

Article

Comparing State-of-the-Art Deep Learning Algorithms for the Automated Detection and Tracking of Black Cattle

Su Myat Noe ¹, Thi Thi Zin ^{2,*} , Pyke Tin ²  and Ikuo Kobayashi ³

¹ Interdisciplinary Graduate School of Agriculture and Engineering, University of Miyazaki, Miyazaki 889-2192, Japan

² Graduate School of Engineering, University of Miyazaki, Miyazaki 889-2192, Japan

³ Field Science Center, Faculty of Agriculture, University of Miyazaki, Miyazaki 889-2192, Japan

* Correspondence: thithi@cc.miyazaki-u.ac.jp

Abstract: Effective livestock management is critical for cattle farms in today's competitive era of smart modern farming. To ensure farm management solutions are efficient, affordable, and scalable, the manual identification and detection of cattle are not feasible in today's farming systems. Fortunately, automatic tracking and identification systems have greatly improved in recent years. Moreover, correctly identifying individual cows is an integral part of predicting behavior during estrus. By doing so, we can monitor a cow's behavior, and pinpoint the right time for artificial insemination. However, most previous techniques have relied on direct observation, increasing the human workload. To overcome this problem, this paper proposes the use of state-of-the-art deep learning-based Multi-Object Tracking (MOT) algorithms for a complete system that can automatically and continuously detect and track cattle using an RGB camera. This study compares state-of-the-art MOTs, such as Deep-SORT, Strong-SORT, and customized light-weight tracking algorithms. To improve the tracking accuracy of these deep learning methods, this paper presents an enhanced re-identification approach for a black cattle dataset in Strong-SORT. For evaluating MOT by detection, the system used the YOLO v5 and v7, as a comparison with the instance segmentation model Detectron-2, to detect and classify the cattle. The high cattle-tracking accuracy with a Multi-Object Tracking Accuracy (MOTA) was 96.88%. Using these methods, the findings demonstrate a highly accurate and robust cattle tracking system, which can be applied to innovative monitoring systems for agricultural applications. The effectiveness and efficiency of the proposed system were demonstrated by analyzing a sample of video footage. The proposed method was developed to balance the trade-off between costs and management, thereby improving the productivity and profitability of dairy farms; however, this method can be adapted to other domestic species.

Keywords: cattle detection; cattle tracking; deep learning; multi-object tracking; precision livestock farming; re-identification



Citation: Myat Noe, S.; Zin, T.T.; Tin, P.; Kobayashi, I. Comparing State-of-the-Art Deep Learning Algorithms for the Automated Detection and Tracking of Black Cattle. *Sensors* **2023**, *23*, 532. <https://doi.org/10.3390/s23010532>

Academic Editor: Yongwha Chung

Received: 1 December 2022

Revised: 28 December 2022

Accepted: 29 December 2022

Published: 3 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Cattle are one of the most commonly raised livestock and a primary protein source for people across numerous cultures and geographic regions. Managing the health of cattle improves productivity and is ideally performed by monitoring individual cows. Moreover, behavior has been well-established as a strong indicator of cattle health. However, manually monitoring the behavior of individual cows is not practical or sustainable with the existing staffing and workflow.

Ranches and dairy farms often contain many similar cattle, which are difficult to distinguish visually. With the development of modern information and automation technology, automated cattle monitoring has become practical. With improvements in sensor technology and wireless network technology, researchers have adapted traditional solutions into numerous methods including: mechanical, electronic, and biometric solutions.

Unfortunately, these methods are disadvantageous in many ways. For example, ear tags cause stress on the cattle and may also be lost or damaged over time. Radiofrequency identification (RFID) technology is also commonly used to track cattle. This technology requires active RFID tags attached to the cattle, tag readers, and radio communication between them, thus requiring a high initial cost and negatively impacting the cattle's health and well-being. The purpose of our future research is to detect and analyze the estrus behavior of cattle. Although the IoT (Internet of Things)-based wearable wireless sensors such as Lora and BLE are available, it is necessary to install the device on the body of the animal. If one uses these wearable devices, installing them on the tailhead of each cow is necessary. Thus, the cattle are stressed when wearing those devices for twenty-four hours, seven days a week. In addition, the accuracy of the sensor results cannot be checked. Low-cost trackers based on the IoT devices have many adverse impacts on the livestock and farms, including network connectivity, power dependence, higher costs, and adverse health issues.

Computer vision is a more practical approach that does not require attaching any material or a sensor to the cattle being monitored [1]. Vision technology is an integral part of the Fourth Industrial Revolution. With the rapid growth of Industry 4.0, manual monitoring of individual animal behavior on commercial livestock farms is no longer sustainable for the workflow [2]. Currently, there is significant research applying deep learning and computer vision techniques to precision livestock farming and promoting the development of intelligent systems for use on livestock. An essential focus of this research is multiple-object detection and tracking for remotely monitoring numerous animals and capturing their activities.

In a computer vision-based system, the applications of deep learning and neural networks are becoming sophisticated, and their use in models for object detection has increased substantially. Such object detection systems are finding many uses in real-world applications, such as autonomous driving, robot vision, and video surveillance. Popular one-stage object-detection systems include the Regions with Convolutional Neural Network features (R-CNN) [3], You Only Look Once (YOLO) [4–11] and its variants, as well as the Single Shot Multi-Box Detector (SSD) [12]. Rather than a traditional, selective search, the Faster R-CNN is a feature extractor that uses a region proposal network (RPN) [13] to check for the occurrence of an object. As an extension of the Faster R-CNN, the Mask R-CNN [14] is an example of a two-stage deep learning network that performs region segmentation at the pixel level.

From these related studies, we can see that these non-contact monitoring methods have facilitated automation and improved the accuracy of precision dairy farming. The above mentioned deep-learning algorithms work well in detecting objects in static images. While multi-object detection refers to locating several objects belonging to a category of interest within the image, multi-object tracking can be described as tracing the movement of objects throughout a consecutive number of video frames, and consistently assigning individual object IDs.

In recent years, many research studies have sought to adapt deep learning and computer vision techniques to promote the development of smart livestock farming. Farm animals besides cattle have also featured in deep learning applications as part of the Industry 4.0 smart farming. In 2020, a study developed an automatic sheep counting system based on multi-object detection, tracking, and extrapolation techniques [15]. In 2022, a semantic segmentation and counting network was proposed to improve the segmentation accuracy and efficiency of counting pigs, an application requiring sophisticated image segmentation [16]. Another paper proposed a new solution to automatic cow detection and tracking, which optimized cow tail detection and tracking with an improved single-shot multi-box detector (SSD) and Kalman filter [17]. In 2021, a Holstein cattle open dataset was developed [18], but no dataset then existed for black cattle re-identification. This gave us an idea for improving the tracking phase in a farm management system.

We have previously developed a method of detecting and analyzing cattle mounting behavior [19]. We presented an approach to detect and track individual cattle outdoors, using inexpensive color (RGB) surveillance cameras under typical commercial farm conditions. This approach does not use individual cattle identifiers such as RFID due to their impracticality and other concerns. The multi-object tracking algorithm assigns a unique ID to each target, and this ID remains constant throughout the sequence. Before tracking, the detection algorithm obtains the object information in each frame. Therefore, tracking performance mainly relies on the detection results [20,21].

This paper proposes two contributions that are unique to our research. The first is an improved Strong-SORT [22] algorithm featuring a re-identification method for black cattle tracking, and the second is the comparison of the use of an extended Detectron-2 [23] with a customized light-weight tracking algorithm. According to the literature review, multi-object-tracking has never been performed on a black cattle dataset. We implemented a way of tracking cattle through deep learning-based multi-object tracking algorithms. Since the cattle are black and constantly moving, it is hard to rely solely on tracking them by their appearance. This too often results in an ID switch or counting mistake due to occlusion. Since YOLO v5 is a one-stage detector, we used it as our baseline detector. The main contributions of the study are summarized as follows:

- We compared the use of the cutting-edge detector YOLO v7 with the two-stage detection and instance segmentation of Detectron-2;
- We developed a customized tracking algorithm after comparing two state-of-the-art MOT algorithms, Deep-SORT and Strong-SORT;
- We used ResNet 50 as the backbone and modified the Strong-SORT algorithm by building a re-identification network structure for feature extraction in the appearance feature stage to make it more suitable for the black cattle dataset;
- We built the Re-ID dataset and presented a process for producing a semi-automatic nested dataset;
- We conducted extensive experiments on a large-scale dataset to validate the proposed approach. The results have indicated that our customized tracking algorithm enables high accuracy in tracking livestock.

The rest of the paper is composed of five sections. Section 2 describes the data used for this analysis and explains the methods applied in this article. Section 3 is the experimental implementation design, and Section 4 describes the experimental results and discussion. Finally, Section 5 concludes this article.

2. Materials and Methods

2.1. Self-Built Dataset Acquisition

The data used in this research were from the customized dataset supported by the Sumiyoshi Ranch, University of Miyazaki, Japan. These self-collected images are more complex and closer to real-life conditions; therefore, they are ideal for testing the merits of the proposed method. The surveillance camera setup monitored 10 to 20 cattle, as illustrated in Figure 1. Cattle were free to move about in this outdoor environment. The images were continuously captured by a GV-FER5700 fish-eye camera located on top of a barn, providing the best possible view. Data for Japanese black cattle were collected for 24 h a day at 25 frames per second from 25 May to 6 June 2022, for a total of 11 days of raw video. The image sequences also included the walking and lying positions of the cattle. The processing of the dataset was challenging for the following reasons: (1) the cattle changed posture frequently; (2) they were the same variety and colored similarly; (3) the lighting changed continually; (4) a complex background (including trees and ground) was difficult to distinguish from individual cattle. Images acquired in the morning were in a low light, while images at noon were over-illuminated with strong shadows. These lighting problems can cause the deep learning algorithms to inaccurately learn patches or shadows as cattle features. In some rare cases, wet soil cannot be differentiated from the cattle. Our

researchers manually annotated 3857 images in the dataset (80% and 20% for training and validation, respectively). We have used an unknown dataset for the testing dataset.

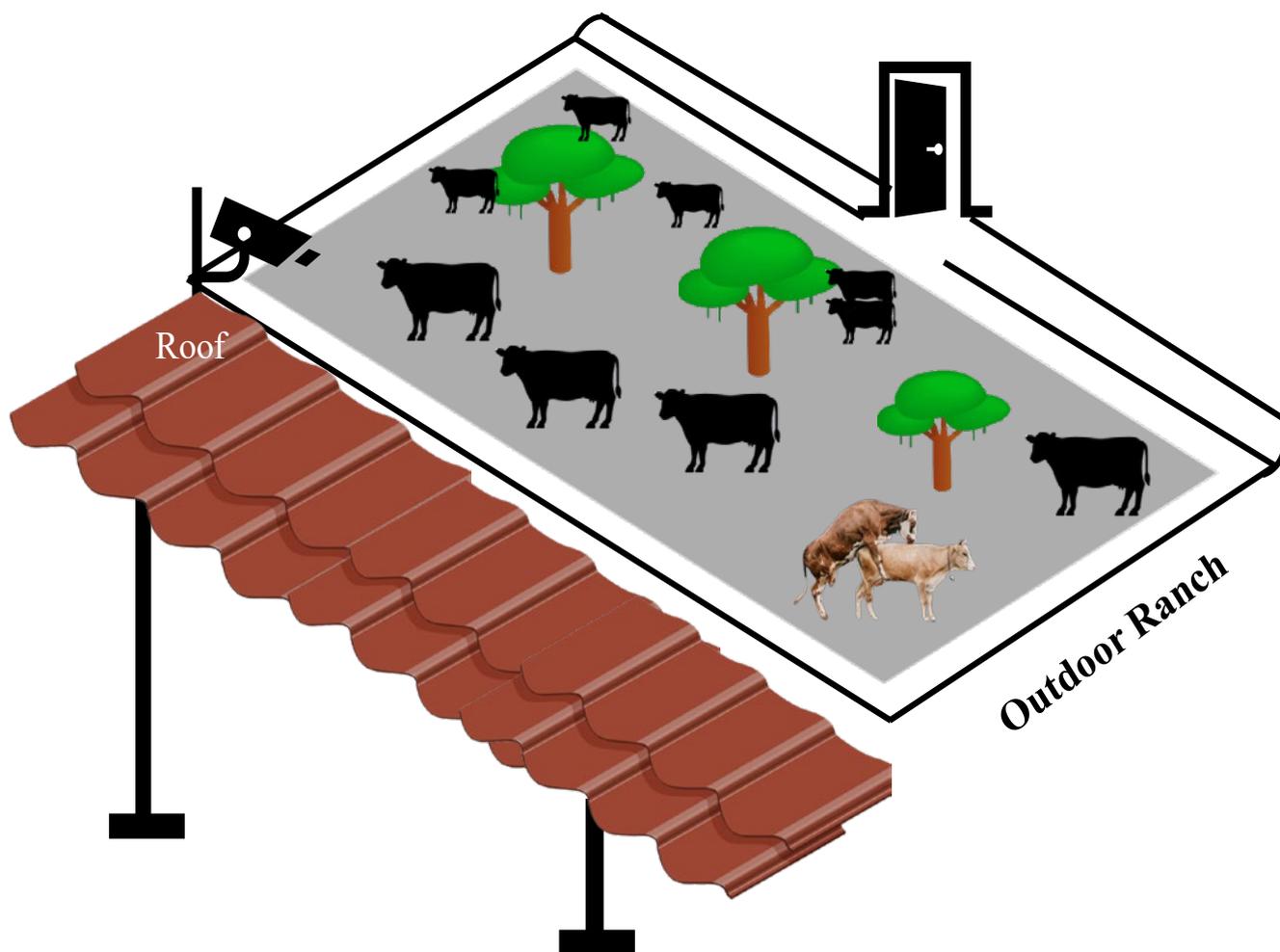


Figure 1. An illustration of the ranch layout.

The challenges have largely resulted from the fact that the data were collected in real-world conditions within a commercial farm environment. For example, there were many annotations per image in the dataset because the cattle were often densely packed into a small space. This made separating the cattle very difficult, even using manual annotations. The larger the scale and higher the quality of the data, the more successfully the model can generalize. To address this problem, data augmentation methods using geometric changes, flipping, rotation, clipping, scaling deformation, and affine operations were applied to the dataset. Also, mosaic augmentation was utilized to enhance images. These methods can increase diversity and variation in data samples to a certain extent, as explained in the following Table 1.

Table 1. The explanation of the dataset.

Dataset	Number of Videos	Number of Images	Image Enhancement Method	Image Resolution
Training set	30	3086	Mosaic augmentation	2560 × 2048
Validation set	5	771	-	2560 × 2048
Total	35	3857	-	2560 × 2048

When labeling images in the cattle dataset for training, the nested dataset production process featuring the Detectron-2 framework for automatic object detection and segmentation was used. When some objects are difficult to detect, manual annotation is inevitable. Blurred images of small cattle must be annotated manually as they are especially difficult to detect automatically. A combination of manual and automatic annotation will help reduce the human workload. The present study proposes an efficient semi-supervised, nested dataset object detection method, which aims to lower labor costs by reducing the need for human intervention.

In the experiments, over-detection is utilized to prevent gaps in tracking cows, and the over-detected regions are removed manually. Moreover, in creating a dataset in a complex environment, accurate detection of occluded or small cattle is the key to preventing the identification of cattle. For this reason, in this proposed annotation framework, after the automatic detection result from the detector was obtained, detection of the missing cattle began because the small cattle can be occluded or partly obscured behind the bigger cattle, which is the nature of the dataset.

2.2. The Proposed System

The proposed system aims at comparing the state-of-the-art deep learning tracking algorithm Strong-SORT with the customized light-weight tracking system.

The proposed cattle detection and tracking system is composed of the following five processing subsystems: (i) video data processing, (ii) cattle detection, (iii) feature extraction, (iv) re-identification (re-ID), and (v) tracking. The re-identification part is only included in the Strong-SORT tracking algorithm. In this part, we proposed the black cattle re-identification dataset as a modification of the Strong-SORT algorithm. An overview of the process components and methods are shown in Figure 2.

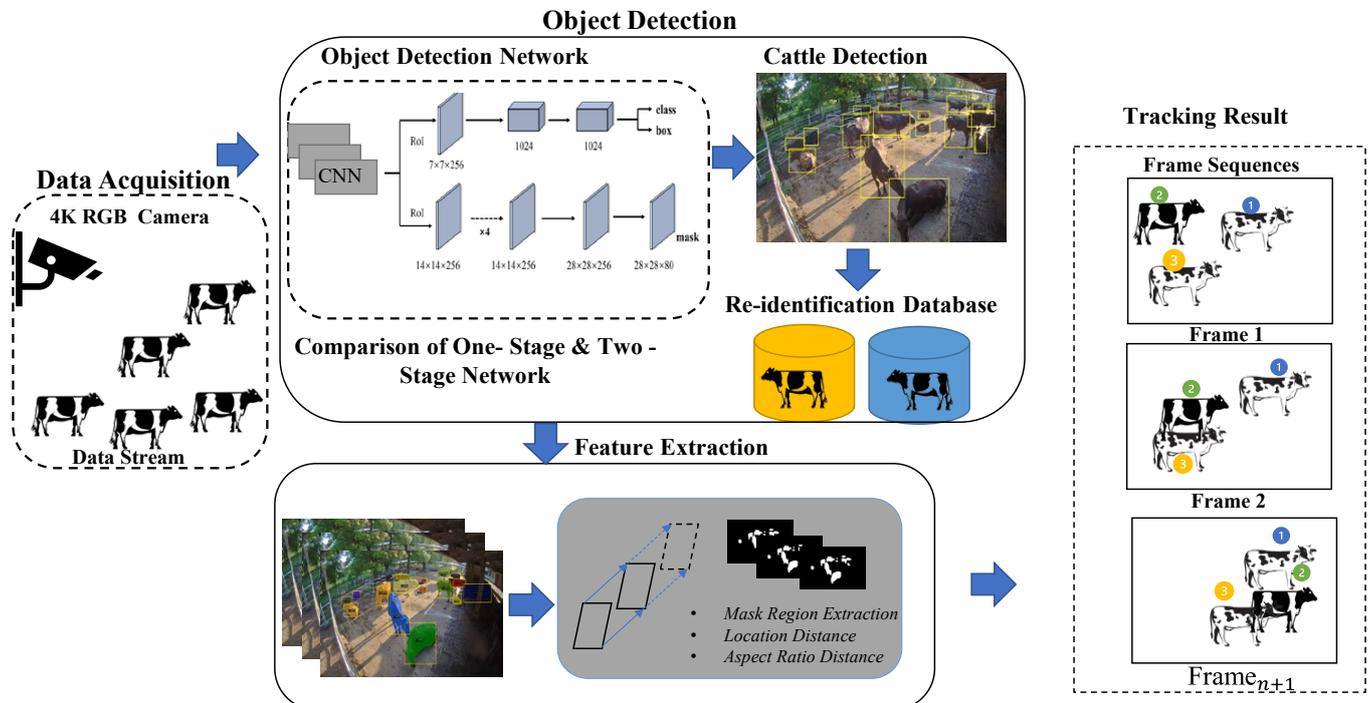


Figure 2. The complete steps of the proposed system.

2.3. The Object Detection

The object detection combines the recognition and localization of the object. It is widely used in various applications, such as intelligent traffic, pedestrian counting, animal recognition, agricultural product pest identification, and defect detection. The Convolutional Neural Network (CNN) is widely used as an important part of computer vision, and object detection technology based on deep learning has also attracted much attention. YOLO (You Only Look Once) is a detection algorithm that pursues both accuracy and speed. It treats detection as a regression problem and can detect objects at fast speeds.

YOLO algorithms have been improving in both speed and accuracy. The YOLO v1 used pre-defined candidate areas to divide the input image into a grid, and each grid can predict multiple bounding boxes. Subsequently, the YOLO v2 adopted many improvements over the YOLO v1, such as normalizing each layer of the network and using a high-resolution classifier. This classifier improved the mAP of the model, but failed to solve the problem of small object recognition. The YOLO v3 introduced a residual module to deepen the network structure. Multi-scale feature information can predict different sizes of objects and improve the accuracy of small object detection. The YOLO v4 and YOLO v5 have further improved in accuracy and speed. Compared with other versions of YOLO, the YOLO v5 is light-weight, and can quickly and accurately detect small objects. The YOLO v7 was released this year, and it includes an instance segmentation module as a branch.

Successful detection is an important precursor to tracking individual cattle. We used two detectors for this purpose: a one-stage and a two-stage detector. Firstly, we used Detectron-2 instance segmentation for subtracting the background and proceeding to the tracking stage. Considering that Detectron-2 outperforms many other detection approaches, we used its output bounding box to train YOLO.

2.4. Object Tracking

Object tracking techniques have become a fundamental part of real-time video-based applications that require object correspondence between the frames. According to the literature, recent advances in multi-object tracking (MOT) have been focusing on two different approaches: (1) a tracking by detection approach [24] and (2) a joint tracking and detection approach [25]. Tracking by detection algorithms are used to detect and classify objects before performing the object association, simplifying the process as tracking becomes a matter of associating objects over consecutive frames. Recent research has also employed the Kalman Filter (KF) algorithm [26] as a motion model, which improves the association of objects over time. SORT [27] is a technique consisting of a KF estimate of object states. Subsequently, a year later, other authors have proposed Deep-SORT [28] as an improvement that includes a novel cascading-association step using CNN-based object appearance features.

The data association algorithm combines similarities in the object appearance features with the Mahalanobis distance between object states. The Deep-SORT method achieved a promising frame rate on object tracking benchmarks [29]. Euclidean distances between extracted object appearance features are used to improve the association step. In 2022, Strong-SORT achieved a new state-of-the-art performance on the popular benchmarks like MOT 17 and MOT 20 [30].

Cattle seek shelter according to the weather, which complicates the detection process. Under these conditions, developing techniques for automatically tracking cattle could significantly lower labor costs and improve statistical performance. In this paper, we propose a new customized light-weight cattle tracking approach that can perform under actual conditions on a cattle ranch without affecting the appearance of the black cattle.

2.5. Evaluation Methods

To assess the overall performance of our framework, we separately evaluated instance segmentation and multi-object tracking. For Detectron-2 instance segmentation, the mean average precision (mAP) is defined as the mean of AP across all the categories (M), which is shown in Equation (1):

$$mAP = \sum_{i=1}^M \frac{AP_i}{M}, \quad (1)$$

For the detection stage, we computed the number of True Positives (TP), False Positives (FP), and False Negatives (FN) over all the test images, and then calculated Recall, Precision rate, and $F1$ -score, defined as in the following Equations (2)–(4):

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3)$$

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (4)$$

If the intersection over union (IoU) threshold was set to 0.5 or 50%, the mAP is called $maP_{0.5}$ or $mAP@50$. $mAP_{0.5:0.95}$, meaning that the mAP is with the $0.5 < \text{IoU} < 0.95$. The tracking performance measure we used was the multi-object tracker accuracy ($MOTA$). This is the most common metric for benchmarking MOT solutions [31], as it accounts for the three types of error that occur: the false negative (FN), false positive (FP), and identity switch (IDs), as described in Equation (5). False negatives are defined as objects that are not tracked, false positives are defined as tracked objects which should not be tracked, and identity switches are when two objects that should be tracked swap identities. Fragmentations are defined as the number of times an identity switches from “tracked” to “not tracked”:

$$MOTA = \frac{\sum_t FN_t + FP_t + IDSW_t}{\sum_t GT_t} \quad (5)$$

where t is frame index, FN is false negative, FP is false positive, $IDSW$ is ID switch, and GT is ground truth objects.

3. Experimental Design

3.1. Experimental Setup

All experiments were conducted on a Windows 10 system with an Intel® Xeon® Gold 6326 CPU @2.90 GHz 16.04, and an NVIDIA RTX a6000 with 32 GB of memory. The proposed cattle instance segmentation model was based on the YOLO version and Detectron-2 implementation used by Pytorch. A Python 3.8.11, TensorFlow 2.5.0, and Keras 2.0.8 with GPU are the requirements. The base model was pre-trained on the Microsoft Common Objects in Context dataset (MS COCO).

3.2. Choice of Detection Model

In the choice of the detection model, four models of YOLO v5, three models of YOLO v7, and Detectron-2 were used. Each one was constructed using basic backbone networks with different widths and depths. To select the base model suitable for detecting cows, the transfer learning method was employed, which uses the COCO dataset with pre-trained weights. As the balanced dataset was used, no over-fitting occurred during the training process. Images with different resolutions were introduced into models of different batch sizes for training. Figure 3 describes the training and validation loss of YOLO v5 on the cattle dataset. Figure 4 describes the results from the Detectron-2 training.

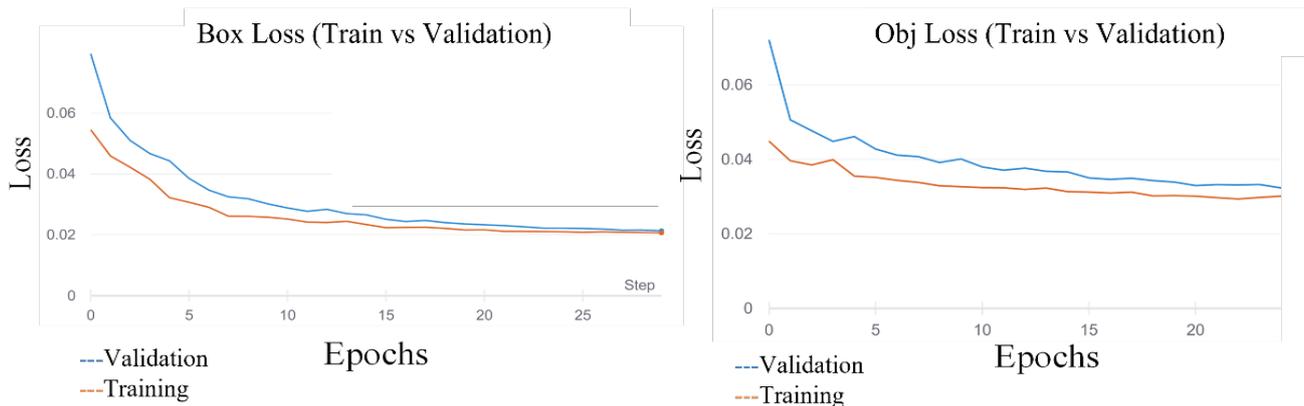


Figure 3. The training and validation loss of YOLO v5 on the cattle dataset.

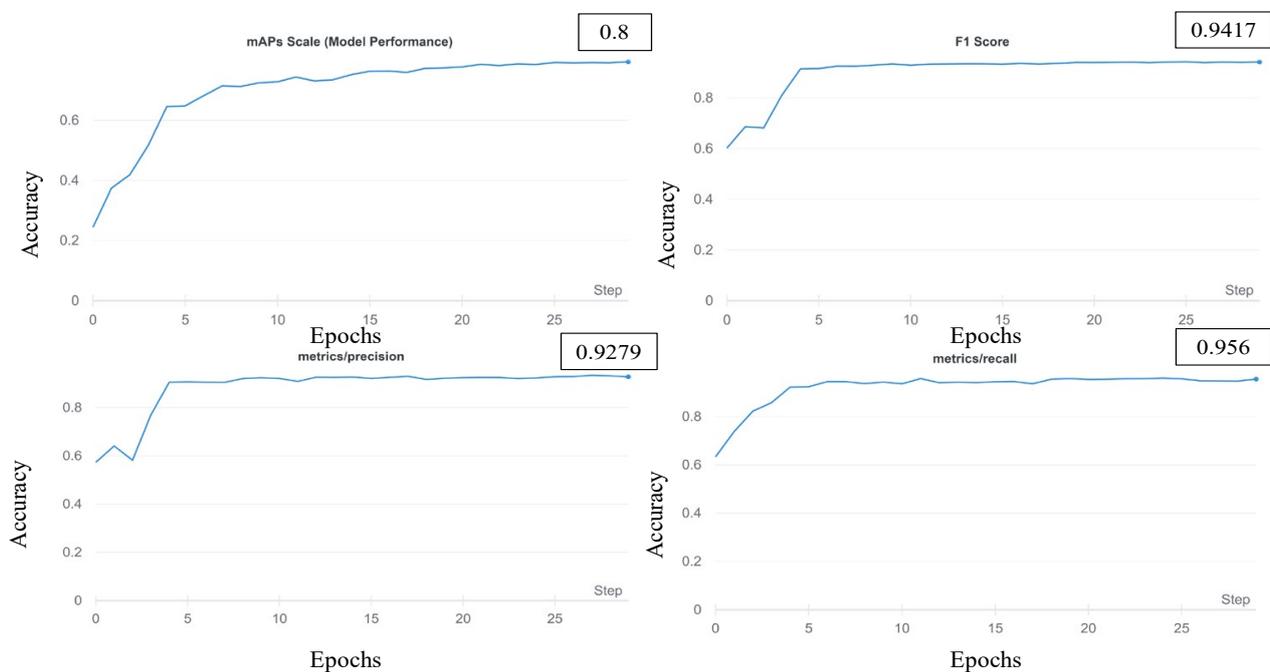


Figure 4. The results from the Detectron-2 training.

3.3. Fitting the Detection Model in the Tracking Process

In this tracking process, the systems describe two tracking algorithms: (1) Modification of the Strong-SORT-based re-identification dataset, and (2) the proposed customized light-weight tracking algorithm.

3.3.1. The MOT Network for Automatic Tracking

In this section, the modified Strong-SORT-based re-identification approach [32] is described. The cattle tracking algorithm presented in this paper has a higher tracking success rate than the original YOLO v5 with the original Strong-SORT model, and is more effective and stable in tracking cows. The cattle are detected using the YOLO v7 network from the detection model, and sent to the Strong-SORT. After that, the algorithm updates the positions of the tracked targets.

Strong-SORT is an improved version of the Deep-SORT multi-object tracking algorithms, and it includes an association method, which improves the accuracy of tracking objects for long periods of time and reduces the rate of ID switching. Strong-SORT uses an appearance branch, which is a stronger appearance feature extractor than Deep-SORT. By

taking ResNet 50 as the backbone and pre-training on the DukeMTMCreID [33] dataset, Strong-SORT can extract more discriminative features than Deep-SORT.

A Strong-SORT tracking algorithm contains six main steps. The first step is object detection. The second is pre-processing and threshold selection. The third is feature extraction, in which the Re-ID model extracts appearance features. The fourth step involves associating the data with that in the previously detected frame. The fifth is track management, which includes updating the Kalman filter, as well as initializing and deleting tracks. After the fifth step of post-processing, the tracking results are obtained. Figure 5 shows the process for cattle tracking with Strong-SORT.

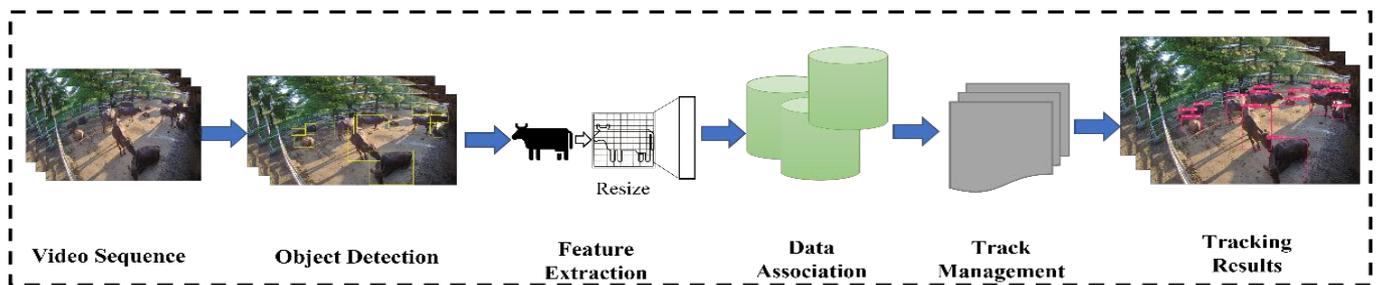


Figure 5. Cattle Tracking with Strong-SORT.

In the proposed modified Strong-SORT algorithm, only the black cattle re-identification part is improved. After cattle are detected by YOLO v5 and YOLO v7, the cattle regions are forwarded to the Strong-SORT algorithm. Following this, the cattle re-identification process is initialized, and cattle are identified. The re-identification step is explained in the next section. Finally, the bounding boxes with IDs are obtained.

3.3.2. The Cattle Re-Identification Dataset

The original Strong-SORT was trained on the Person Re-ID Dataset to extract appearance features. People are distinguished by clothes of various colors and patterns. The cattle in our dataset are uniformly black, which is the main challenge in re-training the Strong-SORT feature extractor. Therefore, for the extraction of appearance features of the cattle, the re-identification is performed. A sample of the cattle Re-ID dataset is shown in Figure 6.



Figure 6. A sample of the cattle Re-ID dataset.

Building the cattle re-identification dataset involves other challenges as well, including intra-class variations, illumination changes, low resolution, and occlusion. The person dataset contains 751 different classes. The cattle Re-ID dataset consisted of 70 cattle identities, each with an average of 240 images, totaling 16,800 images (80% for training and 20% for testing). All annotations were resized to 128×128 for processing by a Convolutional Neural Network (CNN), describe in Table 2. The proposed unsupervised re-identification network removes the need for annotated datasets. To build the cattle Re-ID dataset, separating each cow into each group is a necessary step. After the detection stage, cattle images are saved from the YOLO detector. These are automatically split using a traditional image processing-thresholding method. Based on the input images, we defined the optimal threshold parameters to split the group automatically. The best threshold values for each image are selected by analyzing all of the images. After that, the automatic splitting process is performed. Subsequently, each same image is kept in the same group as described in Figure 7. To the best of our knowledge, it is the first work to propose unsupervised Re-ID models for multi-object tracking.

Table 2. An overview of the CNN architecture used to learn the cattle re-identification.

Layer No.	Name	Patch Size/Stride	Output Size
1	Conv 1	$3 \times 3/1$	$32 \times 128 \times 128$
2	Conv 2	$3 \times 3/1$	$32 \times 128 \times 128$
3	Max Pool	$3 \times 3/2$	$32 \times 64 \times 64$
4	Residual 4	$3 \times 3/1$	$32 \times 64 \times 64$
5	Residual 5	$3 \times 3/1$	$64 \times 32 \times 32$
6	Residual 6	$3 \times 3/2$	$64 \times 32 \times 32$
7	Residual 7	$3 \times 3/1$	$128 \times 16 \times 16$
8	Dense 8	-	128
9	Batch Normalization	$3 \times 3/1$	128
10	Relu	$3 \times 3/1$	128

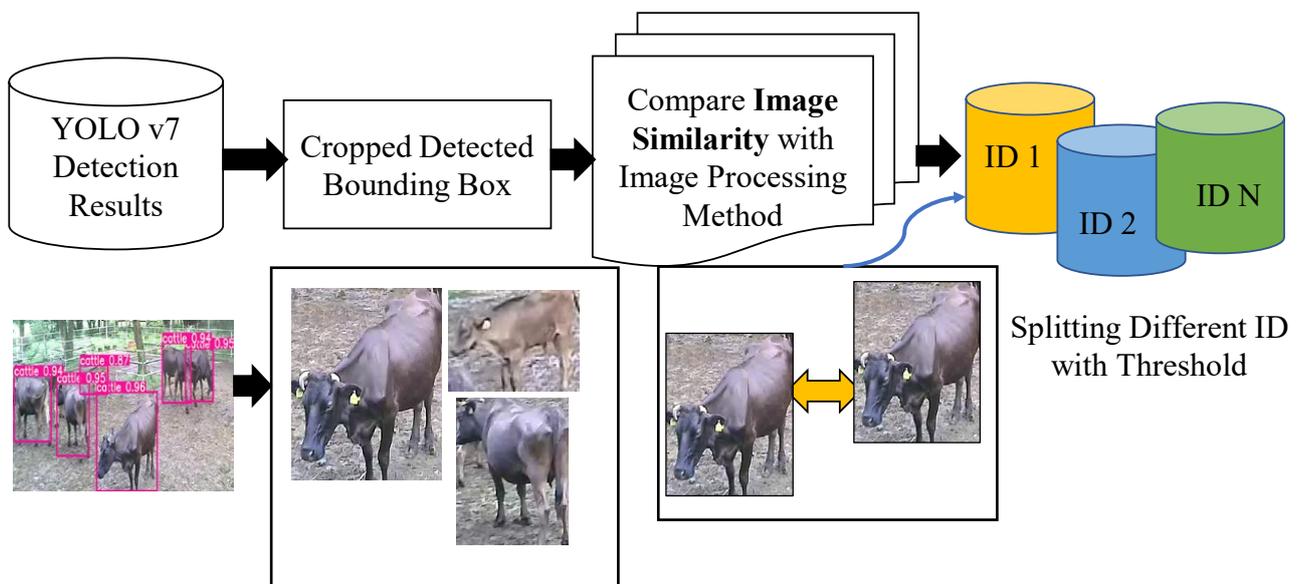


Figure 7. Building the cattle re-identification dataset with the thresholding method.

In the experiments, the Adam optimizer (Adaptive Moment Estimation) [34], which is a computationally efficient optimizer, selected a learning rate of 1×10^{-5} and a batch size of 128. While training with the CNN, the epochs stopped at 100, when the training accuracy reached 95%. When the training reached 200 epochs, the system overfitted the data. In order to facilitate training the Re-ID model and avoid overfitting the model, the

best validated model in the first 100 epochs was used for testing. We used transfer learning, which can improve the generalization ability of the model [35]. The original weight was applied and it was used to pre-train the improved re-identification network.

After completing the process of re-identification, the tracking stage of Strong-SORT is processed.

The proposed modified Strong-SORT tracking stage has the following states with the parameters shown in Table 3: (1) Tentative State: new tracks that have not been confirmed; (2) Confirmed State: tracks that have been validated as confirmed; (3) Deleted State: tracks that will be removed at the end of the iteration. This tracking stage is from the fourth and fifth steps of the Strong-SORT as described in the previous Section 3.3.1. In Table 3, N_{INIT} is the number of frames that a track remains in the initialization phase which is the same stage as the Tentative State; NN_{BUDGET} is the maximum size of the cattle store in the database. The MAX_AGE determines the maximum number of frames for which an ID will be kept alive (or active) without any valid associations, which is the same state as the Confirmed State. If an ID is not associated with any detections after the MAX_AGE number of frames, then the ID is deleted which is the same stage as the Deleted State. This helps to account for some missed detections (false negatives) by the detection model. The Strong-SORT tracking results are described in Section 4.

Table 3. Strong-SORT parameters and settings.

Parameters	Value
MAX_DIST	0.9
MAX_IOU_DISTANCE	0.8
MAX_AGE	1500
N_INIT	1
NN_BUDGET	100

3.3.3. The Proposed Customized Light-Weight Tracking Mechanism

In this section, to compare with the modified Strong-SORT tracking algorithm, we proposed a customized light-weight tracking algorithm which is composed of three primary stages, as shown in Figure 8. Detection is the first stage, adopted from Detectron-2. The second stage is matching, in which the detections in the current frame with the locations of detections in the previous frames are associated (Figure 9). The third stage is re-matching, in which new detections are re-assigned. Sections 3.3.4 and 3.3.5 describe these stages in more detail.

3.3.4. Matching

After completing the detection stage, we associate the detections in the current frame through a matching method that is based on the target's previous locations in previous frames. Once a cow is detected, the position in the next frame can be calculated according to Algorithm 1. When some cattle move quickly, the IoU may be too low between frames, therefore, affecting the matching stage. Similarly, if parts of the cattle are occluded, it affects the IoU and other components of the score.

3.3.5. Re-Matching

The proposed system matched the detection pairs according to the matching method. The system defined detections in previous frames, which have N paired detections, and the current new detections without pairs to the previous detections are called new detections. Every 30 frames, the system recalls the matching method, which is a re-matching with the previous detection and the new detection. At frame t in the re-matching stage, the new detections will re-match with all the previous detections. Once one of the new detections are successfully re-matched with a previous detection, the new detections will be assigned the previous detection IDs, which means the correct IDs are assigned. If those new detections are not re-matched with previous detection IDs, the system assigns them with new tracking

IDs. The system sets a maximum number of frames after which the IDs that are no longer in the detection process are deleted. Algorithm 1 describes the customized light-weight tracking algorithm.

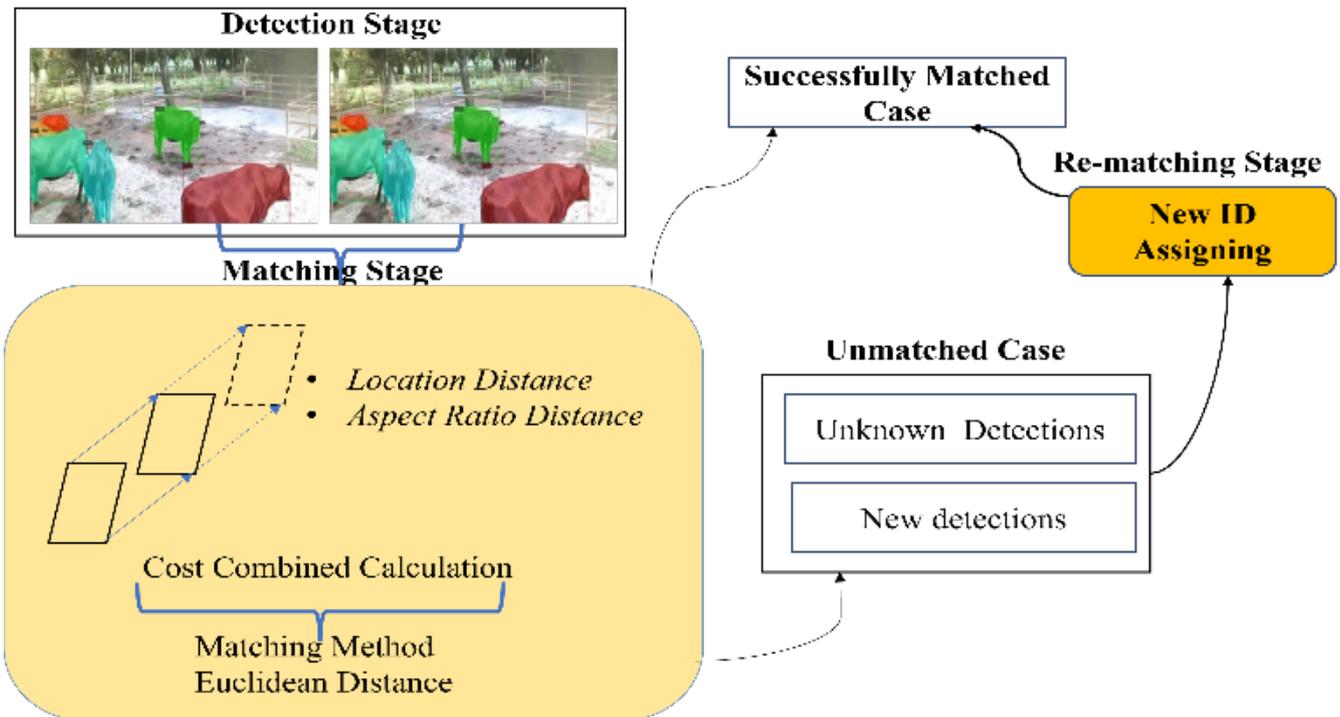


Figure 8. The proposed customized tracking algorithm.

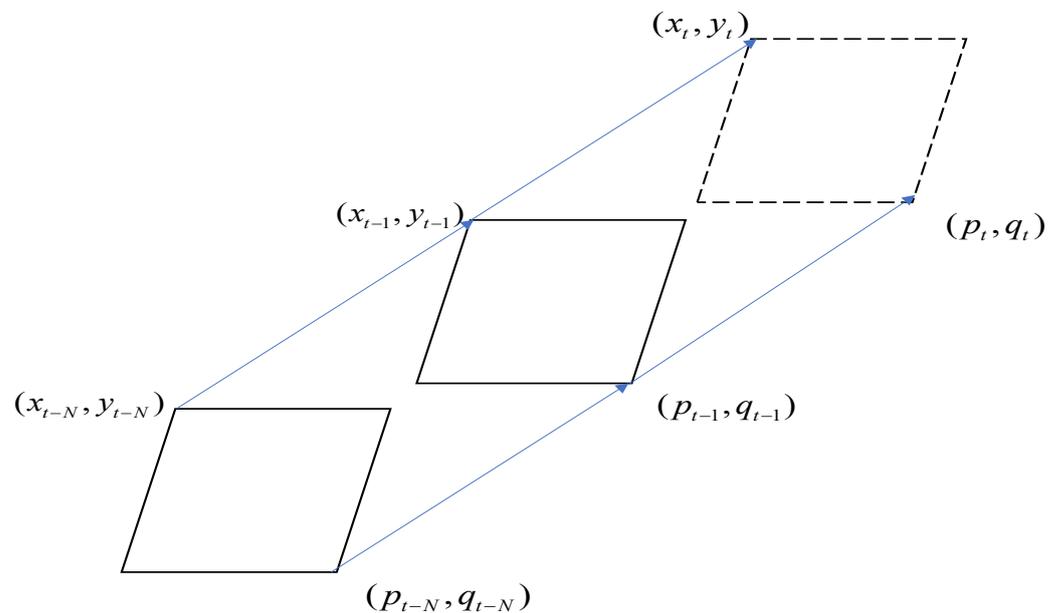


Figure 9. The explanation of the customized tracking algorithm, where (x, p) and (y, q) are the left upper point and the right upper point, respectively. The index $t-N$ represents the N frame previous detection; $t-1$ represents the previous detection; and t means the current frame location.

Algorithm 1. Cattle Tracking**Input:** Bounding Boxes and mask-regions for each cow from Detectron-2 detection**Output:** IDs, the list of cow IDs1: **Function:** Cattle_Tracking (Bounding Boxes, Mask Region):

2: Calculate Euclidean Distance for Aspect Ratio, Centroid values, and differences in Mask Area for each cow from frame to frame.

3: Perform the following steps for each cow

4. Assign each cow a tracking ID, calculating the distance between the previous and current frames using the cost function (8):

$$L_{dist} = \sum_{i=t}^N \frac{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}{N} \quad (6)$$

$$Ar_{dist} = \sum_{i=t}^N \frac{\sqrt{(Ar_i - Ar_{i-1})^2}}{N} \quad (7)$$

$$COST = L_{dist} + Ar_{dist} \quad (8)$$

where L_{dist} is the Location Distance; t is the current frame; x, y is the bounding box location; L_{dist} is the Average Euclidean Distance from current to N frames; Ar is the Aspect Ratio ($Ar = w/h$); and $COST$ is the combination of two distances (Location Distance and Aspect Ratio Distance).

5: Calculate to check the tracking database for the same detected object.

6: Compute the same object-to-tracking ID values from the tracking database.

7: Assign new tracking IDs for new cattle in the tracking database.

8: **Return IDs**9: **End Function**

The system utilized two costs for matching the ID, the location and the aspect-ratio distances; both are Euclidean distances. In this stage, the system associates detections in the current frame with the object's previously detected locations. The location distance is defined in Equation (6). When matching two detections, a match is more likely when their union is smaller, and has a larger overlapping area between their corresponding bounding boxes. A match is also more likely with a higher IoU score, as described in Equation (9):

$$IoU = \frac{Overlapping\ Area}{Union\ of\ Area} \leq 1.0 \quad (9)$$

To test the performance of the algorithm, we first experimented with 15-min video clips and then integrated them into a one-hour video to see the results described in Section 4.

4. Experimental Results and Discussion

4.1. Detection Results

In this section, the detection and tracking results of the experimental results are described. To evaluate the detection performance [36], the precision, recall, mAP of IOU = 0.5:0.95 and 0.5, and inference time are compared in each of Tables 4–8. Some of them require high computation power, while others are more efficient in the processing time. Among these models, the YOLO v5 is the best in both precision, accuracy, and efficiency in training time. In addition, various CNN baselines are tested for the Detectron-2 feature extraction process. Various ResNet [37] model sizes of the tested baselines are compared in Table 7.

Table 4. The detection results of the different YOLO v5 models.

Model	Precision %	Recall%	mAP% (IOU = 0.5:0.95)	mAP% (IOU = 0.5:)	Inference Time (s)
YOLO v5s	0.999	0.985	0.928	0.995	3500
YOLO v5m	0.999	0.985	0.936	0.994	3880
YOLO v5n	0.998	0.985	0.884	0.993	3500
YOLO v5l	0.998	0.985	0.994	0.947	4548

Table 5. The detection results with the different YOLO v7 models.

Model	Precision %	Recall%	mAP% (IOU = 0.5:0.95)	mAP% (IOU = 0.5:)	Inference Time (s)
YOLO v7	0.999	0.985	0.871	0.997	18,292
YOLO v7-X	0.999	0.985	0.891	0.997	7998
YOLO v7-W6	1.000	0.985	0.896	0.997	7815

Table 6. A comparison with the state-of-the-art one-stage detector.

Model	Precision %	Recall%	mAP% (IOU = 0.5:0.95)	mAP% (IOU = 0.5:)	Epoch	Inference Time (s)
Faster RCNN	0.835	0.907	0.953	0.987	1500	2724
YOLO v5s	0.999	0.985	0.928	0.995	100	3500
YOLO v7s	0.999	0.985	0.871	0.997	55	18,292

Table 7. A comparison of the different ResNet backbones for Detectron-2.

Methods	Depth	Iteration	Inference Time (s)	Validation Accuracy
ResNet 18	18	1000	3500	0.894
ResNet 50	50 + FPN	1000	3800	0.916
ResNet 101	Original	800	2500	0.914
		900	3200	0.921
		1000	3340	0.948
		1100	3500	0.948
		C4 3x 32 × 8d FPN 3 ×	1000	4200
		1000	4300	0.948

Table 8. The performance of the two-stage detector for comparison.

Methods	mAP @ 0.5 (%)	Inference Time (s)
Detectron-2	0.927	6300

4.2. Tracking Results

To evaluate the performance of long-term tracking, a long video of 60 min is used to test the model's performance. For a comprehensive analysis of the proposed tracking method, Table 8 shows the experimental setup and results. Table 9 provides an analysis of tracking 20 cattle using 15-min video segments. Sometimes, the ID switching issue occurred because the occlusion between cattle often resulted in false negative detections. If a lost track reappeared, it was not associated with the correct ID. This was due to the detection limitations and challenges. Tables 9–11 show the execution time (inference time), which is the overall processing time for each one-hour video for retrieving the results.

Table 9. A comparison of the Multi-Object tracking methods.

Methods	MOTA	ID-Switch	Inference Time (s)
YOLO v5 + Deep-SORT	92.52%	10	1260
YOLO v7 + Strong-SORT	95.32%	7	1800
Ours (YOLO v7 + Modified Strong-SORT)	96.88%	4	3600
Ours (Detectron-2 + Tracking)	96.88%	4	3900

Table 10. Details of the counting results for each video clip.

Videos	#Cattle Ground Truth	#Tracked Cattle	MOTA (%)
Clip 1	20	19	98
Clip 2	21	20	98
Clip 3	22	21	98
Clip 4	21	21	100
Clip 5	25	23	97

Table 11. The execution time of each module in the proposed tracking system.

Modules	Inference Time (s)
Detection module	1700
Tracking module	7200
Total	8400

Figure 10 shows the results of using one-stage and two-stage detectors. In the result, the YOLO v7 outperforms the other one-stage detector, which was tested on the same video clip. The Faster R-CNN object detector achieved especially poor results in detecting small cattle which were further away from the camera.

Figures 11 and 12 show the qualitative results of using the YOLO v7 with the modified Strong-SORT, compared with using the YOLO v5 with the Deep-SORT, which contains the largest number of cattle that were recorded in extreme lighting conditions. Each video shows the typical cattle movement in the ranch where the cattle were detected and tracked properly. Despite the inclusion of many cattle, the system tracked accurately. This shows the proposed system is robust, even with a high density of cattle moving frequently and with the overlapping images.

The knowledge of social interactions between cattle is fundamental to enhancing farming conditions and promoting animal welfare. The main goal of the proposed system was to observe interactions between cattle and identify behavior patterns associated with estrus. The system first focused on proximity contacts between at least two cattle and then constructed a trajectory map using this information to highlight the individual patterns of movement. Figure 13 provides an example of identifying behavior in a test video sequence with cattle, which can also be used for extracting a distinct contact period by restricting the frame and ID information.

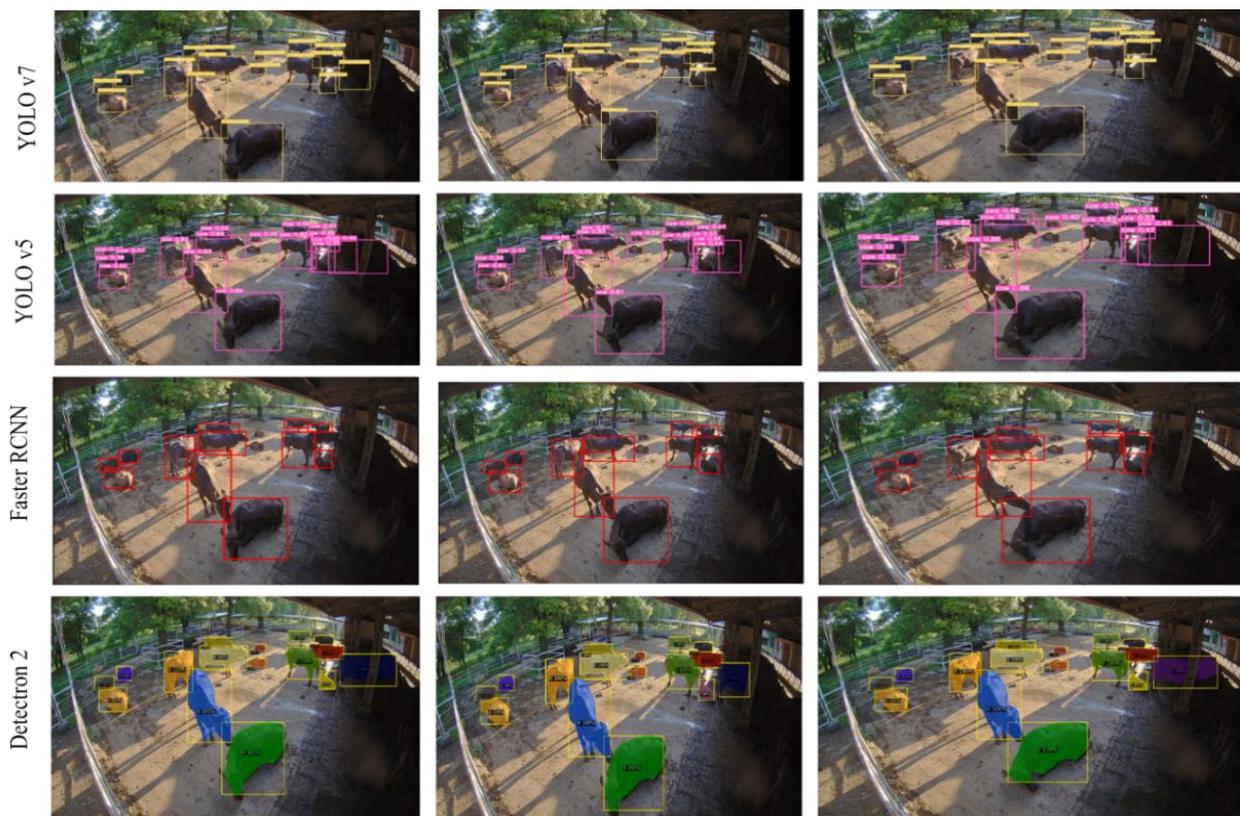


Figure 10. A comparison of detection results between the one- and two-stage detectors.

Clip 1: Number of Cattle = 15, Ground Truth=15, Time Difference Each Frame 8 sec



Figure 11. An analysis of the YOLO v7 with the modified Strong-SORT results.

Clip 1: Number of Cattle = 15, Ground Truth=15, Time Difference Each Frame 8 sec



Figure 12. An analysis of the YOLO v5 with the Deep-SORT results.

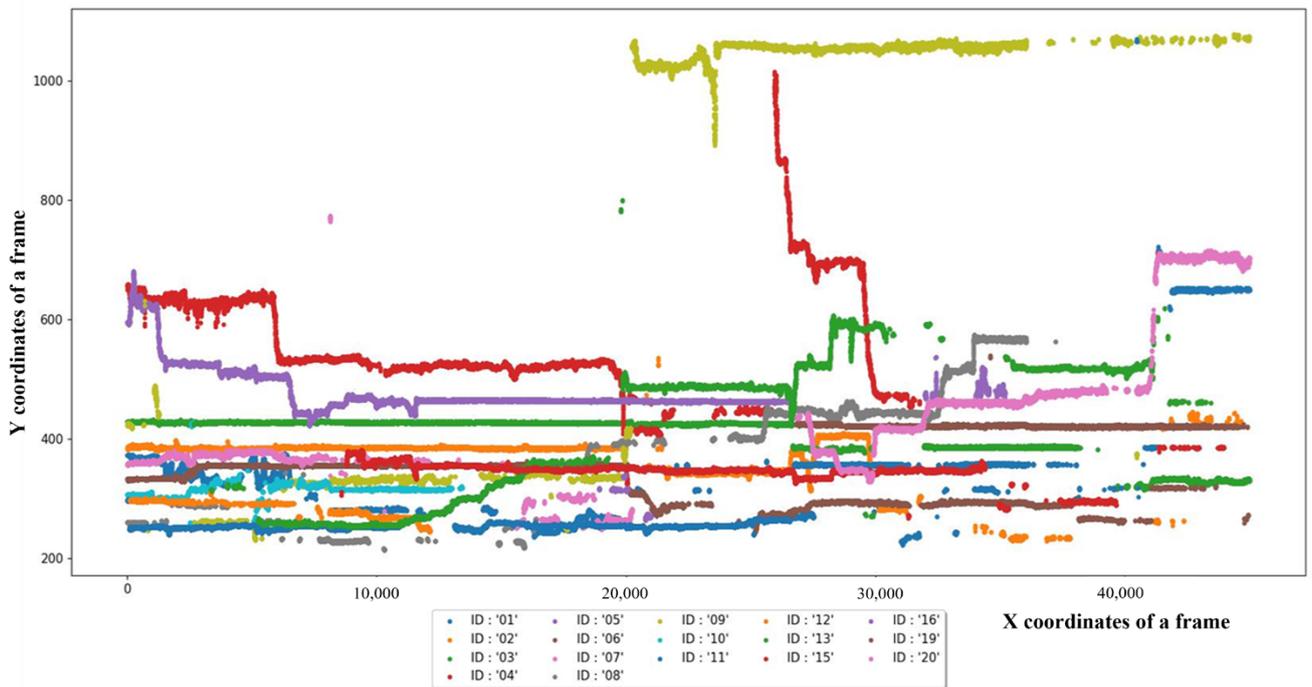


Figure 13. An analysis of tracking 20 head of cattle with the Detectron-2 with the light-weight tracking mechanism.

5. Conclusions

This study compares various combinations of detectors and tracking methods in the effort to improve both functions. For tracking cattle, the system also compared the use of a modified Strong-SORT with the proposed customized light-weight tracking system. As the main contribution of this study is to develop a complete end-to-end system that can process raw RGB images in the accurate detection and tracking of black cattle. This system has overcome the challenges of occlusion and cattle density in real-world conditions on a farm, with a single RGB camera in a fixed location.

In interpreting the findings of this study, we confirmed that the system achieved the tracking of the cattle even when tracking is interrupted by occlusion or changes in illumination. Long-term tracking can provide continuous and automatic monitoring of the health and welfare status of the animals. Moreover, this tracking methodology can be used for other species besides cattle, and for detecting behavior such as lameness and mounting. With improved detection of behavior in estrus, the system can pinpoint the optimal time for artificial insemination. One challenge of this system is the computing time required for the real-time processing of the video data. The comparison between the one and two-stage detection algorithms showed that the one-stage model provided a higher inference speed.

The proposed system compares favorably with other cattle detection and identification methods. The accuracy achieved in [38] was only 85.4%. This study used a dataset with 5042 images of full cattle bodies and featured the Mask R-CNN detection method using VGG 16 as the baseline backbone. In contrast [39], achieved 73% accuracy using the YOLO v3 on 11,754 frames and Dark Net as the backbone. In [40], an accuracy of 94% was achieved using the deep learning Mask R-CNN on 750 images. In [41], a Faster R-CNN, this one-stage detection network achieved an accuracy of 89.1%; however, only 43 images were used. An accuracy of 89.95% was achieved in [42], which used an SDAE detection method on 1000 images with CNN as the baseline network. In paper [43], the authors used CNN and LSTM-based identification methods, achieving an accuracy of 88%. In paper [44], the authors used Temporal Segment Networks (TSN) identification methods and achieved an accuracy of 84.4%. It is evident that the proposed method is a significant improvement over other dairy cattle detection algorithms, and the feasibility of detecting and tracking black cattle is validated at a significant level.

Compared to the aforementioned studies, the proposed system in the present study used a black cattle dataset. The aim of constructing this dataset is to meet the formidable challenge of detecting and tracking black cattle without the additional infrastructure. However, despite the improved performance of the method, the system could still not fully overcome the ID switching issue, which is one of the most frequently stated problems in MOT. In this study we found that the ID switching issue mainly resulted from occlusion, and from when the cattle left the camera's field of view. Through the combination with other systems, we are confident that our work will soon enable the more intelligent monitoring of livestock.

We developed a system based on a comparison of various combinations of deep learning detection and tracking technologies applied to a black cattle dataset. The ability to track individual cattle with deep learning is the key to creating a full system that can provide all the information required for farm management, including the early detection of the optimal time for artificial insemination. The experiment results show that the proposed method allows the tracking of cattle for up to 1 h, with a MOTA of 96.88% without the use of additional hardware and using only a standard RGB camera. The tracks derived from this system can be used to calculate the behavioral metrics for future work. The proposed system also introduced different deep learning-based trackers. In summary, we have proven that our proposed customized light-weight tracking algorithm performs the best tracking accuracy and is better than the modification of Strong-SORT in our black cattle dataset.

In future research we plan to make our tracker more stable by improving the optical flow and addressing the variability in light intensity. A more significant challenge lies in effectively implementing deep learning methods. Future work will involve testing the strategy on more complex trackers and extending our work to apply in real-time cattle monitoring.

Author Contributions: Conceptualization, S.M.N., T.T.Z., P.T. and I.K.; methodology, S.M.N. and T.T.Z.; software, S.M.N.; validation, T.T.Z.; formal analysis, S.M.N.; investigation, T.T.Z.; resources, I.K., S.M.N., T.T.Z. and P.T.; writing—original draft preparation, S.M.N.; visualization, S.M.N.; supervision, T.T.Z.; S.M.N. conducted a series of experiments, and all four authors analyzed the results. Real-life interpretation: I.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by JST SPRING, Grant Number WN922001.

Institutional Review Board Statement: Ethical review and approval were waived for this study, due to no enforced nor uncomfortable restriction to the animals during the study period. The image data process used for analysis in this study were collected by an installed camera without disturbing natural parturient behavior of animals and routine management of the farm.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: We thank K. Sumi, S. Inoue, Y. Kawagoe, and M. Chikunami for setting up the camera on the Sumiyoshi ranch and collecting the data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Guo, Y.; He, D.; Chai, L. A machine vision-based method for monitoring scene-interactive behaviors of dairy calf. *Animals* **2020**, *10*, 190. [[CrossRef](#)] [[PubMed](#)]
2. Morrone, S.; Dimauro, C.; Gambella, F.; Cappai, M.G. Industry 4.0 and Precision Livestock Farming (PLF): An up-to-Date Overview across Animal Productions. *Sensors* **2022**, *22*, 4319. [[CrossRef](#)]
3. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2015; pp. 580–587.
4. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
5. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
6. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
7. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
8. Couturier, R.; Noura, H.N.; Salman, O.; Sider, A. A deep learning object detection method for an efficient cluster's initialization. *arXiv* **2021**, arXiv:2104.13634.
9. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
10. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
11. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
12. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot Multi Box Detector. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 21–37.
13. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems*, 28; MIT Press: Cambridge, MA, USA, 2015.
14. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–27 October 2017; pp. 2961–2969.
15. Zhang, Y.; Yu, C.; Liu, H.; Chen, X.; Lei, Y.; Pang, T.; Zhang, J. An Integrated Goat Head Detection and Automatic Counting Method Based on Deep Learning. *Animals* **2022**, *12*, 1810. [[CrossRef](#)]

16. Kim, J.; Suh, Y.; Lee, J.; Chae, H.; Ahn, H.; Chung, Y.; Park, D. EmbeddedPigCount: Pig Counting with Video Object Detection and Tracking on an Embedded Board. *Sensors* **2022**, *22*, 2689. [[CrossRef](#)]
17. 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* **2022**, *12*, 1339. [[CrossRef](#)]
18. Guo, Q.; Sun, Y.; Min, L.; van Putten, A.; Knol, E.F.; Visser, B.; Rodenburg, T.; Bolhuis, L.; Bijma, P. Video-based Detection and Tracking with Improved Re-identification Association for Pigs and Laying Hens in Farms. In Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, SciTePress, Online-Streaming, 6–8 February 2022.
19. 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* **2022**, *18*, 211–220.
20. Noe, S.M.; Zin, T.T.; Tin, P.; Hama, H. Detection of Estrus in Cattle by using Image Technology and Machine Learning Methods. In Proceedings of the 2020 IEEE 9th Global Conference on Consumer Electronics (GCCE), Kobe, Japan, 13–16 October 2020; pp. 320–321.
21. Noe, S.M.; Zin, T.T.; Tin, P.; Kobayashi, I. Automatic Detection of Mounting Behavior in Cattle using Semantic Segmentation and Classification. In Proceedings of the 2021 IEEE 3rd Global Conference on Life Sciences and Technologies (Life-Tech), Nara, Japan, 9–11 March 2021; pp. 227–228.
22. Du, Y.; Song, Y.; Yang, B.; Zhao, Y. Strong-SORT: Make Deep-SORT great again. *arXiv* **2022**, arXiv:2202.13514.
23. Abhishek, A.V.S.; Kotni, S. Detectron2 Object Detection & Manipulating Images using Cartoonization. *Int. J. Eng. Res. Technol. (IJERT)* **2021**, *10*. [[CrossRef](#)]
24. Mekonnen, A.A.; Lerasle, F. Comparative evaluations of selected tracking-by-detection approaches. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 996–1010. [[CrossRef](#)]
25. Milan, A.; Leal-Taixé, L.; Schindler, K.; Reid, I. Joint Tracking and Segmentation of Multiple Targets. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5397–5406.
26. Montella, C. The Kalman filter and related algorithms: A literature review. *Res. Gate* **2011**. [[CrossRef](#)]
27. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and Realtime tracking. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3464–3468.
28. Wojke, N.; Bewley, A.; Paulus, D. Simple Online and Realtime Tracking with a Deep Association Metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3645–3649.
29. Li, Z.; Tian, X.; Liu, Y.; Shi, X. Vehicle Tracking Method Based on Attention-YOLOv5 and Optimized DeepSORT Models. In Proceedings of the 2022 IEEE 11th Data Driven Control and Learning Systems Conference (DDCLS), Chengdu, China, 3–5 August 2022; pp. 114–121.
30. Dendorfer, P.; Osep, A.; Milan, A.; Schindler, K.; Cremers, D.; Reid, I.; Roth, S.; Leal-Taixé, L. MOT challenge: A benchmark for single-camera multiple target tracking. *Int. J. Comput. Vis.* **2021**, *129*, 845–881. [[CrossRef](#)]
31. Bernardin, K.; Elbs, A.; Stiefelhagen, R. Multiple Objects Tracking Performance Metrics and Evaluation in a Smart Room Environment. In Proceedings of the Sixth IEEE International Workshop on Visual Surveillance, in Conjunction with ECCV, Graz, Austria, 13 May 2006; Volume 90.
32. Ye, M.; Shen, J.; Lin, G.; Xiang, T.; Shao, L.; Hoi, S.C. Deep Learning for Person Re-Identification: A Survey and Outlook. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 2872–2893. [[CrossRef](#)]
33. Gou, M.; Karanam, S.; Liu, W.; Camps, O.; Radke, R.J. DukeMTMC4ReID: A Large-Scale Multi-Camera Person Re-Identification Dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 10–19.
34. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
35. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A comprehensive survey on transfer learning. *Proc. IEEE* **2020**, *109*, 43–76. [[CrossRef](#)]
36. Henderson, P.; Ferrari, V. End-to-End Training of Object Class Detectors for Mean Average Precision. In *Asian Conference on Computer Vision*; Springer: Cham, Switzerland, 2017; pp. 198–213.
37. Wu, Z.; Shen, C.; Van Den Hengel, A. Wider or deeper: Revisiting the ResNet model for visual recognition. *Pattern Recognit.* **2019**, *90*, 119–133. [[CrossRef](#)]
38. Chen, S.; Wang, S.; Zuo, X.; Yang, R. Angus Cattle Recognition using Deep Learning. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 4169–4175.
39. Tassinari, P.; Bovo, M.; Benni, S.; Franzoni, S.; Poggi, M.; Mammi, L.M.E.; Mattoccia, S.; Di Stefano, L.; Bonora, F.; Barbaresi, A.; et al. A Computer Vision Approach Based on Deep Learning for the Detection of Dairy Cows in Free Stall Barn. *Comput. Electron. Agric.* **2021**, *182*, 106030. [[CrossRef](#)]
40. Xu, B.; Wang, W.; Falzon, G.; Kwan, P.; Guo, L.; Chen, G.; Tait, A.; Schneider, D. Automated cattle counting using Mask R-CNN in quadcopter vision system. *Comput. Electron. Agric.* **2020**, *171*, 105300. [[CrossRef](#)]
41. Han, L.; Tao, P.; Martin, R.R. Livestock detection in aerial images using a fully convolutional network. *Comput. Vis. Media* **2019**, *5*, 221–228. [[CrossRef](#)]
42. Lin, M.; Chen, C.; Lai, C. Object Detection Algorithm Based AdaBoost Residual Correction Fast R-CNN on Networks. In Proceedings of the 2019 3rd International Conference on Deep Learning Technologies, Xiamen, China, 5–7 July 2019; pp. 42–46.

43. Qiao, Y.; Su, D.; Kong, H.; Sukkarieh, S.; Lomax, S.; Clark, C. Individual cattle identification using a deep learning-based framework. *IFAC-Pap. Online* **2019**, *52*, 318–323. [[CrossRef](#)]
44. Nguyen, C.; Wang, D.; Von Richter, K.; Valencia, P.; Alvarenga, F.A.; Bishop-Hurley, G. Video-based cattle identification and action recognition. *arXiv* **2021**, arXiv:2110.07103.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.