because of the similarity between shadows or specific areas of the moving targets and the background, leading to misjudgment. Other objects, such as insects, may also interfere with the detection process. To this end, this study introduced morphological processing methods in addition to the Vibe algorithm. First, connected areas are identified and used to fill holes in the image. Second, median filtering[22] removes isolated noise interference without affecting the boundary contour. The image is subjected to the open operation to eliminate noise outside the image contour, followed by the close operation to eliminate noise inside the foreground object. The open operation erodes and then dilates the image, while the closing operation dilates and erodes it. Finally, hole filling is repeated to complete the background area. The process is illustrated in Figure 7.

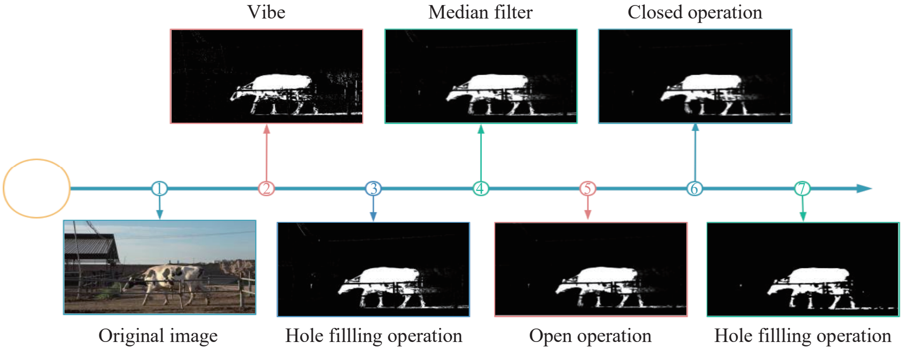


Figure 7 Morphological image processing

2.6 Judgment model for cow lameness

The Vibe algorithm can be used to obtain the pixel area of a cow image, and the OpenCV Python library can accurately extract the boundary between the target and the background (the cow contour). The coordinates of the boundary are stored in a vector container. By utilizing GhostNet_YOLOv4, the cow can be detected and precisely positioned. The coordinates of the detection frame are inputted into the Vibe algorithm's detection result, and each coordinate point in the container is checked to determine if it falls within the detection frame. If it does, it is considered a pixel on the cow's back contour; otherwise, it is discarded. Finally, a polynomial curve equation is fitted to the pixel points of the cow's back contour line through the least squares method.

3 Results and analysis

The coding language used is Python, and the code is written and executed on the Ubuntu operating system. The server is equipped with an NVIDIA K80 GPU. GhostNet_YOLOv4 and its comparison model are developed using the TensorFlow 2.2.0 deep learning framework.

3.1 Evaluation and analysis of the GhostNet_YOLOv4 model

1) Evaluation index

The proposed GhostNet_YOLOv4 model can be used to accurately position the target area, specifically detecting the cow's back area. The average precision (AP) value of the back area detection is used as a performance measure for the algorithm. Metrics such as mean average precision (mAP), number of model parameters, and frames per second (FPS) are comprehensively considered to determine the best detection effect suitable for this experiment.

2) Training results and analysis

This section presents results from training and testing a dataset consisting of 5,000 annotated images of critical parts of dairy cows. GhostNet_YOLOv4, YOLOv4, SSD[23], YOLOv4-tiny[24], Faster-RCNN[25], Efficientdet[26], and other mainstream target detection models were used for training and testing. The results are listed in Table 1.

Table 1 Results of deep learning models

Object detection model	AP						mAP/%	Parameter quantity	FPS/fps
	Head/%	Back/%	LFleg/%	RFleg/%	LHleg/%	RHleg/%			
GhostNet_YOLOv4	98.87	99.04	94.06	96.27	95.44	95.56	96.54	11,165,685	64.95
YOLOv4	98.88	98.95	95.56	97.31	96.25	96.14	97.18	64,106,305	17.46
YOLOv4-tiny	98.21	98.45	93.23	89.81	94.76	92.26	94.45	5,924,214	117.23
SSD	99.08	95.24	96.15	96.23	96.66	94.83	96.37	24,414,218	16.94
Efficientdet	99.05	92.55	94.92	93.46	95.13	92.86	94.66	3,886,253	46.96
Faster-RCNN	98.33	94.88	76.57	75.34	76.85	72.39	82.39	8,627,482	5.36

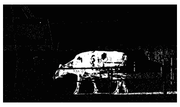
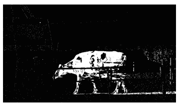
Note: L, R, H, and F represent the left, right, front, and rear, respectively.

Table 1 lists the superior performance of GhostNet_YOLOv4 in back area detection, achieving a detection accuracy of 99.04%. This accuracy is 0.09%, 0.59%, 3.8%, 6.49%, and 4.16% higher than that of YOLOv4, YOLOv4-tiny, SSD, Efficientdet, and Faster-RCNN, respectively. Regarding detecting different parts of dairy cows, GhostNet_YOLOv4, YOLOv4, and YOLOv4-tiny exhibit higher mAP values, surpassing 96%. SSD and Efficientdet follow with mAP values ranging from 94% to 95%, while Faster-RCNN performs the poorest with an mAP of only 82.39%. The number of model parameters, in ascending order, is as follows: YOLOv4, SSD, GhostNet_YOLOv4, Faster-RCNN, YOLOv4-tiny, and Efficientdet. Notably, the GhostNet_YOLOv4 model's parameters are approximately 1/6 of those in YOLOv4. In terms of real-time performance, YOLOv4-tiny and GhostNet_YOLOv4 exhibit higher detection speeds, with FPS values of 117.23 and 64.95, respectively. Efficientdet follows with a speed of 46.96 fps. YOLOv4, SSD, and Faster-RCNN fall short of the test requirements. In conclusion, GhostNet_YOLOv4 is the optimal choice for accurate back position detection of cows while meeting portability requirements for implementation on mobile devices and demonstrating real-time model detection performance.

3.2 Evaluation and analysis of the Vibe model

Table 2 lists the detection and comparison of a cow's walking image sequence using various algorithms, including the Vibe algorithm, dense optical flow method[27], inter-frame difference method, background difference method[28], Gaussian mixture modeling method[29], and other moving target detection algorithms. The average processing time for all images detected by each algorithm is used for comparison. The results in Table 2 reveal that the inter-frame difference method outperforms other algorithms, with the Vibe algorithm ranking second. The addition of partial morphological processing in the Vibe algorithm used in this experiment only slightly increases the processing time while maintaining better processing speed than the dense optical flow method, background difference method, and Gaussian mixture modeling method. On average, the Vibe algorithm achieves a processing time of 0.05 s, meeting the real-time processing requirements.

Table 2 Results of models for extracting pixel area

Object Detection	The original Vibe algorithm	Dense optical flow method	Interframe difference method	Background difference method	Gaussian mixture modelling method	Vibe algorithm used in this study
Image of test results						
The average processing speed of each frame/s	0.046	0.081	0.021	0.214	0.162	0.050

3.3 Evaluation and analysis for the judgment model of cow lameness

By employing the Vibe algorithm in this experiment, each frame of the comprehensive information about the back area of a cow's walking image can be promptly acquired. When combined with the precise positioning provided by the GhostNet_YOLOv4 algorithm, the cow's back area can be isolated within a rectangular frame. Subsequently, the points along the contour line of the cow's back can be obtained, enabling us to fit a curve equation to these points. To assess the density of the sample data point set around the regression line and the fitness of the regression equation, the concept of "Goodness of Fit" was employed[30].

$$N = R^2 \times 100\% \tag{8}$$

$$R^2 = \frac{\text{SSR}}{\text{SST}} = 1 - \frac{\text{SSE}}{\text{SST}} \tag{9}$$

where, the variable N denotes the goodness of fit, while R^2 represents the coefficient of determination, quantifying the proportion of the total variation in the response variable explained by the regression model. SST refers to the total sum of squares, SSR corresponds to the regression sum of squares, and SSE signifies the residual sum of squares.

$$\text{SST} = \sum\nolimits_{i=1}^{n} (y_i - y)^2 \tag{10}$$

$$\text{SSR}=\sum\nolimits_{i=1}^{n} (\hat{y}_i - \bar{y})^2 \tag{11}$$

$$\text{SSE}=\sum\nolimits_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{12}$$

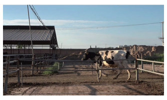
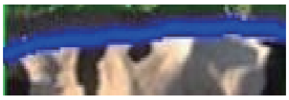
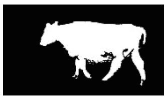
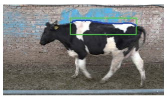
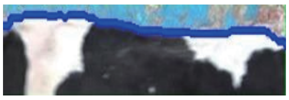
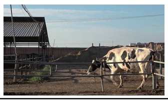
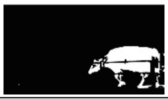
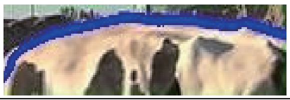
where, y represents the data to be fitted; $\bar{y}$ represents its mean value; $\hat{y}$ is the fitted data.

Tables 3 and 4 list the average goodness of fit for fourth-order

Table 3 Comparison of the goodness of fit

Order of the regression equation	The average value of R^2		
	Normal cows	Early lameness	Moderate-severe lameness
First-order regression equation	0.803 82	0.668 16	0.338 11
Second-order regression equation	0.819 44	0.867 31	0.899 74
Third-order regression equation	0.821 33	0.898 32	0.959 44
Fourth-order regression equation	0.962 47	0.969 32	0.975 51
Fifth-order regression equation	0.962 70	0.970 03	0.978 06

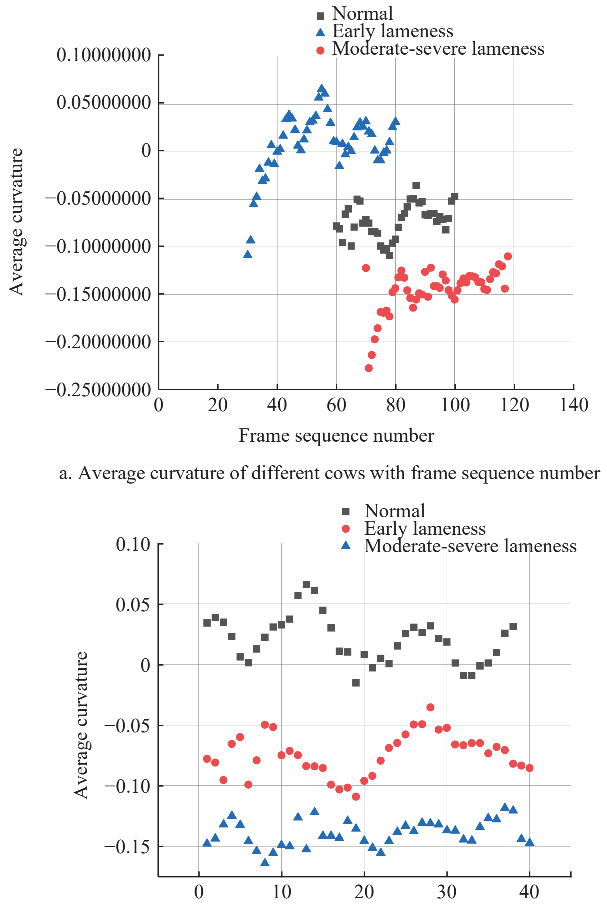
Table 4 Results of different cows for fitting curve

The types of cows	Original image	Target pixel area of cows	Image of results	Results from the back area
Normal cows				
Early lameness				
Moderate-severe lameness				

and fifth-order regression equations applied to normal cows, early lameness, and moderate-severe lameness. The results indicate that the average goodness of fit for the fourth-order and fifth-order regression equations is above 90% for normal cows and early lameness. These findings demonstrate that the selected points used to fit the cow's back contour curve effectively capture the changes in the cow's back during movement. Consequently, the fourth-order regression equation has been chosen as the best fit for the test curve, as its linear feature accurately represents the back contour of a cow.

During the experiment, It was observed that the average curvature of each point on the cow's back contour line gradually increased when more contour points were detected. Once the cow's back contour line is completely visible within the detection range, the average curvature exhibits a regular waveform change as the cow walks, As shown in Figure 8a. To clearly distinguish the average curvature among normal cows, early lameness, and moderate-severe lameness, the following definitions were established in this study: the start point is when the back becomes fully visible in the image, and the endpoint is when the cow's head reaches the image edge. Subsequently, the change data of the average curvature were collected for each frame during a 480-frame sequence. As shown in Figure 8b. Based on this data, the average curvature intervals were established as follows: [−0.125, −0.025] for normal cows walking, (−0.025, +∞] for early lameness, and [−∞, −0.125) for moderate-severe lameness.

Using the obtained classification threshold, a 120-segment test set was tested and achieved an overall accuracy rate of 91.67%. Specifically, 36 correctly classified positive cases of normal cows resulting in an accuracy rate of 90%. For early lameness, 35 positive cases were accurately identified, resulting in an accuracy rate of 87.5%. Lastly, there were 39 positive cases of moderate-severe lameness cows, with an accuracy rate of 97.5%.



a. Average curvature of different cows with frame sequence number

b. Average curvature of different cows in a specific period

Figure 8 Changes in the average curvature of different cows with a frame sequence number and the average curvature of different cows in a specific period

4 Conclusions

This study employed the GhostNet_YOLOv4 model, the Vibe algorithm, and machine vision technology to develop a cow limp detection model and address the challenges posed by the untimeliness and usability of existing artificial limp detection methods. The key findings of this research are outlined below:

1) The critical part of the cow detection dataset was used to detect the back area of cows through various deep learning target detection algorithms, such as GhostNet_YOLOv4, Efficientdet, YOLOv4, SSD, and Faster-RCNN. Comparative analysis of the detection outcomes reveals the superior performance of the

GhostNet_YOLOv4 model in terms of real-time capability, accuracy, and portability. It achieves an impressive back area detection accuracy of 99.04% in terms of AP, satisfying the requirements for routine back detection and positioning.

2) The image sequence dataset capturing cows in motion was employed to determine the pixel area of cows through the utilization of motion target detection algorithms, including the Vibe algorithm, dense optical flow method, inter-frame difference method, background difference method, Gaussian mixture modeling method, and others. In the conducted tests, the improved Vibe algorithm outperformed other detection algorithms. With an average processing time of 0.05 s/frame, it satisfies the requirement of real-time detection and proves to be more suitable for pixel area detection in cows.

3) The combined detection results of GhostNet_YOLOv4 and the improved Vibe algorithm were utilized to determine the back contour of cows and fit a fourth-order curve regression equation. The resulting average goodness of fit values for normal cows, early lameness, and moderate-severe lameness were 0.962 47, 0.969 32, and 0.975 51, respectively. These findings demonstrated the effectiveness and feasibility of the experimental method in describing the lameness characteristics of dairy cows.

4) Through multiple experiments, the thresholds for different ranges of normal cows, early lameness, and moderate-severe lameness are established by analyzing the change in average curvature with the number of frames over a specific period. The determined thresholds are [−0.125, −0.025], (−0.025, +∞], and [−∞, −0.125], respectively. By utilizing these thresholds, a detection accuracy rate of 91.67% was achieved for detecting lameness in dairy cows. This indicates the effectiveness and feasibility of the proposed approach for detecting lameness in dairy cows.

Acknowledgements

This work was supported by Shanxi Province Basic Research Program (Free Exploration) Project (No. 202103021224149), Shanxi Province Postgraduate Education Teaching Reform Project (2021YJJG087), Shanxi Province Educational Science “14th Five-Year Plan” Education Evaluation Special Project (PJ-21001) funded.

[References]

[1] Han S Q, Zhang J, Cheng G D, Peng Y Q, Zhang J H, Wu J Z. Current state and challenges of automatic lameness detection in dairy cattle. Smart Agriculture, 2020; 2(3): 21–36. (in Chinese)

[2] Kang X, Zhang X D, Liu G. A review: Development of computer vision-based lameness detection for dairy cows and discussion of the practical applications. Sensors, 2021; 21(3): 753.

[3] Novotna I, Langova L, Havlicek Z. Risk factors and detection of lameness using infrared thermography in dairy cows - A review. Annals of Animal Science, 2019; 19(3): 563–578.

[4] Sprecher D J, Hostetler D E, Kaneene J B. A lameness scoring system that uses posture and gait to predict dairy cattle reproductive performance. Theriogenology, 1997; 47(6): 1179–1187.

[5] Thomsen P T, Munksgaard L, Tøgersen F A. Evaluation of a lameness scoring system for dairy cows. Journal of Dairy Science, 2008; 91(1): 119–126.

[6] Van Nuffel A, Zwertvaegher I, Van Weyenberg S, Pastell M, Thorup V M, Bahr C, et al. Lameness detection in dairy cows: Part 2. Use of sensors to automatically register changes in locomotion or behavior. Animals, 2015; 5(3): 861–885.

[7] Chapinal N, de Passillé A M, Pastell M, Hanninen L, Munksgaard L, Rushen J. Measurement of acceleration while walking as an automated method for gait assessment in dairy cattle. Journal of Dairy Science, 2011; 94(6): 2895–2901.

[8] Bahr C, Leroy T, Song X Y, Vranken E, Maertens W, Vangeyte J, et al. Automatic detection of lameness in dairy cattle - Analyzing image parameters related to lameness. Livestock Environment VIII, Iguassu Falls: ASABE, 2008; paper No. 701P0408. doi: 10.13031/2013.25608.

[9] Song H B, Jiang B, Wu Q, Li T, He D J. Detection of dairy cow lameness based on fitting line slope feature of head and neck outline. Transactions of the CSAE, 2018; 34(15): 190–199. (in Chinese)

[10] Zhao K, Bewley J M, He D, Jin X. Automatic lameness detection in dairy cattle based on leg swing analysis with an image processing technique. Computers and Electronics in Agriculture, 2018; 148: 226–236.

[11] Van Hertem T, Viazzi S, Steensels M, Maltz E, Antler A, Alchanatis V, et al. Automatic lameness detection based on consecutive 3D-video recordings. Biosystems Engineering, 2014; 119: 108–116.

[12] Viazzi S, Bahr C, Schlageter-Tello A, Van Hertem T, Romanini C E B, Pluk A, et al. Analysis of individual classification of lameness using automatic measurement of back posture in dairy cattle. Journal of Dairy Science, 2013; 96(1): 257–266.

[13] Poursaberi C, Bahr C, Pluk A, Van Nuffel A, Berckmans D. Real-time automatic lameness detection based on back posture extraction in dairy cattle: Shape analysis of cow with image processing techniques. Computers and Electronics in Agriculture, 2010; 74(1): 110–119.

[14] Zhang H Y, Cisse M, Dauphin Y N, Lopez-Paz D. Mixup: Beyond empirical risk minimization. arXiv Preprint, 2018. arXiv: 1710.09412.

[15] Devries T, Taylor G W. Improved regularization of convolutional neural networks with cutout. arXiv Preprint, 2017. arXiv: 1708.04552.

[16] Yun S, Han D, Chun S, Oh S J, Yoo Y, Choe J. CutMix: Regularization strategy to train strong classifiers with localizable features. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul: IEEE, 2019; pp.6022–6031. doi: 10.1109/ICCV.2019.00612.

[17] Buslaev A, Iglovikov V I, Khvedchenya E, Parinov A, Druzhinin M, Kalinin A A. Albumentations: Fast and flexible image augmentations. Information, 2020; 11(2): 125.

[18] Han K, Wang Y H, Tian Q, Guo J Y, Xu C J, Xu C. GhostNet: More features from cheap operations. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle: IEEE, 2020; pp.1577–1586. doi: 10.1109/CVPR42600.2020.00165.

[19] Bochkovskiy A, Wang C Y, Liao H Y M. YOLOv4: Optimal speed and accuracy of object detection. arXiv Preprint, 2020. arXiv: 2004.10934.

[20] Jiang B R, Luo R X, Mao J Y, Xiao T T, Jiang Y N. Acquisition of localization confidence for accurate object detection. arXiv Preprint, 2018. arXiv: 1807.11590.

[21] Barnich O, Van Droogenbroeck M. ViBe: A universal background subtraction algorithm for video sequences. IEEE Transactions on Image Processing, 2011; 20(6): 1709–1724.

[22] Dudi B, Rajesh V. Optimized threshold-based convolutional neural network for plant leaf classification: a challenge towards untrained data. Journal of Combinatorial Optimization, 2022; 43: 312–349.

[23] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C Y, et al. SSD: Single Shot Multibox Detector. Computer Vision – ECCV 2016. 2016; 9905: 21–37. doi: 10.1007/978-3-319-10578-9_23.

[24] Jiang Z C, Zhao L Q, Li S Y, Jia Y F. Real-time object detection method based on improved yolov4-tiny. arXiv Preprint, 2020. arXiv. 2011.04244.

[25] Ren S Q, He K M, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017; 39(6): 1137–1149.

[26] Tan M X, Pang R M, Le Q C. EfficientDet: Scalable and Efficient Object Detection. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle: IEEE, 2020; pp.10778–10787. doi: 10.1109/CVPR42600.2020.01079.

[27] Farnebck G. Two-grame motion estimation based on polynomial expansion. 13th Scandinavian Conference on Image Analysis (SCIA 2003), Springer-Verlag, 2003; pp.363-370. doi: 10.1007/3-540-45103-X_50.

[28] Muro S, Yoshida I, Hashimoto M, Takahashi K. Moving-object detection and tracking by scanning LiDAR mounted on motorcycle based on dynamic background subtraction. Artificial Life and Robotics, 2021; 26: 412–422.

[29] Wang, Y, Tan Y H, Tian J W. Video segmentation algorithm with Gaussian mixture model and shadow removal. Opto-Electronic Engineering, 2008; 3: 21–25.

[30] Tamara F, Nicolas R, Xu W K, Arthur G. Kernelized stein discrepancy tests of goodness-of-fit for time-to-event data. arXiv Preprint, 2020. arXiv: 2008.08397v2.