Detecting Mounting Behaviors of Dairy Cows by Pre-Training with Pseudo Images

Yuta Okuda[1 [0000−0003−1847−1882]], Yota Yamamoto[1 [0000−0002−1679−5050]], Kazuaki Nakamura[1 [0000−0002−4859−4624]], and Yukinobu Taniguchi[1 [0000−0003−3290−1041]]

Tokyo University of Science, Tokyo, Japan
4621505@ed.tus.ac.jp, {yy-yamamoto, nakamura.kazuaki, taniguchi.yukinobu}@rs.tus.ac.jp

Abstract. A key part of improving the productivity of the dairy industry is detecting signs of estrus in dairy cows. In this paper, we propose a method based on deep learning to automatically detect mounting behavior, an indicator of estrus, using cameras installed on the ceiling of the barn. Mounting behavior occurs rarely, and it is virtually impossible to manually collect actual data of mounting events. The novelty of the proposed method lies in the pre-training scheme, which uses pseudo-mounting images generated by overlapping randomly selected cow images that are easily collected. The pre-trained model is then fine-tuned on a small amount of actual mounting data. We introduce a consistency regularization term based on using background replacement images to reduce the influence of background changes in pre-training. We show the effectiveness of the proposed method through experiments that compare the proposed method and previous self-supervised learning methods in terms of their detection accuracy using data collected in actual cattle barns.

Keywords: Self-Supervised Learning · Anomaly Detection · Image Generation.

1 Introduction

Recently, the number of dairy farms in Japan has been decreasing due to the drop in the population of dairy farmers because of aging and the lack of successors [11]. However, the demand for dairy products remains constant. For higher production efficiency, dairy farms are getting larger in scale. To improve dairy production, it is necessary to increase the rate of estrus detection and thus pregnancy. Dairy farmers can efficiently increase the number of dairy cows by artificially inseminating those dairy cows in estrus. The signs of estrus appear in cow behavior (e.g., mounting, accepting mounting, walking around, etc.), that is dairy cows during the estrus period tend to mount other cows and accept mounting, so dairy farmers need to observe each dairy cow carefully. However, the burden of managing individual dairy cows has become significant with the scale of herds.

Fig. 1: Mounting image.

One way to reduce the burden on dairy farmers is to attach acceleration sensors to dairy cows. The acceleration sensor detects signs of estrus by measuring the cow movements. However, the sensor, which is attached to the neck of each cow, is costly and fails often as dairy cows love to rub against the wall. In addition, attaching them to dairy cows causes stress. In this paper, we focus on the individual management of dairy cows by installing cameras on the ceiling of the barn (ceiling camera).

One way to detect signs of estrus is to detect mounting behavior (Fig. 1) in which one dairy cow mounts another dairy cow. Wang et al. [15] used the object detector YOLOv5 to detect mounting behaviors. However, it requires a large amount of data on mounting behavior for training. Unfortunately, it is time-consuming to collect the large amount of training data needed because mounting behavior rarely occurs. Fortunately, it is easy to prepare a large amount of non-mounting data from barn images.

To address the problem, this paper proposes a method of detecting mounting behavior that uses for pre-training i) pseudo-mounting images generated by overlapping randomly selected cow images (which are easily collected), and ii) a consistency regularization loss term based on background replacement images. The pre-trained model is then fine-tuned on a small amount of actual mounting data.

2 Related Work

2.1 Mounting Behavior Detection

Several methods have been developed for detecting the mounting behaviors of cows or pigs. Most of them are based on region features. Nasirahmad et al. [8] use an ellipse fitting technique to locate the position of pigs and detect mounting behaviors from the distance between each pig. Li et al. [5] use Mask R-CNN [3] to detect specific regions of pigs from which mounting behavior is identified. From

the detection results, three features, length around the pig, region of the half of the mask, and distance between the centers of the bounding boxes (BBOX), are selected. The eigenvectors are classified by a kernel extremal learning machine (KELM) to detect mounting behavior. Noe et al. [9] proposed a method for detecting the mounting behaviors of dairy cows that uses object detection and tracking techniques. It extracts segmented regions of dairy cows by using Mask R-CNN while a lightweight tracking algorithm is used to detect mounting behavior. It takes advantage of the fact that the body region of a cow rises up when mounting. These methods detect mounting from just the region features of cows and pigs.

Wang et al. [15] proposed a detection method based on mounting-specific postures (image features) using improved YOLOv5, which has stronger detection ability against complex environments and multi-scale objects. However, this method requires a large amount of manually annotated mounting data.

2.2 Self-Supervised Learning

There are many studies on self-supervised learning to deal with the lack of training data.

CutPaste [4] is an anomaly detection model based on a two-stage framework with self-supervised learning. The first step is to learn a deep representation using data augmentation. Data augmentation is simple: cut an image patch and paste it at a random position in the original image. Next, a classifier builds on the learned deep representation in one class. The original image is defined as normal, and the data augmentation image is defined as abnormal. Inferencing is performed using the Gaussian density estimator to find anomalies, and then Grad-CAM [12] detects the locations of anomalies.

Noroozi et al. [10] proposed a method for learning image representations through a task in which models solve jigsaw puzzles. The model is trained by inputting images divided into tiles whose positions are swapped and predicting the original order. Therefore, the images do not need to be labeled. The learned models can be reused for tasks such as object detection and classification.

SimMIM [16] learns an image representation through the task of predicting the original image from a randomly masked image. Masking is done at patch level, that is, a patch is either fully visible or completely masked. The encoder uses the transformer model [2, 7]. The prediction head can be as light as linear, and the image representation is learned by regression of the RGB values.

This paper improves the accuracy for actual tasks by performing self-supervised learning specifically for mounting detection.

3 Proposed Method

The main idea behind our proposed method is to generate pseudo mounting images from dairy cow images (non-mounting images), which are easy to collect.

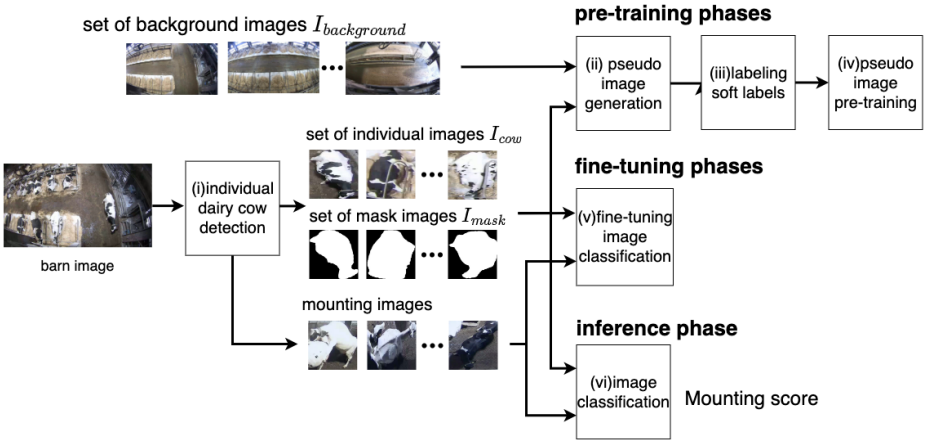


Fig. 2: Proposed method

We draw inspiration from the CutPaste [4] method. However, instead of randomly cutting out rectangular image patches, we use Mask R-CNN to extract cow regions that are then cut out and pasted.

The proposed method is shown in Fig. 2. It comprises six processes: (i) individual dairy cow detection, (ii) pseudo-image generation, (iii) assigning soft labels, (iv) pseudo image pre-training, (v) fine-tuning image classification, and (vi) image classification.

(i) **Individual dairy cow detection.** Mask R-CNN takes as input a barn image taken by a ceiling camera and detects individual cows $I_{cow} = \{I_1, I_2, \cdots, I_n\}$ and the corresponding segmentation masks $M_{cow} = \{M_1, M_2, \cdots, M_n\}$. Mask R-CNN used is fine-tuned using barn data.

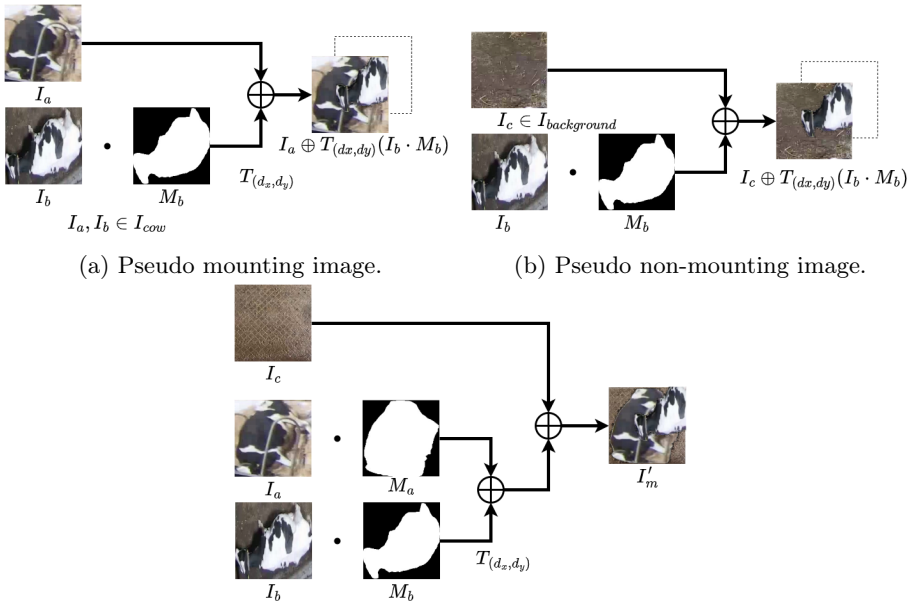
(ii) **Pseudo-image generation.** We generate pseudo images to be used for training from individual images I_{cow} and masks M_{cow}. The pseudo-image generation methods are described in Section 3.1.

(iii) **Assigning soft labels.** We assign soft labels, the likelihood of mounting behavior, to the pseudo images. The method of assigning soft labels is explained in Section 3.2.

(iv) **Pseudo image pre-training.** The images and labels generated in (ii) and (iii) are used to train the classifier. The structure of the classifier used is described in Section 3.3.

(v) **Fine-tuning image classification.** The pre-trained classifier model is fine-tuned on a small number of mounting images and a large number of other images. Details of this process are shown in Section 3.4.

(vi) **Image classification.** The classifier takes an individual image detected by (i) as input and outputs a real number ranging from 0 to 1, which indicates the likelihood of mounting behavior. If the score is higher than a threshold value, the cow image is considered to contain mounting behavior.



(a) Pseudo mounting image.

(b) Pseudo non-mounting image.

(c) Background replacement image.

Fig. 3: Pseudo-image generation.

3.1 Pseudo-Image Generation

The method for generating pseudo images is shown in Fig. 3. There are three types of pseudo images: pseudo mounting images, pseudo non-mounting images, and background replacement images. Fig. 4 shows examples of (a) pseudo mounting images, (b) pseudo non-mounting images, (c)(d) background replacement images for (a), (b).

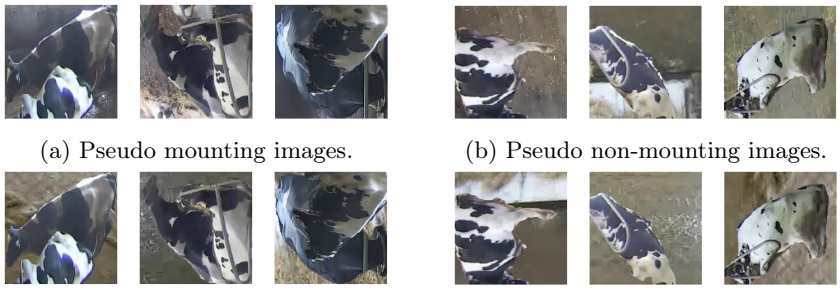
Pseudo-Mounting Images: To offset the paucity of mounting images, pseudo-mounting images are generated. As in Fig. 3(a), pseudo-mounting image I_m is generated as follows:

$$I_m = I_a \oplus T_{(d_x,d_y)}(I_b \cdot M_b), \tag{1}$$

where cow image $I_a, I_b \in \mathcal{I}_{cow}$ is randomly selected, mask image M_a, M_b corresponding to I_a, I_b, $A \oplus B$ is the result of replacing the image value of image A with that of image B, $\cdot$ is the logical product of the images, and $T_{(d_x,d_y)}$ is the operation of shifting the image by (d_x, d_y). The image I_a to be pasted is called the base image and the image I_b to be pasted is called the overlapping image.

So that the overlapping ratio s is uniformly distributed, we determine displacement vector (d_x, d_y) as follows,:

$$d_x = \pm(1 - x_{ratio})W, \quad d_y = \pm(1 - y_{ratio})H, \tag{2}$$



(a) Pseudo mounting images.

(b) Pseudo non-mounting images.

(c) Background replacement images of (a).

(d) Background replacement images of (b).

Fig. 4: Example of pseudo images.

where (W, H) indicates image size, sign is chosen randomly, and overlapping ratio $s \sim U(0, 1)$, $x_{ratio} \sim U(s, 1)$, $y_{ratio} = s/x_{ratio}$.

Pseudo Non-Mounting Images: When a classifier is trained with both pseudo-mounting images and real non-mounting images, it learns to detect pasting seams. To avoid this problem, pseudo non-mounting images are generated and added for training. Background images $I_{background}$ are the collection of barn images captured when there are no cows.

The pseudo non-mounting images are generated by selecting one from each dairy cow I_{cow} and one from each background image $I_{background}$. The selected background image is resized to 224×224 after random cropping.

$$I_n = I_c \oplus T_{(d_x,d_y)}(I_b \cdot M_b), \tag{3}$$

where $I_c \in \mathcal{I}_{background}, I_b \in \mathcal{I}_{cow}$.

Background Replacement Images: To make the image classifier pay attention to the pose of dairy cows instead of the background, we generate background replacement images from pseudo-mounting and pseudo non-mounting images as follows:

$$I'_m = I_c \oplus (I_a \cdot M_a) \oplus T_{(d_x,d_y)}(I_b \cdot M_b), \tag{4}$$

$$I'_n = I_c \oplus T_{(d_x,d_y)}(I_b \cdot M_b), \tag{5}$$

where I_a, I_b are the same as the generated-pseudo image and $I_c \in \mathcal{I}_{background}$.

3.2 Assigning Soft Labels

The proposed method assigns a soft label to a pseudo image that indicates the likelihood of mounting behavior. We assign soft label y to pseudo-mounting image I_m generated by eq.(1) as follows:

$$y = \frac{\left|M_a \cap T_{(d_x,d_y)}(M_b)\right|}{\left|M_a \cup T_{(d_x,d_y)}(M_b)\right|}, \tag{6}$$

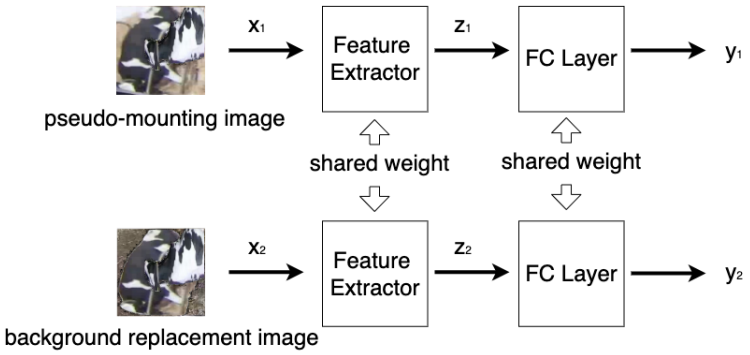


Fig. 5: Image classifier.

which indicates the Intersection over Union (IoU) between the masked regions of two cows. The soft label of the pseudo non-mounting image is always set to 0. The background replacement image is given the same label as the original pseudo image.

3.3 Pseudo image pre-training

We train an image classifier on the pseudo images generated above and the soft labels. The image classifier outputs a score ranging from 0 to 1 indicating the likelihood of mounting behavior. As illustrated in Fig. 5, the classifier model is composed of feature extraction and fully-connected layers (FC Layer).

Feature Extraction: Pseudo image $\boldsymbol{x}_1$ and the corresponding background replacement image $\boldsymbol{x}_2$ are input to the feature extractor which outputs feature values $\boldsymbol{z}_1, \boldsymbol{z}_2$. The weights of the feature extractors are shared.

FC Layer: Inputs features $\boldsymbol{z}_1, \boldsymbol{z}_2$ to the FC layer + sigmoid and outputs the score y'_1, y'_2.

We use two types of loss: binary cross entropy loss L_{score} and consistency regularization loss $L_{consistency}$. The binary cross entropy loss (BCELoss) is defined by:

$$L_{score} = \sum_{n=1}^{2} y \log y'_n + (1-y)\log(1-y'_n), \tag{7}$$

where y is the value of the soft label of the image. The consistency regularization loss is defined by:

$$L_{consistency} = \|\boldsymbol{z}_1 - \boldsymbol{z}_2\|^2, \tag{8}$$

The loss constrains the classifier to output the same features for a pseudo image and the background replacement image. The total loss is $L = L_{score} + L_{consistency}$.

3.4 Fine-tuning image classification

We fine-tune the pre-trained model on a small number of mounting images and a large number of non-mounting images. The images are assigned binary labels instead of soft ones, 1 for mounting images and 0 otherwise. Focal loss [6] is employed to reduce the effect of data imbalance and is defined by:

$$L_{focal} = -(1 - p_t)^{\gamma} \log(p_t), \tag{9}$$

where $p_t = y'$ if $y = 1$, otherwise $p_t = 1 - y'$, γ is a focusing parameter.

4 Experimental Settings

4.1 Dataset

We used barn image data captured by 13 ceiling cameras installed in an actual barn. We prepared a dataset of individual images, which were cropped barn images of individuals detected by Mask R-CNN. The dataset was manually annotated and contains images of mounting behaviors. Table 1 shows the number of individual images used for pre-training, fine-tuning, and testing.

Pre-training: We prepared two datasets (pseudo and individual image dataset) for pre-training. For each epoch, both pseudo-mounting and pseudo-non-mounting images were generated with a probability of 50% by the method described in Sec. 3.1. Thirteen empty barn images (no cows) were used as sources of background images.

Fine-tuning and test: The fine-tuning and testing dataset consisted of mounting images and other images. We split the dataset into fine-tuning and test subsets so that temporally consecutive images were not present in either subset.

4.2 Evaluation Metric

The evaluation metric is Area Under the Curve (AUC), which is the value of the area under the ROC curve. The ROC curve is a plot of the true positive rate (TPR) on the vertical axis and the false positive rate (FPR) on the horizontal axis, with varying threshold values. Since the number of mounting images is small, the experiment was conducted using three-fold cross validation.

Table 1: Number of individual images.

dataset		non-mounting	mounting
pretrain	pseudo image	3349	-
	individual image	23431	-
fine-tuning		5547	124
test		5892	

4.3 Implementation Details

The implementation details are as follows.

Pre-training: We used Adam optimizer with the learning rate of 0.0001. The number of epochs was 100, and the image size was 224×224. EfficientNet-B0 [14] was used as the feature extractor; it output 1,000 dimensional features and was trained by ImageNet [1].

Fine-tuning: The learning rate was 0.00001, and focusing parameter γ was 2. The other conditions were the same as in pre-training, and the number of epochs was 30.

4.4 Baseline Methods

This paper compares the following five methods: the proposed method, data augmentation,two self-supervised methods, Jigsaw [10] and SimMIM [16], and a bbox-based method.

Proposed Method: To evaluate the impact of pre-training with pseudo-images, we evaluated two methods: with and without pre-training, and with unsupervised learning but using only pseudo-images.

Data Augmentation: We used a simple data augmentation method instead of pseudo-images. Three data augmentations were used: random rotation(90-degree increments), random flip, and color jitter(all parameters ranged from to ± 0.2). The training model was similar to the proposed method.

Jigsaw: Jigsaw [10] is a pre-training method that solves jigsaw puzzles. We compared three different methods: pre-training on a pseudo-image dataset, pre-training on an individual image dataset, and no pre-training. With regard to implementation details, the image was divided into 3×3 and 250 different puzzle patterns. The number of epochs of pre-training was 300 . Other details followed those in 4.3.

SimMIM: SimMIM [16] is a pre-training method that predicts the mask portion given to an image. We compared three settings following the Jigsaw test above. We employed the transformer model VIT [2]. For pre-training, images were masked with a probability of 0.6, where the mask patch size was 32. Other details followed those in 4.3.

Bbox-Based Method: Since it is difficult to reproduce the previous method[5] accurately, we implemented a simple method that used the positional relationship of the detected bounding boxes. Taking $B_i (i = 1, 2, \cdots)$ to be the bounding boxes of individual dairy cows detected in barn images, we computed score S_i as the maximum of $IoU(B_i, B_j)(j \neq i)$. If score S_i exceeds a threshold, bounding box B_i is judged to contain mounting behavior. The threshold parameter was chosen empirically instead of fine-tuning based on training data.

5 Results

Table 2 shows the experimental results and Fig. 6 shows the ROC curves. Our method achieved the highest AUC value of 0.914 when pre-trained on pseudo-mounting images and fine-tuned on actual mounting and non-mounting images.

Table 2: Experimental results.

method	pre-training	fine-tuning	AUC
Bbox-based method	-	-	0.578
Jigsaw [10]	-	✓	0.777
	pseudo image	✓	0.799
	individual image	✓	0.746
SimMIM [16]	-	✓	0.794
	pseudo image	✓	0.772
	individual image	✓	0.750
Data augmentation	-	✓	0.888
Ours	-	✓	0.856
	pseudo image	-	0.759
	pseudo image	✓	**0.914**

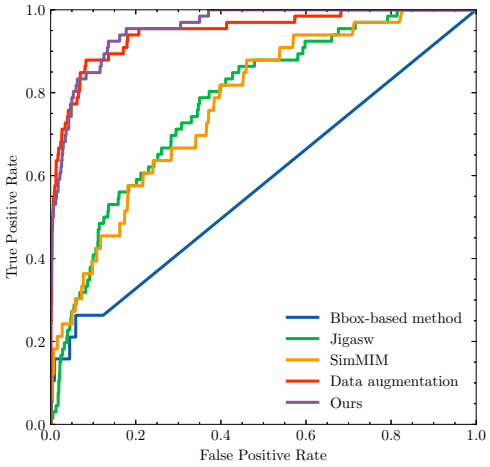
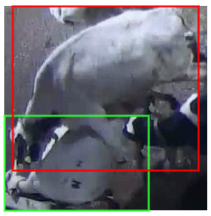


Fig. 6: ROC curves.

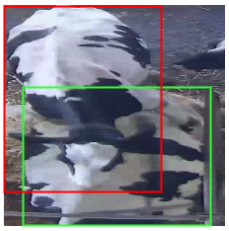
By assessing our method in different settings, we see a 15.5 point improvement with fine-tuning, and a 5.8 point improvement with pre-training. This shows the effectiveness of pre-training and fine-tuning. Although pseudo-mounting images are not an accurate representation of mounting images, they aided mounting behavior detection by providing an appropriate task in pre-training.

Comparing pseudo and individual images used for pre-training, we can see that the accuracy of both Jigsaw and SimMIM pre-trained on pseudo images was better than those pre-trained on individual images. It shows that the pseudo-image is more effective in extracting the image features necessary for mounting detection. Whereas, there is no significant difference between the two methods with and without pre-training. This is probably because the pre-training task is very different from mounting detection and does not make up for the lack of data.

(a) Successful cases attained by pre-training without fine-tuning.


(b) Successful cases with fine-tuning

(c) Failure cases.

Fig. 7: Example of prediction results (Red bboxes show mounting cows. Green bboxes show cows accepting mounting).

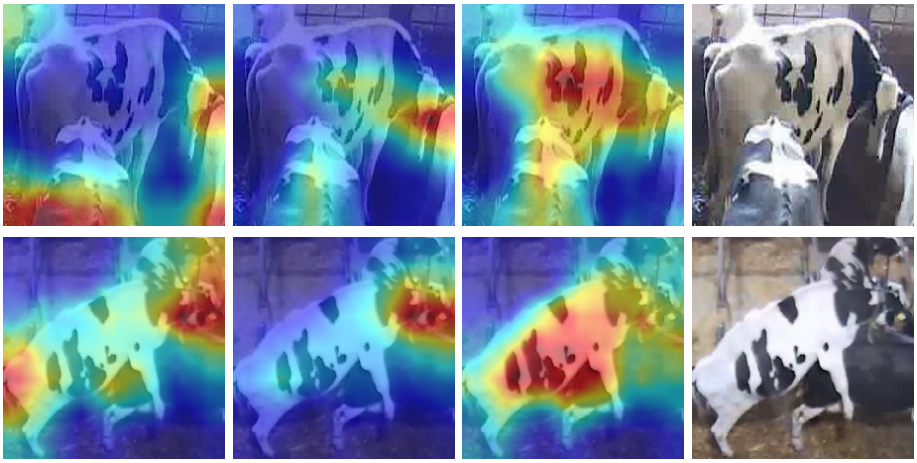
Fig. 7 shows the examples of successful and failure cases. Fig. 7(a) shows the example of images that have already been successfully detected without fine-tuning, trained using only pseudo-images. The positional relationship between the two dairy cows is easy to understand, so the detection could be done by learning only pseudo-images. Fig. 7(b) is an example of failure without fine-tuning. The failure can be caused by the difficulty of generating pseudo-mounting images shot from a side view, as depicted in Fig. 7(b). Fine-tuning is necessary to learn poses that are difficult to generate. The failure cases shown in Fig. 7(c) could be caused by the misidentification of two mottled cows as one cow.

We conducted ablation experiments to evaluate the effect of background replacement images and $L_{consistency}$. Table 3 shows the results of the experiments. Comparing our method in different settings, we see a 4.4 point improvement with background replacement, and a 1.3 point improvement with $L_{consistency}$. The accuracy without background replacement is equivalent to that without pre-training. By training with both pseudo images and background replacement images, it is possible to extract background-independent image features. Furthermore, by adding $L_{consistency}$, common image features such as the positioning of dairy cows could be extracted.

Fig. 8 show the saliency maps generated by Full-Gradient [13] on the pre-trained model without fine-tuning. Comparing the saliency maps obtained by our methods (a), (b), and (c), we can see in Fig. 8(c) that the proposed method (ours(c)) tends to successfully give attention to the body of dairy cows. As in Fig. 8(a) and (b), the model pre-trained without using background replacement

Table 3: Ablation study

method	background replacement	loss	AUC
ours(a)	-	L_{score}	0.857
ours(b)	✓	L_{score}	0.901
ours(c)	✓	$L_{score} + L_{consistency}$	**0.914**



(a) Ours(a). (b) Ours(b). (c) Ours(c). (d) Input.

Fig. 8: Results of saliency map.

images or consistency loss gives attention to the background region outside the cow region. The differences in the attention acquired in the pre-training phase may have affected the mounting detection performance after fine-tuning.

6 Conclusions

In this paper, we proposed a method for the automatic detection of cow mounting behavior. The proposed two-phase learning scheme drastically reduces the burden of capturing mounting behaviors, which are a rare occurrence. In the first phase, we train the model on a large number of pseudo-images generated from two dairy cow images. In the second phase, we fine-tune the pre-trained model on a small number of actual mounting images. In future work, we will develop synthesis methods that can generate more realistic mounting behaviors by taking into account dairy cow pose rather than random synthesis. Furthermore, to further reduce the burden of data collection, unsupervised methods that do not require actual mounting images need to be developed.

Acknowledgements The authors thank the members of Tsuchiya Manufacturing Co. Ltd. for helpful discussions and for providing the video data of barns. This work was supported by JSPS KAKENHI Grant Number JP20K12115.

References

1. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
2. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: International Conference on Learning Representations (2021)
3. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
4. Li, C.L., Yoon, J., Sohn, K., Pfister, T.: CutPaste: Self-Supervised Learning for Anomaly Detection and Localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
5. Li, D., Chen, Y., Zhang, K., Li, Z.: Mounting Behaviour Recognition for Pigs Based on Deep Learning. Sensors **19**(22) (2019)
6. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
7. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022 (2021)
8. Nasirahmadi, A., Hensel, O., Edwards, S.A., Sturm, B.: Automatic detection of mounting behaviours among pigs using image analysis. Computers and Electronics in Agriculture **124**, 295–302 (2016)
9. Noe, S.M., Zin, T.T., Tin, P., Kobayashi, I.: Automatic detection and tracking of mounting behavior in cattle using a deep learning-based instance segmentation model. Int. J. Innov. Comput. Inf. Control **18**, 211–220 (2022)
10. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: European conference on computer vision. pp. 69–84. Springer (2016)
11. Sato, M., Kato, H., Noguchi, M., Ono, H., Kobayashi, K.: Gender differences in depressive symptoms and work environment factors among dairy farmers in japan. International journal of environmental research and public health **17**(7), 2569 (2020)
12. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: Visual explanations from deep networks via gradient-based localization. In: ICCV (2017)
13. Srinivas, S., Fleuret, F.: Full-Gradient Representation for Neural Network Visualization. In: Advances in Neural Information Processing Systems (2019)
14. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning. pp. 6105–6114. PMLR (2019)

15. Wang, R., Gao, Z., Li, Q., Zhao, C., Gao, R., Zhang, H., Li, S., Feng, L.: Detection Method of Cow Estrus Behavior in Natural Scenes Based on Improved YOLOv5. Agriculture **12**(9), 1339 (2022)
16. Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., Hu, H.: SimMIM: A Simple Framework for Masked Image Modeling. In: International Conference on Computer Vision and Pattern Recognition (2022)