Smart Agricultural Technology 12 (2025) 101561

Contents lists available at ScienceDirect

Smart Agricultural Technology

journal homepage: www.journals.elsevier.com/smart-agricultural-technology

AutoCattlogger: Track-assisted, automatic, instant cattle identity cataloger for recognition and beyond

Manu Ramesh *, Amy R. Reibman

Elmore School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA

A R T I C L E I N F O

Keywords:
Cattle recognition
Cattle tracking
Cattle identification
Video analytics

A B S T R A C T

We create an automatic Holstein cattle cataloging system called the AutoCattlogger to address the dynamic needs of dairy farms with frequently changing herds. This system instantly registers the identities of cattle by tracking them passing underneath a camera. The AutoCattlogger saves the identities of each individual cow in a predefined, interpretable representation space as barcodes, which allows for direct addition or deletion of cow identities without expensive model retraining. Thus, this system only requires a farmer to make a new cow walk under a camera for its identity to be instantly learned. Additionally, it is also capable of one-shot learning from single images of cows when videos are not available, has built-in illumination calibration functionality, and is robust to varying cow poses. The AutoCattlogger supports AutoCattleID, which is a cattle identification functionality that yields consistently high Track-level Top-1 recognition accuracy of above 90% even after a year on unannotated, unsegmented dairy videos. This demonstrates the robustness of our system to long-term changes in cow sizes and scene lighting conditions, which is a hard problem for Deep Metric Learning based recognition methods. The AutoCattlogger, with its tracks and identities, can thus serve as the foundation for many cattle video analytics tasks such as video retrieval, weight estimation, and primarily, cattle recognition.

1. Introduction

Computer vision systems can improve the efficiency of cattle herd management in dairy farms. Dairy farms in the Americas usually have hundreds of cows, and require significant labor resources to manage the herd. The dairy workers must regularly monitor and record several attributes for individual cows. These attributes include symptoms of health issues such as lameness, abnormalities such as reduced feed intake, weight and body condition, and estrus behavior. Automating these monitoring tasks will reduce labor and operational expenses, while potentially increasing operational reliability. Using computer vision methods for automation further ensures ease of use and lower cost of equipment.

The task of cattle identification underpins all other automation systems mentioned above. The weight, body condition, behavior or any other recorded characteristic, must always be associated with an individual cow. Designing a robust and reliable computer vision cattle identification system is thus an important first step towards developing herd management systems.

A good computer vision cattle identification system satisfies accuracy requirements even within the limitations of a real-world setting. Unlike curated datasets, real-world barns come with limited data, and limited time and workforce available to collect and annotate this data. Moreover, dairy farms regularly take in new cows, and have older ones removed. A good recognition system must be able to quickly adapt to the changing herd without the need for expensive computation.

However, almost all recent computer-vision cattle identification methods use either Deep Learning (DL) or Deep Metric Learning (DML) approaches, which cannot operate within the aforementioned limitations of a real-world setting. The DL methods approach cattle identification as a multi-class classification problem and force neural network systems to memorize the appearances of hundreds of cows. These systems cannot learn to identify new individuals without modification to their neural network architecture, and expensive model retraining. Identification systems that use DML, train neural networks to generate feature embeddings for cow instances using variants of contrastive loss functions. These systems can learn the identities of new individuals without modifications to the neural network architecture and without requiring any retraining. However, similar to DL systems, they too require large amounts of annotated training data to achieve the required levels of performance. Satisfying this requirement is difficult in a real-world setting.

* Corresponding author.
E-mail address: reibman@purdue.edu (A.R. Reibman).

https://doi.org/10.1016/j.atech.2025.101561
Received 4 September 2025; Received in revised form 10 October 2025; Accepted 20 October 2025

Available online 24 October 2025

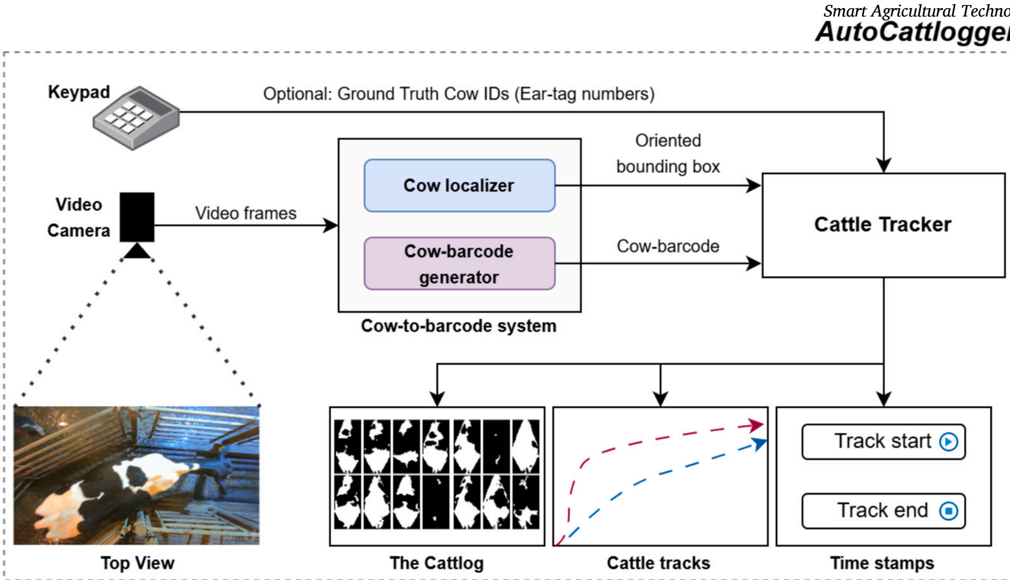

Fig. 1. Block diagram of the AutoCattlogger. In this diagram, the boxes indicate the operators, the illustrations indicate the form of data, and the arrows indicate data flow. The AutoCattlogger tracks each cow passing underneath a top-view camera and instantly generates its identifying feature vector and saves it in the 'Cattlog'. The track information along with its start and end times for each cow is also returned.

On the contrary, our system, the AutoCattlogger, is designed to work within the limitations of the real-world while also achieving commendable cattle identification performance. The AutoCattlogger is built by combining an ID-agnostic cattle tracker inspired from [1], with a Cow-to-barcode conversion system developed from the Eidetic Cattle Recognition system [2]. This unique combination results in a system that not only automatically labels each cow instance, but also instantly registers their identities for recognition. This means that as soon as a new cow walks past our camera, our system is ready to recognize the cow. Further, the AutoCattlogger is also capable of learning identity representations from single training images per cow. This one-shot learning capability is useful when farmers need to quickly add a cow to the catalog without having to make it walk under a camera. Thus, the AutoCattlogger builds a cattle identity catalog by seamlessly blending the three steps of data collection, annotation and training, requiring virtually zero human effort in the process.

The block diagram in Fig. 1 shows the inputs and outputs of the AutoCattlogger along with a high level overview of the components involved. The AutoCattlogger ingests raw videos, which are unsegmented and unannotated videos of cows from a top-view camera. In our data, these videos have cows walking in a single file after being milked. Optionally, a list of the ground-truth identities (ear-tag numbers) of these cows in the order in which they appear can also be provided. The AutoCattlogger has three main outputs – the Cattlog of cow-barcodes serving as the identity model of all the seen cows, the path information of each cow as tracks, and the start and end frame numbers for each track.

Since cattle recognition and tracking provide a foundation for the different herd management tasks mentioned earlier, many video analytics applications can be built on top of the AutoCattlogger. These are shown in Fig. 2. Of these, this paper addresses Cattle Recognition, with the Automatic Cattle Identifier, also called AutoCattleID, which is an application that utilizes video context information embedded in the cattle tracks. Cattle weight estimation from side-view videos [3] is another application. The start and end points of each cattle-track can be used to obtain video clips of individual cows from synchronized side-view cameras observing the same scene. The side-view weight estimator in [3] can then be applied to the obtained video clips to automatically log the weight estimates of each cow in the herd. A basic version of the AutoCattlogger was also used to retrieve video clips of individual cattle in [4].

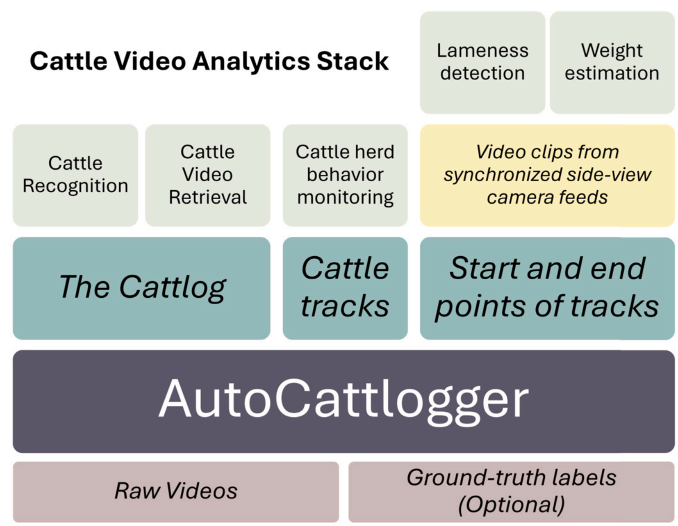

Fig. 2. Cattle video analytics stack built atop the AutoCattlogger. Blocks with italicized text represent data, be it raw or processed. Blocks with normal text represent the application tools built primarily on top of the underlying data.

AutoCattleID utilizes the foundational principle of the ECR system [2], which is to use stochastic, learning-based models to replicate tasks which humans are intuitively good at, and to use deterministic algorithms that are not learning-based to perform tasks which computers are historically good at. Accordingly, AutoCattleID uses a learning-based model to identify parts of a cow's anatomy via keypoints, and uses deterministic algorithms to memorize the appearances of hundreds of cows and find the best match. Further, the AutoCattlogger is designed with checks and balances that prevent it from blindly accepting every output of the learning-based detectors. Rules based on a shape model are used to verify the validity of the detected keypoints before they are allowed to be used for cattle pose correction. Further, the cow-barcodes are interpretable intermediate representations that enable designers to directly compare them with pose-corrected cow instances and pinpoint representational errors. This explainability of system input-output relationship allows us to accurately identify sources of errors and design parts such as the aforementioned color-corrector to address them.

The AutoCattlogger builds on top of our previous efforts in the domain of cattle recognition which introduced a barcode based cattle recognition system [2], a keypoint rectifier [5], and an illumination calibration system [6]. Our key contributions in this paper are:

- We combine the cow-to-barcode system from our previous works [2,5,6] which translates unique coat patterns into binary representations, with an object tracker to create the AutoCattlogger. The AutoCattlogger is thus a modular, track-based, automatic and an instant cattle identity registration system. This system, presented in Sec. 3, reduces human effort to just making cattle walk below a camera and noting down their ear-tag numbers.
- The AutoCattlogger, despite being designed to learn cattle identity representations from videos, is also capable of one-shot learning from images of cows when videos are not available. This is demonstrated in Sec. 6.1.
- We present AutoCattleID, an improved cattle recognition application built atop the AutoCattlogger, in Sec. 3.4. AutoCattleID uses contextual information from the cattle tracks for identifying individual cows, unlike other candidate systems that totally neglect this crucial information.
- We demonstrate that AutoCattleID is robust to long-term changes in cow appearance even after a year (See Sec. 6.3.). This makes it more accurate than Deep Metric Learning-based methods as shown in Sec. 6.1.

We study the related works next in Sec. 2.

2. Related works

Automatic recognition and tracking of cattle provide avenues to many cattle analytics applications including their health tracking, behavior monitoring, milk production at an individual level. Moreover, automation of these tasks ensures savings in cost, time and labor. The primary functions supported by the AutoCattlogger are cattle tracking and identification. Since the AutoCattlogger uses a very simple online visual Multi Object (cattle) Tracker (MOT) inspired by [1], in this section, we discuss the related works only in the domain of cattle recognition.

Cattle recognition techniques have come a long way since the traditional methods of branding, ear-notching and ear-tagging. Newer methods such as ones that use RFID tags [7] are also intrusive, expensive, and easily susceptible to wear and tear in the harsh barn environments. Methods that recognize cattle using muzzle-prints [8–11] or retinal patterns [12] are just too slow to sample instances. Also, they cannot be automated. So, they require expensive human labor for implementation.

Lately, many computer-vision based cattle recognition algorithms have emerged which identify cattle from their faces [13–18], ear-tags [19–22], coat patterns in the side-view [23] or top-view [24,2], or keypoints on their backs [25,26]. Because they are executed on computers, they are automation friendly. However, some have operational limitations. Identifying cattle from their faces would mean that the algorithm must wait till the face of the required animal is facing the camera. These faces could also easily be occluded by other cattle, fences or other objects in between them and the camera. The same problems also affect computer-vision methods that identify cattle by reading their ear-tag numbers. Using the side view presents challenges such as occluding fences and other animals, which need extra processing to remove. Using top-view for identification has no such limitations. The backs of cattle are always visible without obstruction to a top-view camera.

Of the top-view based identification systems in the literature, many [27,16,28,29] use monolithic neural networks that perform end-to-end cattle identification or provide the embeddings for identity classification. These monolithic systems are burdened to localize cows within their (already) detected bounding-boxes, and then identifying them without errors. Unlike them, our AutoCattlogger is a non-monolithic, modular system built from a small set of components that perform specific tasks. This modular design allows us to upgrade specific components such as the instance mask or keypoint detectors if the need arises.

In a conventional barn, the composition of the cattle herd changes regularly as new cows are added and older ones are removed. A useful cattle recognition system must be able to overcome this 'herd adaptability' problem. Many methods including the more recent ones still use neural-networks [27,14,16,28,29] or other models such as SVMs [23] to predict cow identities as individual classes. These pretrained classifiers cannot adapt to changing herds without expensive, time consuming and data hungry model retraining. Retraining on new herds also demands intense data annotation efforts. Moreover, neural-network classifiers are constrained to identify only a fixed number of cow individuals if the model architecture is frozen.

Systems that were developed to solve this herd adaptability problem include those that use embeddings from Deep Metric Learning [24], [30], and those that use a predefined embedding space [2]. The idea behind the DML systems in [30], [24] and all its derivatives [1,31,32] is to learn an embedding model that clusters images from the same individual together while pushing images from different individuals far apart in its embedding space, using a suitable loss function. Embeddings for previously unseen cattle are then generated using the same trained model in the same embedding space. A cow-instance that needs to be identified is first transformed into a vector in this embedding space using the trained model. This embedding vector is then used to predict the identity of the cow-instance using the embeddings from a support-set of training instances and algorithms such as K-nearest neighbors.

However, a simpler approach to solve the herd adaptability problem is using a predefined embedding space for identity representation as done by the systems in [2] and its derivatives [5,6,4] and our AutoCattleID. The idea is to first correct all cow instances for pose and lighting, and then transform them into a barcode of predefined dimensions. This makes learning or unlearning cattle identities as easy as adding or deleting these barcodes. A cow instance to be identified is first converted into a barcode, which is then matched to its nearest barcode from the set of training instances to get the identity prediction.

For a theoretical understanding of why using predefined cow-barcode features for cattle identification is better than using features from DL or DML, we must look at the problem from the reverse perspective. Training Multi Layered Perceptrons, Convolutional Neural Networks, or transformers to read everyday barcodes or QR codes, implies having them memorize an impossibly large number of patterns. Using DML for the task implies training the embedding-generator models to sort all possible barcodes into separable bins. Both these approaches make the problem intractable.

We postulate that once all non-uniformities are eliminated from an image of a cow, its appearance can be reduced to a representation like an everyday barcode. Using neural-networks to detect a fixed number of keypoints that have similar appearance across all cows is definitely a more tractable problem. These keypoints can then be used to eliminate non-uniformities in cow size and pose as done by our AutoCattlogger. Experimental results proving that our AutoCattleID is better than the DML method [24] is provided in Sec. 6.1.

In addition, DML methods are not directly interpretable. The authors of [30] present example images where their identification system errs. While most of them can be attributed to their system lacking a pose corrector, they do show a few instances where their system decides cows with large differences in coat-patterns to be the same. The authors of [32] also present example images where their system gets confused between two different cows that look clearly different. They even have examples where cows with large white regions are predicted as a cow that is almost completely black, and vice versa. There is no deterministic explanation available for this behavior, and one can only speculate. The cow to barcode relationship established by the AutoCattlogger ensures the desired interpretability of representations.

Our Eidetic Cattle Recognition system [2] was the first to involve keypoints in identifying cattle. Later methods such as [25] that iden-

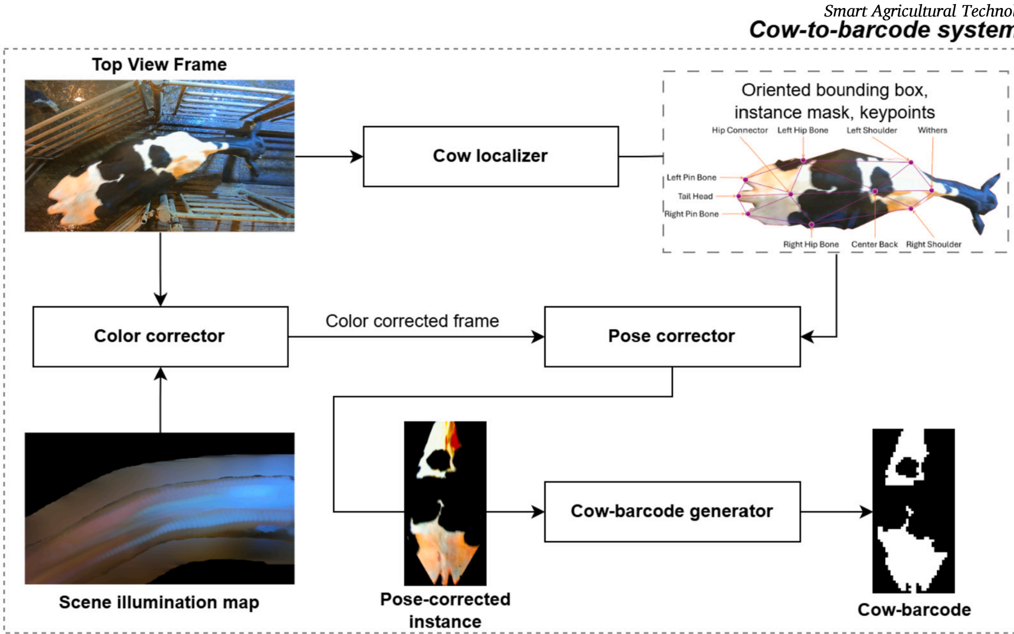

Fig. 3. Block diagram depicting the parts of the cow-to-barcode system. The oriented bounding box of each detected instance, and the cow-barcode if generated are passed to the cattle tracker. The figure in the top-right corner of the block diagram shows the ten keypoints on the back of the cow used for pose correction.

tify cattle using only the relative locations of keypoints were developed. While they perform reasonably well without depending on the color of cows, these methods could be sensitive to major errors in the detected locations of the keypoints that we know from experience can be introduced by keypoint detectors. Their design prevents them from using any keypoint error correction strategies as used by our AutoCattlogger. Additionally, they too train their classifiers using DML, which again generates non-interpretable intermediate representations, unlike our cow-barcodes. The solution in [26], of using keypoints to first estimate pose, and then creating a smaller subset of train-embeddings based on the pose for predicting the cattle identity using nearest neighbor matching, is resource-intensive.

Further, all the DL and DML based identification methods require large amounts of annotated training data, which is labor intensive to acquire. The system in [1] and [31] is designed to reduce human annotation effort by tracking cows and collecting all instances from the same track (or tracklet) under the same identity. The authors of [31] claim that their system can be used to label an entire herd with a few minutes of human annotation effort to combine tracklets of the same individual cows. The recognition system in [32] also needs to train from instances from multiple cameras to improve its accuracy. Meanwhile, the AutoCattlogger eliminates this problem entirely and practically reduces cattle annotation efforts to ZERO. This is because the AutoCattlogger is not data hungry, and can learn the identity of a cow from instances of a single cow track. It can also learn the identity of a cow instantly even from a single training image as demonstrated in Sec. 6.1.

3. Method

The AutoCattlogger system ingests unannotated, unsegmented videos of cattle in the top view and computes tracks for each individual cattle. Representative features for each of these individuals are then computed from their respective tracks. These features are then stored as the learned identity information of that individual. The same system can also be used to search for a given individual in an unsegmented video, or to identify an individual from a given track.

The AutoCattlogger is composed of two building blocks – the cow-to-barcode system that converts cow instances into barcodes in a predefined feature space, and the cattle tracker. These components, along with the input and outputs of the AutoCattlogger namely the Cattlog, the cattle tracks, and the start and end time-stamps of each track, are all shown in the block diagram in Fig. 1. Explanation of the working of the components of the AutoCattlogger is in Sec. 3.1 and Sec. 3.2. In Sec. 3.3, we explain how the barcodes from multiple instances of the tracked cows are combined to form a single barcode for each individual cow, which collectively form the Cattlog. Finally in Sec. 3.4, we explain the functioning of AutoCattleID for identifying cattle using the cattlog-barcodes.

3.1. Cow-to-barcode system

The cow-to-barcode subsystem is a modular system with checks and balances that is based on the Eidetic Cattle Recognition system [2,5]. Its working is shown in Fig. 3. This cow localizer first localizes all cows in the frame using instance masks and ten keypoints per instance. For this, it uses a Mask R-CNN model with a ResNet50-FPN backbone [33], and an HRNet Keypoint Detector [34]. An example image of a cow with all ten keypoints are shown in the same figure. Further, note that the mask and keypoint detectors are independent of individual identities, and hence are not re-trained when cows are added or deleted.

The cow localizer also handles keypoint detection errors such as missing and misplaced keypoints. To handle missing keypoints, the localizer uses an interpolator that leverages axial symmetry of the cow in the top view to estimate the left/right side keypoints. To estimate a missing keypoint on the spine of a cow, the interpolator selects a point on a second-order polynomial curve fit through keypoints detected on the spine. To classify an instance as having misplaced keypoints, the localizer uses a rules-checker with hand-crafted rules based on a shape model of the cow. This shape-model is defined by bounds on the ratio of distances between pairs of keypoints, ratio of angles between pairs of keypoint triplets, and distance of keypoints from the edge of the instance mask. Each rule checks a detected set of keypoints for a breach of bound. If any rule is broken, the instance is deemed unfit to be used for identification.

The localizer then applies a keypoint rectifier to recover some of these unfit instances. This keypoint rectifier, introduced in [5], applies a rules-checker as above and identifies the misplaced keypoint(s) by identifying the pattern in the list of broken rules. All misplaced keypoints are deleted, and then reinserted using the interpolator from above.

Next, the system applies a color-corrector from [6] to the video frame to reduce the effect of specular highlights from non-uniform scene illumination. This is needed because specular highlights on black regions on the cows can cause it to appear white and reduce the fidelity of its computed barcode.

This color-corrector uses a precomputed scene illumination map as pixel-level black-point values to apply black-point-correction to a given video-frame. These illumination maps are computed by using cows with fully black coats as Black-Mirror Light-Probes (BMLPs). Specifically, we measure and accumulate the pixel values on the surface of these cows as they move around in the scene. So, a farmer has to just make a black cow walk under the camera when recalibration of the lighting model is needed. Further, this color corrector is very effective as it samples the ground-truth non-uniform illumination directly from the scene and does not rely on illumination models learned from some other datasets. It is also fast because it does not compute image statistics or use neural-networks during inference.

The pose corrector uses the detected instance mask, keypoints, and the color-corrected video frame to compute images of the cow instances in a canonical pose with a straight spine and a fixed image-size as shown in Fig. 3. The bar-code generator pixelates and binarizes the pose-corrected images using a predefined threshold to produce the cow-barcodes.

These barcode images have a size of 512×1024px with blocks of size 16×16px. The barcodes are further serialized and stored as 2048 bit bit-vectors which serve as the identifying features for the individual cows.

3.2. The cattle tracker

We apply a simple online Multi Object Tracker (MOT) to the unsegmented videos to compute the trajectories for each individual cow. This tracker uses only an appearance model and does not include a motion model. The tracker tracks cattle by detection. That is, it compiles the detection outputs from the cattle localizer in Fig. 3 from every video frame into individual tracks. Because we work with top-view videos, chances of multiple tracked objects occluding each other are minimal. This implies that a simple tracker is adequate as the chance of identity switching is negligible.

Each detected cow instance is a track-point. A track is a list of track-points that our tracker decides to be of the same individual. Initially, the tracker opens a new track for every new track-point. Each track-point found in the later frames are matched with one of the open tracks and is added to it. A track is closed when no matching track-points are found in a few consecutive frames.

Each track-point always stores the video-name, frame number, and the corners of the oriented bounding box of the cow instance. It stores the cow-barcode only if the instance has all keypoints detected without errors. So, track-points of partly visible cow instances entering or exiting the scene are still attached to their tracks, but without any barcode information. For each track, cow-barcodes from specific track-points that contain them are averaged to form the cattlog-barcode, and this is saved along with the track information. The procedure to compute the cattlog-barcode is explained in Sec. 3.3.

AutoCattlogger uses a cattle object detector with a low confidence threshold to reduce false negatives and improve the detection recall metric of the tracker. This is to ensure that no cow is missed. This however increases false positives, leading to creation of tracks without any actual cows in them. But, all such false-positive tracks are easily rejected by checking for existence of at least one of their track-points with a stored cow-barcode. This improves the detection precision metric of the tracker. Together, this scheme ensures very high cattle object detection and tracking accuracy, and is another example of checks and balances in our design.

Further, the tracks can also store additional information about the entry and exit directions of the cow in the scene to help filter unwanted tracks. If the ground truth identities of the individual cows are available, as in our dataset (Sec. 4.2), we attach them to their respective tracks.

To match the track-points on a given frame with an existing open track, we follow the standard approach of using a bipartite graph matching algorithm. Specifically, we apply the Jonker–Volgenant algorithm [35] which performs maximum-weight-full-bipartite graph matching by formulating it as a Linear assignment problem. The bi-partite graph is constructed by considering track-points in the current frame as vertices in the first partition, and the last seen track-points of all open tracks as vertices in the second partition. The IoU values between pairs of oriented bounding-boxes of the track-points form the edge weights of the bi-partite graph. We open a new track for every track-point that has no found match, and close every open track that has no matching track-point. After completely processing all frames of all given videos, we close all the open tracks.

3.3. The cattlog

We define the process of computing the identifying features for the individual cows as 'cattlogging'. For every closed track with a cow, a bit-wise statistical mode is computed from a selected few of its barcodes to obtain the mode-barcode called the cattlog-barcode. The cattlog-barcodes are stored as a 2048 bit bit-vectors which serve as the identifying feature vectors for the cows.

The statistical mode averaging helps to even out the effects of scene illumination and reduce the influence of variation in locations of detected keypoints. The cattlog-barcodes from all tracks computed from the videos used for training the cattle recognition system are saved in the automatically generated Cattlog (the cattle-catalog).

The barcodes selected to compute the cattlog-barcode are the Top 20% of track-points based on the proximity of the centers of their oriented bounding boxes to the center of the video frame. Because we use top view cameras, such filtering ensures that we obtain cow instances with minimal perspective distortion, and in full view. The keypoint interpolator of the cow localizer (Fig. 3) can force-fit keypoints when parts of cows are beyond the edge of the video frame. So, using instances closer to the center of the frame prevents cattlogging from instances with force-fitted keypoints.

3.4. AutoCattleID

AutoCattleID, the cattle recognition application built atop the AutoCattlogger leverages the contextual information present in the cattle videos in the form of cattle tracks. To recognize any cow from a video, AutoCattleID first computes the track information of the cow as it walks across the scene. It then matches the cow barcode at every accumulated track-point with its nearest neighboring cattlog-barcode using the Hamming distance metric. It then declares the identity associated with the cattlog-barcode with the highest number of matches to be the predicted identity.

4. Datasets and data collection

This section details the collection and curation of the video datasets and the ground-truth track annotations used to evaluate the AutoCattlogger system. Details of datasets used to train the components of the AutoCattlogger, and the datasets used for evaluating the cattle recognition systems from our previous works [2,5,6] are also provided in the in Apx. B and Apx. C for completeness.

4.1. Data procurement

We mount a top-view camera in the holding area of the Purdue Dairy so it can continuously record a path along which cows walk in a single file after being milked. The recorded videos have a resolution of 1920×1080 px, frame rate of 30FPS, use H.264 encoding, and are stored

Table 1
Number of cows that are common between any two days. The numbers on the principle diagonal represent the number of cow individuals present on each corresponding day.

	S22-Day1	S22-Day2	S23-Day1	S23-Day2
S22-Day1	153	148	81	80
S22-Day2	148	169	93	90
S23-Day1	81	93	177	149
S23-Day2	80	90	149	175

as contiguous segments that are each an hour long. We call these the raw videos or the hour-long videos. Most of these hour-long videos are generally empty as the cows are milked only twice a day. So, we retain only those videos that were recorded during the milking hours. Separate sets of these hour-long videos are used to create the datasets for evaluating the AutoCattlogger system, as explained next.

4.2. The raw videos datasets

Sets of continuous, hour-long videos with cows in them were recorded on two consecutive days of Summer (June) 2022, and two days a month apart in June and July of Summer 2023. We refer to these sets of videos by S22-Day1, S22-Day2, S23-Day1, and S23-Day2. Workers in the Purdue Dairy recorded the ground-truth cow-labels in the same order in which the cows walked by under the top-view camera, in a CSV file called the ‘human-record’. We call the ground-truth labels the ‘cow IDs’ (short for cow identities), which are usually unique four digit numbers seen on the ear-tags of the cows. The four sets of hour-long videos from the four different days, together with their human-record annotations constitute the Raw Videos Datasets. We use these datasets to create the cow-tracks for evaluating our AutoCattlogger system.

The experiments in Sec. 6.1, 6.3 and 6.4 use these datasets. Note that not all cows are present in the videos of all the days. Table 1 gives the number of cows that are common across any two given days.

4.3. Ground-truth track annotations

The tracker used by our AutoCattlogger is sufficiently accurate to retrieve every single cow from the set of videos that it ingests. Since the videos in the Raw Videos Datasets also have cows that are undocumented in the human-record, we automatically filter the data using the entry and exit directions, and consider only those cows that appear within the duration of ground-truth recording. The number of cow tracks found after this filtering is the same as the number of cows in the human-record. After forming this one-to-one association, we annotate every required track with its cow ID from the human-record. Detailed explanation of the operations of the cattle-tracker, contents of a track-object, and the track filtering techniques are in Sec. 3. We use the videos in the Raw Videos Datasets in conjunction with the above ground-truth track annotations to evaluate the AutoCattlogger, using the evaluation procedure detailed in Sec. 5.

5. Evaluation method and metrics

This section details the evaluation methodology and metrics used to assess the performance of the AutoCattlogger. The tracking system is sufficiently accurate to retrieve every single cow from the set of videos that it ingests. So, we evaluate only the quality of the identifying barcode features that are generated by the AutoCattlogger in the experiments in Sec. 6.

Performance consistency of downstream video analytics tasks built on top of the AutoCattlogger requires the computed barcodes of cows to be consistent across days. To measure this consistency, we evaluate the AutoCattlogger using the downstream task performance of cattle identification by utilizing the video data of the same cows from multiple days in our datasets. In doing so, we not only prove the robustness of AutoCattlogger to time-varying factors such as background, illumination, and cow sizes across years, but also prove its worth in supporting a cattle identification system.

For this evaluation, we use the Cattlog from one of the days of our datasets as our training barcodes to recognize cows in tracks from the other days using AutoCattleID (Sec. 3.4).

We evaluate the cattle recognition performance using two metrics: the track level accuracy that measures the performance of AutoCattlogger based recognition (AutoCattleID) considering the video context, and the instance level accuracy that measures the performance of the same on random instances drawn *without any video context*. In cases where there are no common denominators available to compute the accuracy as a percentage, we use the number of correct identifications metric in place of the instance level accuracy. For all these metrics, higher numbers indicate better performance, and better cattle identification performance indicates the ability of AutoCattlogger to generate consistent barcodes for the same cow.

1. **Track Level Top-K accuracy:** The track level top-K accuracy measures the proportion of times the correct cow ID is among the top K predictions at the track level. The top-K predictions at the track level are obtained by collecting the predicted identity for every instance in the track and then arranging them in decreasing order of frequency of occurrence. The maximum number of different ID predictions for the track instances among all the tracks is the maximum meaningful value of K. Therefore, increasing the value of K beyond this maximum value will not increase track-level accuracy. The experiments in Sec. 6.1, 6.2, and 6.3 use this metric.
2. **Instance level Top-K accuracy:** This is a standard classification accuracy metric. The instance level top-K accuracy is the proportion of times the correct cow ID is among the top-K predictions at the instance level. These top-K predictions for a given cow instance are the top-K nearest neighbors in the Cattlog. The experiments in Sec. 6.1, and 6.3 use this metric.
3. **Number of correct identifications:** As the name suggests, this metric represents the total number of instances for which the predicted cow ID is the correct cow ID. The experiments in Sec. 6.2 and Sec. 6.4 use this metric because the total number of instances available for generating the barcodes varies among the recognition systems that are being compared resulting in no common denominator.

The track level top-K accuracy is analogous to the video level Top-K accuracy metric used to present the results in [2,5]. The only difference is that the recognition systems in [2] and [5] were evaluated on human-cut video segments that contain only one cow each, and here, the AutoCattlogger automatically finds the tracks of each cow. So, in Sec. 6, we compare the Track level Top-K accuracy values from AutoCattleID with the Top-K Video level accuracy values from [2] and [5]. Since both these metrics measure the accuracy at the level of cow-individuals, we collectively refer to them by **‘Cow Level Top-K Accuracy’**.

Note that, due to the sheer number of instances in our datasets, the instance level accuracy uses a denominator that is in the order of 10^3 or 10^4 instances. This makes them vulnerable to biases from easily identifiable cows lingering in the scene for longer durations. So, the instance-level accuracy can be misleadingly high even if there are multiple cow individuals that are never correctly identified. However, these cows that are never correctly identified will not bias the track-level accuracy. Hence, the track-level accuracy presents a normalized version of the recognition performance.

In our datasets, we find three cows with completely black backs. These cows were assigned the same fully black barcode, which corresponds to an all zero bit-vector. So, in all our evaluations, we treat the recognition of a black cow as any other black cow as a correct recognition. We call this ‘discounted evaluation’.

Table 2

Experimental results from comparison with Deep Metric Learning method for cow identification [24]. The AutoCattleID method proposed in this paper is marked with an asterisk (*). All the support-set instances come from S22-Day1. The highest accuracy value in each result column is highlighted in bold.

Training Information				Instance level Top-1 Accuracy (%) (% correctly identified instances)			Cow/track level Top-1 Accuracy (%) (% correctly identified cows)		
Case #	Method	Train & Val. Datasets	# Support-set Instances	S22-Day2 (7698 inst.)	S23-Day1 (6613 inst.)	S23-Day2 (4024 inst.)	S22-Day2 (148 cows)	S23-Day1 (81 cows)	S23-Day2 (80 cows)
1	DML [24]	OpenCows2020 [24]	7642	52.34	20.02	23.88	88.51	45.68	37.50
2	DML [24]	S22-Day1 & S22-Day2	7642	84.71	36.82	49.06	93.92	58.02	60.00
3	Auto CattleID*	(20% images per cow)	1471	**90.48**	**84.83**	85.98	**95.27**	**92.59**	91.25
Experimental cases on One Shot Learning									
4	DML [24]	OpenCows2020 [24]	153	30.94	11.89	14.89	56.08	22.22	22.50
5	Auto CattleID	(1 image per cow)	153	87.62	83.47	**87.40**	**95.27**	**92.59**	**92.50**

6. Experiments and results

This section presents the results of evaluating the AutoCattlogger based on cattle recognition performance. The experiments in this section are designed to answer the following research questions.

1. *Is our approach of using a predefined embedding space of cow-barcodes better than using an embedding space obtained from Deep Metric Learning for cattle identification?* The results in Sec. 6.1 show how our approach is better.
2. *Is our tracker based AutoCattleID better than our own previous approaches?* This is answered in Sec. 6.2.
3. *Can AutoCattleID maintain its performance when cross-evaluated on multi-year data?* This is answered in Sec. 6.3.
4. *How well does AutoCattleID perform when parts of the AutoCattlogger are removed?* This ablation study is in Sec. 6.4.

Finally, we conclude this section with the discussion of our results in Sec. 6.5.

6.1. AutoCattleID vs. Deep metric learning

In this section, we compare the performance of our method with that of the Deep Metric Learning cow identification method [24] on our datasets. We choose this method among those discussed in Sec. 2, because this is the only other method designed to adapt to new individuals without the need for model retraining. This method is also the foundation for other subsequent works from the same authors including [1,31,32].

Here, it is important to note the difference between the dataset used to train the embedding model itself using some contrastive learning technique, and the training dataset from the new herd that is used to get the support set of embeddings for cow identification. For convenience, we call the latter dataset the support-set and call the embeddings derived from it the support-embeddings. Furthermore, for experiments with the DML models involving our data, the instances in the support-set, testing set, and the training set where needed, are generated by the AutoCattlogger itself. All cow instances that the AutoCattlogger utilized for generating a barcode are cropped to their oriented bounding boxes to form the samples in the datasets mentioned above.

We present five experimental cases in this section. The instances in the support-sets for all cases are sampled from S22-Day1. This implies that the support-embeddings for the DML cases and the Cattlogs for the AutoCattleID cases are created from instances sampled from the S22-Day1 dataset. All cases are evaluated on the instances sampled from S22-Day2, S23-Day1, and S23-Day2.

Case 1 explores the adaptability of a learned DML embedding space to a completely new herd. This case uses the embedding model from OpenCows2020 dataset to generate the support-embeddings for the cows in our S22-Day1 dataset with 7642 instances from 153 cows. Case 2 tests the accuracy of the DML system when we train the embedding model on cows from our own herd using the S22-Day1 dataset. The weights from the epoch that resulted in the best accuracy on the S22-Day2 validation dataset with 7698 instances from 148 cows are selected as the final model weights. The support-embeddings also come from the same set of 7642 instances from S22-Day1. Case 3 presents the results from our preferred AutoCattlogger based cattle identification method – AutoCattleID – that uses only the top 20% of instance per cow based on their proximity to the center of the frame to learn its identity.

Next, to demonstrate the one-shot learning ability of our AutoCattleID method, we include cases that force the selected methods to learn cow identities using just one instance per cow. This ability is useful when very little training data is available for some or all of the cows. For each of the 153 training cows, AutoCattlogger automatically fetches this instance from the track-point closest to the center of the frame to avoid image distortions. Both Case 4 and 5 use this support-set. Case 4 uses the same DML embedding model from Case 1, but generates the support-embeddings from only one image per cow. Exp 4 also uses a K value of 1 instead of the default value of 5 for K-NearestNeighbors matching to predict the cow identity [24]. In Case 5, we make the AutoCattlogger learn the cow identities from the same set of 153 images as in Case 4.

The results of the comparison are presented in Table 2 with both the Instances and Cow/Track level Top-1 accuracy metrics. For all cases on DML cattle identification, the 'Train & Val. Datasets' column has the name of the dataset used to train the embedding model, and the validation set used to select the best model weights. For cases 1 and 4, separate training and testing dataset of the OpenCows2020 dataset were used for training and validation. Further, since our AutoCattlogger does not involve training embedding-models, the 'Training Dataset' column has additional information about the training datasets.

The DML model trained on all training instances of the OpenCows2020 dataset produced validation accuracies of 99.8% (instance level) and 100% (cow level) when evaluated on the test set of the OpenCows2020 dataset [24]. However, we see from Case 1 in Table 2 that this learned embedding space on the smaller dataset with 46 cows does not adapt well to our dataset with 153 cows. From Case 2, we observe that training the DML embedding model on all training instances of our own S22-Day1 dataset produces good results on S22-Day2. However, this performance dips greatly when evaluated on data from the same cows a year later. The accuracy results from our proposed method in Case 3 are very good, even on data from a year later. From the results of Case 4 and 5, we see that our AutoCattleID method outperforms DML by a huge margin. This proves that our AutoCattlogger can learn from single images of cows when their videos are not available.

Note that for both the AutoCattleID cases, Case 3 and Case 5, the color corrector of the AutoCattlogger uses illumination maps from the evaluation days (see Fig. 3). This calibrates the system to accommodate the variations in scene illumination conditions. The system automat-

Table 3
Number of correct identifications. The results for the recognition system in BMLP, and AutoCattleID include 65 discounted instances – because they both find three equivalent completely black cows in the training set which get to share the same fully black barcode.

Cattle ID Systems	# correct identifications
ECR [2]	1211
SURABHI [5]	2279
BMLP [6]	4198
AutoCattleID	**7042**

ically generates these illumination maps once we mark a cow to be completely black. More details on this strategy are presented in Sec. 6.3.

These experimental results prove that by using a predefined embedding space of cow-barcodes, AutoCattlogger enables us to build very efficient and accurate cattle recognition systems. Further, we found that using additional training images per cow in Case 3 rather than just one per cow in Case 5 helps reduce noises due to lighting and perspective variations, and obtain better quality cow-barcodes.

6.2. AutoCattleID vs. Our previous works

The Eidetic Cattle Recognition (ECR) system [2] that identifies cows using only one training instance underwent multiple improvements to reach its current form – AutoCattleID. In this section, we show that the current iteration of our cattle recognition system based on AutoCattlogger performs better than all the previous iterations.

All versions of our cattle recognition system in this comparison are evaluated on cut-videos of S22-Day2, which are re-encoded segments of the hour-long videos with only one cow in them (Apx. C). Also, the Cattlogs for all versions are generated from cut-videos of S22-Day1. Cut-videos are used in this section because the previous versions of our cattle recognition system lack the ability to track individual cows. Thus, the Cattlog for AutoCattleID is generated by running the AutoCattlogger on the same set of re-encoded cut-videos from S22-Day1 as used by our previous works. Also, for consistency, here we use the illumination map from S22-Day1 to color correct video frames from S22-Day2 just like our previous work [6].

For comparing the performance of these system versions on cow instances, we use the total number of correct identifications metric due to the lack of a common denominator for comparison. This is because these different versions of cattle recognition systems use different cow localizers, and hence the number of cow instances available for barcode generation (the denominator) varies.

Table 3 compares the total number of correct identifications on the cut-videos of the common cows on S22-Day2, from all our works on cattle recognition. We observe a dramatic improvement in performance from version to version, culminating with the current method.

The plot in Fig. 4 presents the Cow Level Top-K Accuracy values. This Cow Level Top-K Accuracy refers to the Top-K Video Level Accuracy metric for all our previous work, and to the analogous Track Level Top-K Accuracy metric for the results from AutoCattleID. This Top-K accuracy metric reaches 100% for the first time ever for a K value of 3.

6.3. Multi-year cross-evaluation

In Sec. 6.1, we used the Cattlog from S22-Day1 to identify cows in all the other days in our datasets. In this section, we apply the same evaluation strategy to Cattlogs from every available day in our multi-year datasets. We thus cross-evaluate AutoCattleID on data from multiple years and provide detailed results.

The aim of these evaluations is to verify that the recognition system can maintain its accuracy despite possible variation in cow sizes due to growth or pregnancy, scene backgrounds, and illumination conditions. This is a more robust evaluation than the traditional ‘leave one day out’ evaluation. Our system learns the identities from only one day, and AutoCattleID is evaluated on all the other days.

We note that the scene illumination can vary considerably across months due to factors such as seasonal changes and weather if the barn gates are open to outside light, changing of lamps in the dairy etc. An example of variation in color of the coat of the same cow on differ-

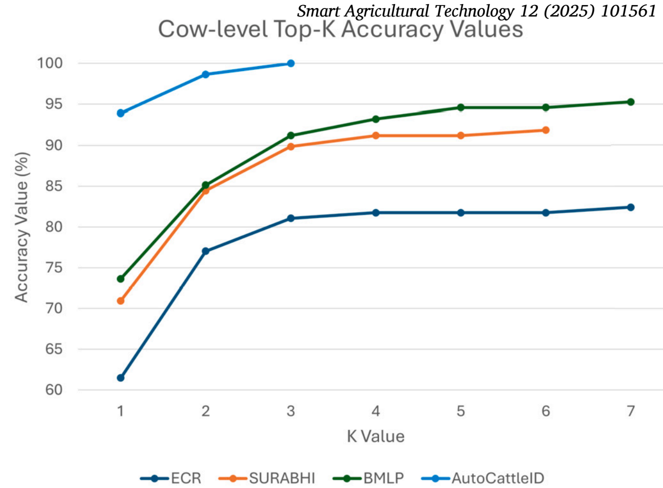

Fig. 4. Comparison of Cow-level Top-K Recognition accuracy values from ECR [2], SURABHI [5], BMLP [6], and the current AutoCattleID.

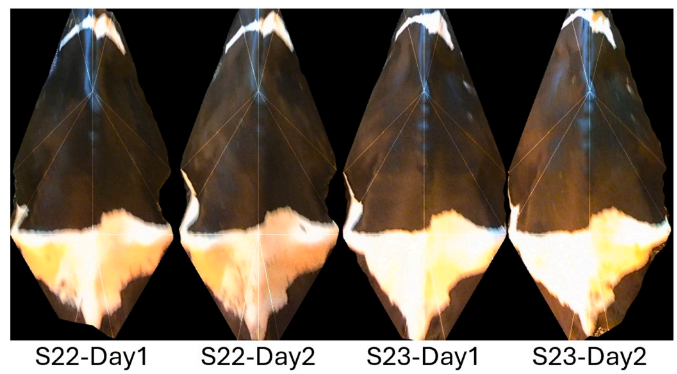

Fig. 5. Variation in coat color of the same cow on the four different days due to changes in illumination conditions.

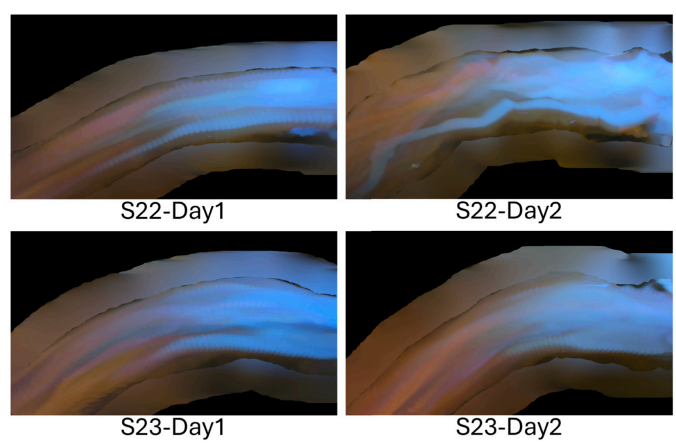

Fig. 6. The scene illumination maps from our four different data days. Observe the variation in hue and intensity at the same points in the illumination maps from the different days.

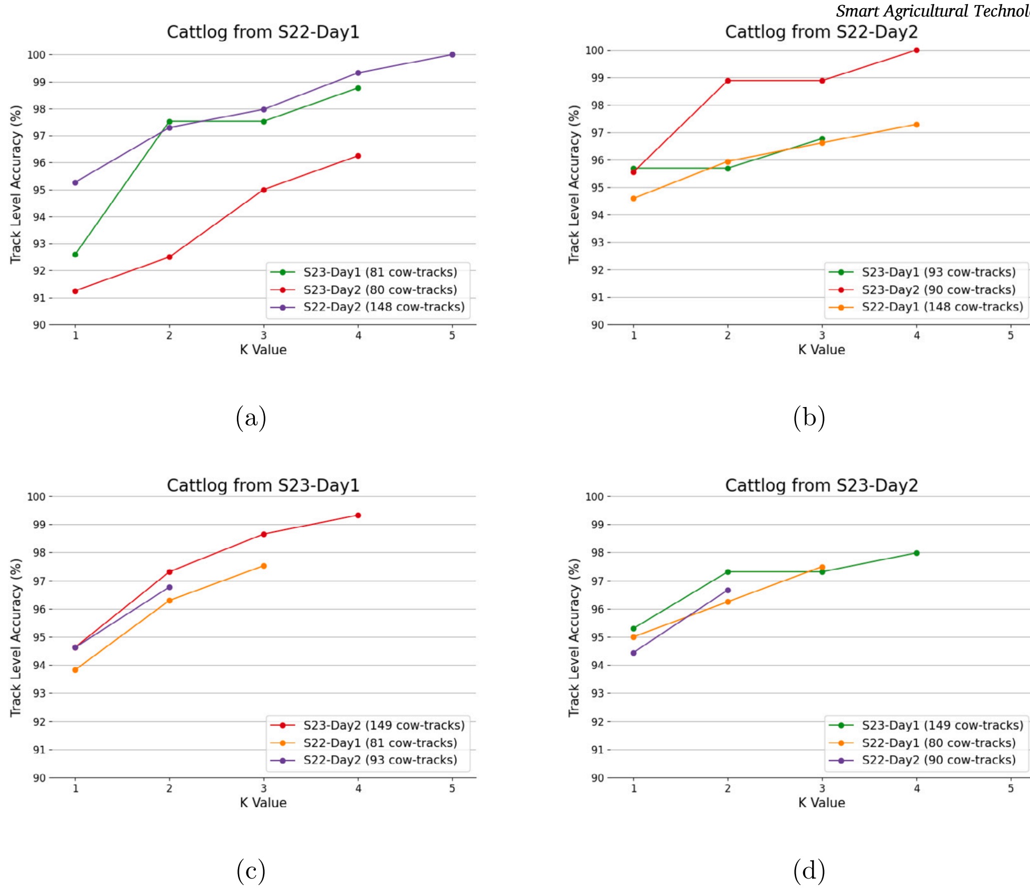

Fig. 7. Track-level Top-K Accuracy values for training data from different days. Track-level accuracy measures the performance of AutoCattleID on cows considering the video/track context.

ent days due to changes in illumination conditions is shown in Fig. 5. Therefore, we make accommodations for these scene lighting variations for improved AutoCattleID accuracy. So, unlike Sec. 6.2 where we used the illumination maps from S22-Day1 for color correcting videos from S22-Day2, here, and similar to Sec. 6.1, we compute and use the illumination maps directly from the evaluation days. These illumination maps are shown in Fig. 6. Additionally, to evaluate real-world performance, in this section, we run experiments directly on the two Raw Videos Datasets (Sec. 4.2).

The track level results are in Fig. 7 and the instance level results are shown in Fig. 8. The plots in Fig. 7 reveal that the cow level Top-1 accuracy is consistently above 90%. In more practical terms, this means that for over 90% of the tracked cows, AutoCattleID correctly identifies them as the most likely candidate from the database. The Top-K accuracy eventually rises beyond 90% with K, and also reaches 100% in two cases.

Upon evaluating the recognition system on out-of-context images, from the plots in Fig. 8, we find the instance level Top-1 accuracy hovering around the 80 to 90% interval. This accuracy is almost always above 95% for $k = 5$.

6.4. Ablation study

To determine the contribution of each component of the AutoCattlogger, we measure the number of correct identifications by AutoCattleID by removing the components one at a time. Specifically, we conduct an ablation study where AutoCattleID uses the Cattlog from the raw-videos of S22-Day1 and is evaluated on all the other days in our raw-videos datasets (Sec. 4.2). This study measures the performance of AutoCattleID when the keypoint rectifier of the cow localizer in Fig. 3, and the color corrector are removed.

Recall from Sec. 3.3 that the AutoCattlogger uses only the top 20% of track-points based on their nearness to the center of the camera frame for cattlogging. This is to avoid errors due to perspective distortions and force-fitted keypoints. We also study the consequences of ignoring this strategy to include all available instances for cattlogging.

Again, as mentioned in Sec. 5, these results will not be presented as percentages as all the rows do not share a common denominator. The number of correctly localized instances available for identification (the denominator for accuracy measurement) is lower when the keypoint rectifier is removed.

The results from all experiments in this ablation study, along with the baseline results from the full AutoCattlogger system are in Table 4. From the results, we see that the performance drops only slightly when the keypoint rectifier is removed. This shows that the accuracy of the HRNet based keypoint detector is adequate. Omission of the color corrector leads to different levels of performance dips for different days. This means that the color corrector can help save a good number of instances from the troubles of non-uniform scene lighting.

Lastly, we see that using all available instances to create the cattlog-barcode has a strong negative impact. As explained earlier, the keypoint detector tries to force-fit keypoints on plausible looking partial images of cows as they cross the frame boundaries. Looking at cows from an off-normal angle may also hide parts of their coat patterns. So, by selecting only the 20% of instances closest to the center of the video frame, our method filters many unsuitable instances out before creating the cattlog-barcode.

It is also interesting to note the slight difference in performance when cattlogging and evaluating on raw videos versus the re-encoded cut-videos. The performance on cut-videos as seen in the last row of Table 3, 7042, is slightly higher than that on raw videos as seen in the first entry of Table 4, 6965. In both these cases, the Cattlogs are generated

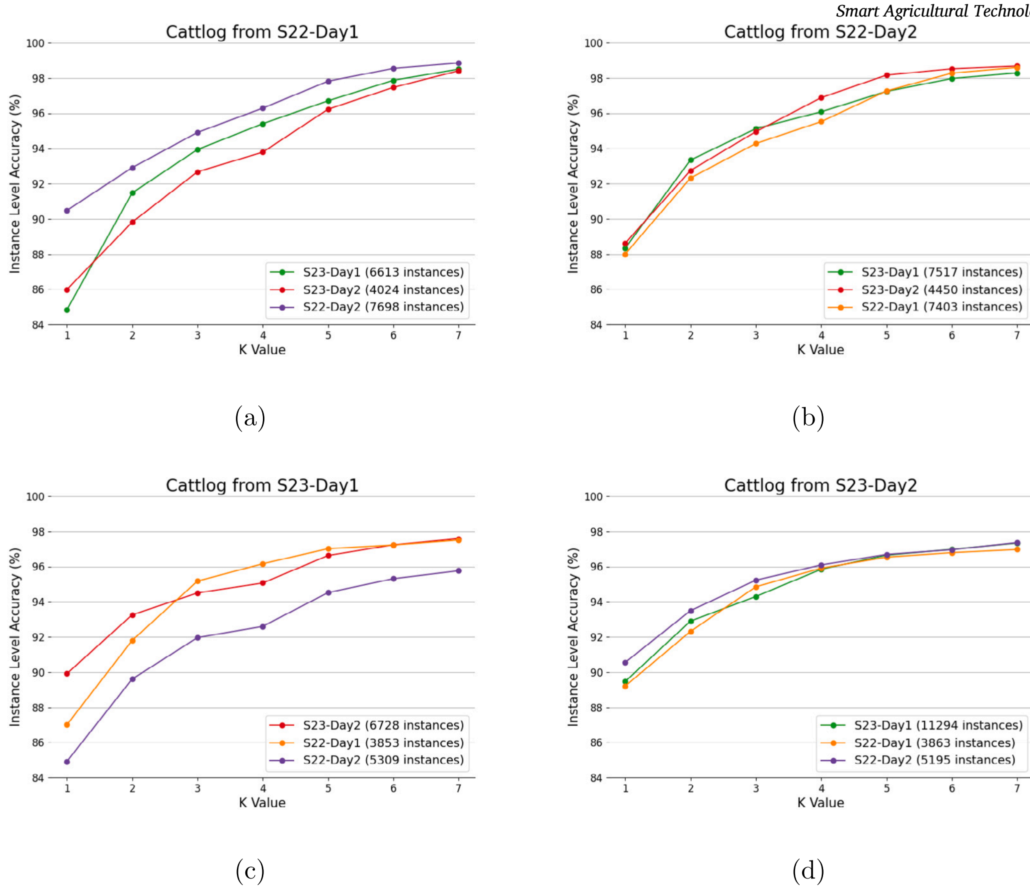

Fig. 8. Instance-level Top-K Accuracy values for training data from different days. Instance-level accuracy measures the performance of AutoCattleID on cow instances without the video/track context.

Table 4
Results of the ablation study. The values in the table represent the 'Number of correct identifications'.

Cattlog from S22-Day1			
Description	S22-Day2	S23-Day1	S23-Day2
Full system	**6965**	**5610**	**3460**
Without keypoint rectifier	6944	5603	3450
Without color corrector	6687	5407	3459
Cattlogging from all instances	6094	4576	2983

using data (be it raw or cut-videos) from S22-Day1, and the recognition system is evaluated on data (be it raw or cut-videos) from S22-Day2. From further inspection, we found that the cow-localizer (Fig. 3) provided more correctly localized instances from MJPG encoded cut-videos (Apx. C) than from the H.264 encoded raw videos (Sec. 4.2). This could be attributed to the keypoint (and mask) detector of the cow-localizer being trained on JPEG encoded images as mentioned in Apx. B. Thus, differences in encoding type between training data and evaluation data for a deep-learning based keypoint detector can negatively impact its performance [36].

6.5. Discussion

The results above demonstrate that the barcodes generated by the AutoCattlogger are mostly consistent, even from videos recorded after a year. It is fascinating to see a system that uses a simple identity feature space can perform so well. However, despite the impressive Top-K accuracy numbers, we still need a 100% Top-1 accurate recognition system for it to be viable for real-world deployment. In its current state, the system could be used to assist human observers by producing Top-K predictions for a given cattle instance.

The system is dependent on the proper functioning of its components. Sometimes, errors creep in despite the checks and balances that we have built into the system. Fig. 9 shows an example where the cattle tracker identified two cows in close proximity as just one cow. This leads to track-ID switching in the subsequent frames. However, this can be improved by using a better cattle mask detector, by improving the keypoint rules-checker from Sec. 3.1, or by using a slightly more sophisticated tracker. The modularity of the AutoCattlogger allows easy alterations of such components.

Again, fully black cows play an important role in our system. Although [31,1] exclude black cows entirely during evaluation, we use them to generate scene illumination maps to help identify all the other cows better. The results would look very much the same even if we excluded the three black cows from evaluation.

7. Conclusion

With a user centric approach, we address the needs of a dairy farmer with continuously changing herd by developing an automatic cattle identity registration system, the AutoCattlogger. This is enabled by the use of a predefined and interpretable identity space that allows us to directly add and delete identities. The AutoCattlogger is modular and allows us to upgrade its components such as the keypoint and mask detectors, the keypoint rectifier, and even the identity space if necessary. We find that a cattle recognition application based on the AutoCattlogger – AutoCattleID – performs well in identifying cattle even after they have aged a year. All this performance can be harnessed with zero training time and no human annotation effort.

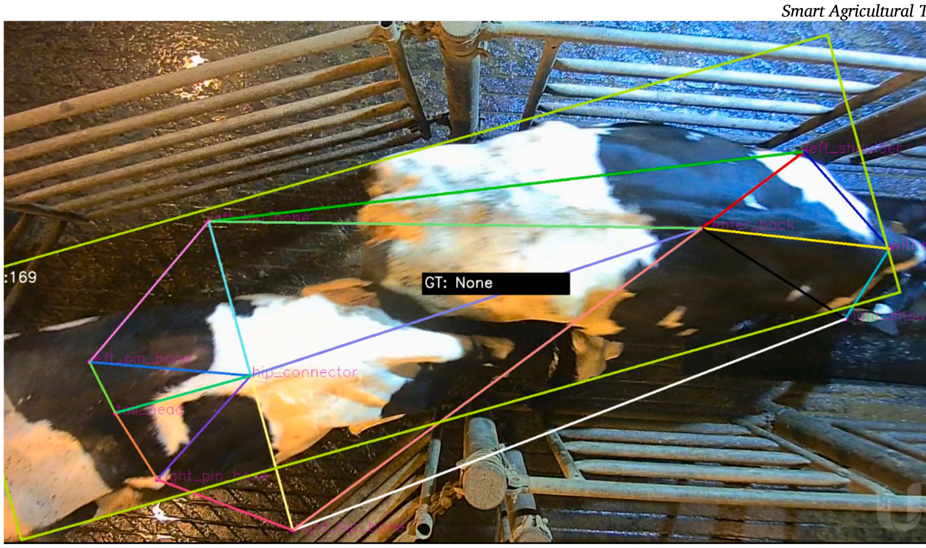

Fig. 9. An example where the mask detector detects two cows that are very close together as a single individual, resulting in a track-ID switch in the subsequent frames.

Further, the AutoCattlogger utilizes video context that is commonly ignored by other cattle recognition systems. This video context gets embedded in the cattle-track information. Specifically, the AutoCattlogger

- uses track information to select the best instances for cattlogging, and,
- via AutoCattleID, uses a majority vote on track-point identity predictions to estimate the cow identity during inference.

Future work could aim for improving Top-1 cattle recognition accuracy by including more identifying features per individual cow. It could include an improved tracker with a motion model as in [37], [38], and [39] complementing our appearance model to overcome the identity switching problem discussed above in Sec. 6.5. Additionally, it could also use depth features to distinguish between cows having the same fully black coat patterns.

CRediT authorship contribution statement

Manu Ramesh: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Amy R. Reibman:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Formal analysis.

Declaration of competing interest

This work was made possible because of the data collected by students and workers at the Purdue Dairy.

Appendix A. Details of training the models

The mask detector uses a Mask R-CNN [33] model with a ResNet50-FPN backbone. This was trained using the Detectron2 [40] framework, on NVIDIA A6000 GPUs (NVIDIA, Santa Clara, CA, USA) for 50,000 iterations with a batch size of 8. This model is the same as the one used in [2]. The keypoint detector uses an HRNet [34] model trained using the MMPOSE [41] framework on the same NVIDIA A6000 GPUs for 210 epochs with a batch size of 6.

The embedding models of the Deep Metric Learning methods discussed in Sec. 6.1, were trained on the same NVIDIA A6000 GPUs with a batch size of 16 and for 100 epochs – as per the descriptions in the source paper [24].

Appendix B. Datasets for training sub-components

The AutoCattlogger uses instance mask and keypoint detector models that were established in our previous works [2,5,6]. These mask and keypoint detectors were trained on the Cow Keypoints Dataset – with keypoint and mask annotations. The training and validation images for this dataset were sampled from the videos in S22-Day1 and S22-Day2 respectively. This training set also has images sampled from raw-videos from the holding-area in Summer 2021, and the barn area in Summer 2022. The barn area is where cows can walk unconstrained by fences. All these images are encoded in JPEG format.

Appendix C. Datasets for older cattle recognition systems

The cattle recognition systems in our previous works did not have the capability to track individual cows in video frames. They used segments from the raw videos datasets from Sec. 4.2, with each segment containing only one cow walking across the scene. These video segments are called ‘cut-videos’, and are encoded in MJPG format. Our older recognition systems used one or more sampled images for each cow from cut-videos of S22-Day1 to create their Cattlogs. The recognition systems were evaluated on cut-videos from S22-Day2. During this evaluation, we ensured that only those cows in S22-Day2 that were also in the training day S22-Day1 were used.

The experiments in Sec. 6.2 are run on this dataset of cut-videos. Further, no data from the year 2023 was used in our previous works. So, we do not have cut-videos from Summer 2023.

Data availability

The data that has been used is confidential.

References

[1] J. Gao, T. Burghardt, W. Andrew, A.W. Dowsey, N.W. Campbell, Towards self-supervision for video identification of individual Holstein-Friesian cattle: the Cows2021 dataset, arXiv:2105.01938, 2021, http://arxiv.org/abs/2105.01938.

[2] M. Ramesh, A.R. Reibman, J.P. Boerman, Eidetic recognition of cattle using keypoint alignment, Electron. Imaging 35 (7) (2023) 279-1, https://doi.org/10.2352/EI.2023.35.7.IMAGE-279, https://library.imaging.org/ei/articles/35/7/IMAGE-279.

[3] H. Liu, A.R. Reibman, J.P. Boerman, Feature extraction using multi-view video analytics for dairy cattle body weight estimation, Smart Agric. Technol. 6 (August 2023) 100359, https://doi.org/10.1016/j.atech.2023.100359.

[4] J. Lyu, M. Ramesh, M. Simonds, J.P. Boerman, A.R. Reibman, Automatic retrieval of specific cows from unlabeled videos, arXiv preprint, Anim. Sci. Proc. 16 (4) (2025)

653–655, https://doi.org/10.1016/j.anscip.2025.08.242, 3rd U.S. Precision Livestock Farming Conference (USPLF 2025), https://www.sciencedirect.com/science/article/pii/S2772283X25008271.

[5] M. Ramesh, A.R. Reibman, SURABHI: self-training using rectified annotations-based hard instances for eidetic cattle recognition, Sensors 24 (23) (2024) 7680, https://doi.org/10.3390/s24237680, https://www.mdpi.com/1424-8220/24/23/7680.

[6] M. Ramesh, A.R. Reibman, Black-mirror light-probe cow model for specular highlight removal to improve Holstein cattle identification, in: 2025 IEEE International Conference on Image Processing Challenges and Workshops (ICIPCW), 2025, in press.

[7] A. Zhao, H. Wu, D. Fan, K. Li, Individual cow recognition based on ultra-wideband and computer vision, Animals 15 (3) (2025), https://doi.org/10.3390/ani15030456.

[8] N.S. Russel, A. Selvaraj, Two stream CNN for muzzle print recognition using Gabor filters, Multimed. Tools Appl. 84 (2025), https://doi.org/10.1007/s11042-025-20604-9.

[9] A.I. Awad, M. Hassaballah, Bag-of-visual-words for Cattle identification from muzzle print images, Appl. Sci. 9 (22) (2019) 4914, https://doi.org/10.3390/APP9224914, https://www.mdpi.com/2076-3417/9/22/4914/htm.

[10] S. Kumar, S.K. Singh, Automatic identification of cattle using muzzle point pattern: a hybrid feature extraction and classification paradigm, Multimed. Tools Appl. 76 (2017) 26551–26580, https://doi.org/10.1007/s11042-016-4181-9.

[11] S. Kumar, S.K. Singh, R.S. Singh, A.K. Singh, S. Tiwari, Real-time recognition of cattle using animal biometrics, J. Real-Time Image Process. 13 (3) (2017) 505–526, https://doi.org/10.1007/S11554-016-0645-4.

[12] A. Saygılı, P. Cihan, C.Ş. Ermutlu, U. Aydın, Ö. Aksoy, CattNIS: novel identification system of cattle with retinal images based on feature matching method, Comput. Electron. Agric. (2024).

[13] X. Xu, H. Deng, Y. Wang, S. Zhang, H. Song, Boosting cattle face recognition under uncontrolled scenes by embedding enhancement and optimization, Appl. Soft Comput. 164 (March 2024) 111951, https://doi.org/10.1016/j.asoc.2024.111951.

[14] S. Han, A. Fuentes, J. Park, S. Yoon, J. Yang, Y. Jeong, D.S. Park, Utilizing farm knowledge for indoor precision livestock farming: time-domain adaptation of cattle face recognition, Comput. Electron. Agric. 234 (August 2024), https://doi.org/10.1016/j.compag.2025.110301, 2025.

[15] N. Bergman, Y. Yitzhaky, I. Halachmi, Biometric identification of dairy cows via real-time facial recognition, Animal 18 (3) (2024) 101079, https://doi.org/10.1016/j.animal.2024.101079.

[16] Z. Weng, F. Meng, S. Liu, Y. Zhang, Z. Zheng, C. Gong, Cattle face recognition based on a two-branch convolutional neural network, Comput. Electron. Agric. 196 (March 2022) 106871, https://doi.org/10.1016/j.compag.2022.106871.

[17] L. Yang, X. Xu, J. Zhao, H. Song, Fusion of RetinaFace and improved FaceNet for individual cow identification in natural scenes, Inf. Process. Agric. 11 (4) (2023) 512–523, https://doi.org/10.1016/j.inpa.2023.09.001.

[18] H.H. Dac, C. Gonzalez Viejo, N. Lipovetzky, E. Tongson, F.R. Dunshea, S. Fuentes, Livestock identification using deep learning for traceability, Sensors 22 (21) (2022) 8256, https://doi.org/10.3390/S22218256, https://www.mdpi.com/1424-8220/22/21/8256/htm.

[19] T.T. Zin, M.Z. Pwint, P.T. Seint, S. Thant, S. Misawa, K. Sumi, K. Yoshida, Automatic cow location tracking system using ear tag visual analysis, Sensors (Switzerland) 20 (12) (2020) 3564, https://doi.org/10.3390/s20123564.

[20] T. Gao, D. Fan, H. Wu, X. Chen, S. Song, Y. Sun, J. Tian, Research on the vision-based dairy cow ear tag recognition method, Sensors 24 (7) (2024), https://doi.org/10.3390/s24072194.

[21] A. Pretto, G. Savio, F. Gottardo, F. Uccheddu, G. Concheri, A novel low-cost visual ear tag based identification system for precision beef cattle livestock farming, Inf. Process. Agric. 11 (1) (2024) 117–126, https://doi.org/10.1016/j.inpa.2022.10.003.

[22] M. Smink, H. Liu, D. Dopfer, Y.J. Lee, Computer vision on the edge: individual cattle identification in real-time with ReadMyCow system, in: Proceedings - 2024 IEEE Winter Conference on Applications of Computer Vision, WACV 2024, 2024, pp. 7041–7050.

[23] A. Bhole, S.S. Udmale, O. Falzon, G. Azzopardi, CORF3D contour maps with application to Holstein cattle recognition from RGB and thermal images[formula presented], Expert Syst. Appl. 192 (December 2021) (2022) 116354, https://doi.org/10.1016/j.eswa.2021.116354.

[24] W. Andrew, J. Gao, S. Mullan, N. Campbell, A.W. Dowsey, T. Burghardt, Visual identification of individual Holstein-Friesian cattle via deep metric learning, Comput. Electron. Agric. 185 (July 2020) (2021) 106133, https://doi.org/10.1016/j.compag.2021.106133.

[25] G.L. Menezes, A.A. Alves, A. Negreiro, R.E. Ferreira, S. Higaki, E. Casella, G.J. Rosa, J.R. Dorea, Color-independent cattle identification using keypoint detection and Siamese neural networks in closed- and open-set scenarios, J. Dairy Sci. 5 (2025), https://doi.org/10.3168/jds.2024-26069, https://linkinghub.elsevier.com/retrieve/pii/S0022030225003340.

[26] M. Perneel, I. Adriaens, J. Verwaeren, B. Aernouts, Dynamic multi-behaviour, orientation-invariant re-identification of Holstein-Friesian cattle, Sensors 25 (10) (2025) 2971, https://doi.org/10.3390/s25102971, https://www.mdpi.com/1424-8220/25/10/2971.

[27] X. Xu, Y. Wang, Y. Shang, G. Yang, Z. Hua, Z. Wang, H. Song, Few-shot cow identification via meta-learning, Inf. Process. Agric. 12 (1) (2024) 80–90, https://doi.org/10.1016/j.inpa.2024.04.001.

[28] W. Andrew, C. Greatwood, T. Burghardt, Visual localisation and individual identification of Holstein Friesian cattle via deep learning, in: IEEE International Conference on Computer Vision Workshops 2018-Janua, 2017, pp. 2850–2859.

[29] T.T. Zin, C.N. Phyo, P. Tin, H. Hama, I. Kobayashi, Image technology based cow identification system using deep learning, in: Proceedings of the International Multiconference of Engineers and Computer Scientists, 2018, pp. 236–247.

[30] R. Wang, R. Gao, Q. Li, C. Zhao, L. Ru, L. Ding, L. Yu, W. Ma, An ultra-lightweight method for individual identification of cow-back pattern images in an open image set, Expert Syst. Appl. 249 (PB) (2024) 123529, https://doi.org/10.1016/j.eswa.2024.123529.

[31] J. Gao, T. Burghardt, N.W. Campbell, Label a Herd in Minutes: Individual Holstein-Friesian Cattle Identification, LNCS, vol. 13374, Springer International Publishing, 2022.

[32] P. Yu, T. Burghardt, A.W. Dowsey, N.W. Campbell, Holstein-Friesian re-identification using multiple cameras and self-supervision on a working farm, Comput. Electron. Agric. 237 (PB) (2025) 110568, https://doi.org/10.1016/j.compag.2025.110568.

[33] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, IEEE Trans. Pattern Anal. Mach. Intell. 42 (2) (2020) 386–397, https://doi.org/10.1109/TPAMI.2018.2844175.

[34] K. Sun, B. Xiao, D. Liu, J. Wang, Deep high-resolution representation learning for human pose estimation, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2019-June, 2019, pp. 5686–5696.

[35] R. Jonker, A. Volgenant, A shortest augmenting path algorithm for dense and sparse linear assignment problems, Computing 38 (4) (1987) 325–340, https://doi.org/10.1007/BF02278710, http://link.springer.com/10.1007/BF02278710.

[36] P. Singh, E.J. Delp, A.R. Reibman, Video-analytics task-aware quad-tree partitioning and quantization for HEVC, in: 2022 IEEE International Conference on Image Processing (ICIP), IEEE, 2022, pp. 2936–2940, https://ieeexplore.ieee.org/document/9897965/.

[37] C.C. Aye, T.T. Zin, I. Kobayashi, Black cow tracking by using deep learning-based algorithms, ICIC Express Lett. Part B, Appl. 13 (12) (2022) 1313–1319, https://doi.org/10.24507/icicelb.13.12.1313.

[38] A.S. Thu Moe, T.T. Zin, P. Tin, M. Aikawa, I. Kobayashi, Minimizing resource usage for real-time network camera tracking of black cows, in: Y.-W. Chen (Ed.), Tenth International Conference on Multimedia and Image Processing (ICMIP 2025), vol. 13701, SPIE, 2025, pp. 89–96, https://www.spiedigitallibrary.org/conference-proceedings-of-spie/13701/3070347/Minimizing-resource-usage-for-real-time-network-camera-tracking-of/10.1117/12.3070347.full.

[39] S. Myat Noe, T.T. Zin, I. Kobayashi, P. Tin, Optimizing black cattle tracking in complex open ranch environments using YOLOv8 embedded multi-camera system, Sci. Rep. 15 (1) (2025) 6820, https://doi.org/10.1038/s41598-025-91553-4.

[40] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, R. Girshick, Detectron2, https://github.com/facebookresearch/detectron2, 2019.

[41] MMPose Contributors, OpenMMLab pose estimation toolbox and benchmark, https://github.com/open-mmlab/mmpose, 2020.